Mathematical Sciences Technical Reports (MSTR)

Mathematics

7-27-2012

Robust Analysis of Metabolic Pathways

Emily Gruber
Rose-Hulman Institute of Technology

Amy Ko
Rose-Hulman Institute of Technology

Michael MacGillvray
Rose-Hulman Institute of Technology

Miranda Sawyer
Rose-Hulman Institute of Technology

Recommended Citation

Robust Analysis of Metabolic Pathways

Emily Gruber, Amy Ko, Michael MacGillivray, and Miranda Sawyer

Adviser: Allen G. Holder

Mathematical Sciences Technical Report Series
MSTR 12-02

July 27, 2012

Robust Analysis of Metabolic Pathways

Emily Gruber[a], Amy Ko[b], Michael MacGillivray[c], and Miranda Sawyer[d]

[a]Carleton College, [b]Amherst College , [c]University of Notre Dame, [d]Northeastern State University

Advised by Dr. Allen Holder, Rose-Hulman Institute of Technology

July 27, 2012

Abstract

Flux Balance Analysis (FBA) is a widely used computational model for studying the metabolic pathways of cells and the role individual metabolites and reactions play in maintaining cell function. However, the successes of FBA have been limited by faulty biological assumptions and computational imperfections. We introduce Robust Analysis of Metabolic Pathways (RAMP) to provide a more theoretically sound and computationally accurate model of cellular metabolism. RAMP overcomes the faulty assumptions of traditional FBA by allowing deviation from steady-state and accounting for variability across a cellular culture. Computationally, RAMP more successfully predicts the lethality of gene knockouts and reduces degeneracy in optimal flux values. Analytical results establish the stability of RAMP under perturbations in modeling parameters. The inclusion of new modeling parameters in RAMP opens the possibility of modeling different cellular cultures in a wider range of conditions, including non-optimized cultures. We conclude that RAMP is an improvement over traditional FBA and deserves further study.

Contents

1 Introduction

Every cell is composed of three main networks: the gene-regulatory network, the protein-interaction network, and the metabolic network. The metabolic network consists of a series of chemical reactions that convert nutrients into products used in cellular functions. The chemicals involved in these reactions are called metabolites, and the net rate of a metabolic reaction is called the reaction flux. The reactions and metabolites can be experimentally determined by genome sequencing, but it is difficult to obtain empirical information about the rate and necessity of each reaction.

Flux Balance Analysis (FBA) is a mathematical model of cellular metabolism that is capable of computationally inferring information about reaction rates and predicting cellular responses to environmental changes and mutations. Traditionally, FBA makes these inferences on the basis of three modeling assumptions. First, the model asserts that a cell's metabolism exists in steady-state, meaning the concentration of each metabolite remains constant. Second, the model imposes environmental bounds on each reaction flux. Finally, the model assumes that cells achieve some biological objective. The most commonly used objective is the optimization of growth rate, while other objectives include the maximization of ATP production and minimization of energy expenditure.

Under these three modeling assumptions, traditional FBA analyzes metabolism through the lens of a linear optimization problem. The model has been used in various applications, such as predicting the essentialness of genes and computing possible values for each reaction rate. Fields making use of FBA range from metabolic engineering to drug development. For instance, understanding how to increase or eliminate gene expression can be useful in strain engineering, which is used extensively in the development of pharmaceuticals, fuels, and specialty chemicals.

However, the successes of FBA have been limited by faulty biological assumptions and computational imperfections. The FBA model optimizes cellular growth rate through the use of a pseudo chemical reaction that represents the conversion of certain metabolites in the correct proportions into biomass. The metabolites in this reaction have experimentally-derived, non-integer coefficients that are averages calculated over the entire culture. Individual cells need not behave according to these averages; in fact, experimental evidence suggests significant deviation from these mean values [?]. Thus, traditional FBA fails to account for variation across cellular cultures. Additionally, the steady-state assumption is biochemically inaccurate. Individual cells typically exist near but not at steady-state, so the steady-state assumption represents an oversimplification of the biological reality at hand. On the computational level, FBA yields many false positives in predicting lethal gene knockouts and suffers from degeneracy in optimal flux values.

We introduce a new approach to metabolic modeling, Robust Analysis of Metabolic Pathways (RAMP), to correct these two flaws. RAMP allows variation in the growth coefficients across a culture and permits deviation from steady-state, subject to some maximum bound. The traditional requirement that all cells be at steady-state is then reformulated as a probabilistic requirement that some percentage of cells in the culture exist near steady-state.

What follows is a thorough exposition of RAMP and its results. We begin with a mathematical overview of metabolic modeling and develop the optimization problems on which the old and new models are predicated. We then detail the theoretical aspects of robust optimization and its computational implementation. We discuss the computational results

of RAMP and compare them to the results of the traditional model. Finally, we establish analytic results showing the stability of a robust linear program under perturbations in modeling parameters.

2   Mathematical Overview

2.1   Metabolic Modeling

To model a biological system mathematically, it is first necessary to represent its elements as mathematical objects. We therefore begin by introducing a mathematical representation of a metabolic system.

A metabolic system is comprised of the following biological elements and corresponding mathematical quantities:

◇ A set of m metabolites $x_1, \ldots, x_m$ and their concentrations $[x_1], \ldots, [x_m]$.

◇ A set of n chemical reactions involving the metabolites, with the jth reaction having the form
$$a_{1j}x_1 + \cdots + a_{mj}x_m \rightleftharpoons b_{1j}x_1 + \cdots + b_{mj}x_m,$$
with forward and reverse rate coefficients k_+^j and k_-^j.
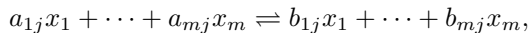
Each reaction has a net flux, or rate, given by a rate law from chemistry:
$$v_j = k_+^j[x_1]^{a_{1j}}[x_2]^{a_{2j}} \cdots [x_m]^{a_{mj}} - k_-^j[x_1]^{b_{1j}}[x_2]^{b_{2j}} \cdots [x_m]^{b_{mj}}.$$

The rate of change of the concentration of the ith metabolite is then calculated by scaling each reaction rate by the stoichiometric coefficient of that metabolite and summing the rates, i.e.
$$\frac{d[x_i]}{dt} = \sum_{j=1}^{n}(b_{ij} - a_{ij})v_j.$$

The traditional FBA model develops these fundamental concepts into a metabolic model in the following way.

2.2   A Linear Model

FBA first expresses the above data in a way that is convenient to analyze mathematically. Let S (the stoichiometric matrix) be the $m \times n$ matrix whose i^{th} row S_i contains the stoichiometric coefficients $b_{ij} - a_{ij}$ of the ith metabolite and v be the the $n \times 1$ column vector containing the reaction fluxes. Then $\frac{d[x_i]}{dt}$ as expressed above is $\sum_{j=1}^{n} S_{ij}v_j$, the ith entry of the product Sv.

The stoichiometric reaction coefficients S_{ij} are known and the goal is to determine the reaction fluxes v_j. Several assumptions are imposed to determine realistic values for v.

Traditional FBA first assumes that cells exists in steady-state, meaning the concentration of each metabolite remains constant, or $\frac{d[x_i]}{dt} = S_i v = 0$ for each i. Hence v must solve the homogeneous linear system $Sv = 0$.

Since many solutions to this system typically exist, additional constraints are necessary. Biologically, the environment of the cell places bounds on possible flux rates. Let L and U be vectors whose j^{th} entries contain the lower and upper bounds (sometimes infinite) on flux v_j, and add the constraint $L \le v \le U$.

For evolutionary reasons a cell typically allocates its resources in such a way as to maximize growth rate. Among the environmentally constrained solutions to $Sv = 0$, FBA thus selects those which maximize the growth rate of the cell. To this end it seeks flux vectors v such that the growth flux v_{growth} of the cell is maximized. FBA is sometimes based on optimization of some other biological objective, but growth rate is the predominantly used objective. We therefore assume FBA maximizes growth rate from this point onward.

The traditional FBA model is therefore the following constrained linear optimization problem:
$$\max\{v_{\text{growth}} \colon Sv = 0, L \le v \le U\}.$$

2.3   RAMP: A Probabilistic Model

We now describe the modeling changes made by Robust Analysis of Metabolic Pathways to address the faulty assumptions of the linear FBA model.

While stoichiometric coefficients are typically integer-valued and hence are the same for every cell and can be known with certainty, the coefficients of the growth reaction are averages estimated from batch cultures. The coefficients for any individual cell need not equal these averages, and indeed experimental data suggests substantial deviation from these averages across a culture. A more accurate model of a cellular culture can therefore be obtained by treating the growth coefficients as random variables rather than fixed values, which in turn makes the product $S_i v$ a random variable.

Rather than setting each $S_i v = 0$, we therefore assume each $S_i v$ is a normally distributed random variable and impose a probabilistic constraint that $S_i v$ remains within the interval $[-M_i, M_i]$ with some degree of certainty, i.e.

$$P(S_i v > M_i) \le \epsilon \text{ and } P(S_i v < -M_i) \le \epsilon.$$

Since $S_i v$ is normally distributed, $\left\{\frac{S_i v - \mu_i}{\sigma_i}\right\}$ is normally distributed with mean 0 and standard deviation 1. Let μ_i denote the mean of each scenario and σ_i denote the standard deviation. The above constraints are therefore equivalent to

$$\frac{M_i - \mu_i}{\sigma_i} \ge \delta_{1-\epsilon} \text{ and } \frac{-M_i - \mu_i}{\sigma_i} \le \delta_\epsilon$$

where $\delta_{1-\epsilon}$ and δ_ϵ are the $1 - \epsilon$ and ϵ percentiles. Rearranging these inequalities and noting that $-\delta_\epsilon = \delta_{1-\epsilon}$, this constraint is equivalent to the condition that

$$\sigma_i \le \min\left\{\frac{M_i - \mu_i}{\delta_{1-\epsilon}}, \frac{M_i + \mu_i}{\delta_{1-\epsilon}}\right\}.$$

RAMP can thus be expressed as an optimization problem subject to a probabilistic constraint on each σ_i:

$$\max\left\{v_{\text{growth}} : \sigma_i \leq \frac{M_i \pm \mu_i}{\delta_{1-\epsilon}} \,\forall\, i, L \leq v \leq U\right\}.$$

RAMP's probabilistic constraint is advantageous in that it better represents the true biological state of a cellular culture, but it risks losing mathematical tractability with the loss of the convenient linear optimization problem. Fortunately, the probabilistic RAMP constraint can be made computationally feasible through conversion to two well-studied types of optimization problems, robust linear programs and second-order cone programs. The following sections detail that conversion.

2.4 RAMP as Robust Linear Program

2.4.1 Introduction to Linear Programming and Robust Linear Programming

Both FBA and RAMP are based on maximization of v_{growth}, which means that each model consists of an optimization problem. The type of optimization used in FBA is Linear Programming (LP), in which the objective function being maximized and constraints on that objective function are linear. We introduce an LP problem in standard form.

Definition. A *linear programming problem* is an optimization problem of the form

$$\min\{c^T x : Ax \leq b\}$$

where c is a constant vector called the cost vector and $c^T x$ is being minimized subject to the two constraints on x.

Many linear optimization problems that appear in different form can be converted into standard LP form. For example, an optimization problem with an inequality constraint $Ax \leq b$ can be expressed as an equality constrain by the introduction of slack variables, $Ax + s = b$. Similarly, a maximization problem can be converted to a minimization problem by minimizing the negation of the set to be maximized.

An optimization problem in standard LP form is called a primal problem, and every such problem has a related optimization problem called the dual problem. The primal problem has its own primal variables and primal constraints and its solutions are an upper bound for possible optimal solutions to the dual problem. The related dual problem is a maximization problem with its own dual variables and dual constraints, and dual solutions are a lower bound to optimal solutions to the original problem. An example of the primal and dual of the standard LP problem is

The canonical forms of a primal SOCP and its dual are the following:

$$\begin{array}{cc} \text{primal:} & \text{dual:} \\ \min\left\{c^T x : Ax \leq b\right\}, & \max\left\{\rho^T b : (c^T - \rho^T A) \geq 0\right\}. \end{array}$$

Here, the primal variable is x and the dual variable is ρ. Both variables have their own constraints. It is not immediately evident how we form the dual from the primal. We discuss

this algorithm in Section 2.5.1. A pair of primal and dual variables x, ρ yield an optimal solution to the primal and dual problem under three necessary and sufficient conditions:

1. Primal Feasibility (x satisfies the primal constraints.)
2. Dual Feasibility (ρ satisfies the dual constraints.)
3. Complementarity (The values of the objective functions are equal, i.e. $c^T x = \rho^T b$.)

Take note of the complementarity condition, which verifies the optimality of the solution. Because the primal problem was an upper bound on the solution and the dual problem was a lower bound, the point at which they are equal must be an optimal solution to the problem.

FBA consists of a linear programming problem, while RAMP can be formulated as a robust linear programming problem, a type of adaptation of an LP problem. Robust linear programs were designed to treat uncertainty in the rows of the matrix A in the LP constraint $Ax = b$. Instead of requiring each row of A to be one specific vector, a robust linear program enforces each row constraint $A_i x = b_i$ for a set of possible values for A_i, called an uncertainty set.

Definition. A *Robust linear programming problem* is an optimization problem of the form

$$\min \{c^T x : A_i x \leq b_i \,\forall\, A_i \in \mathcal{U}_i\}$$

where c is a constant vector and $\mathcal{U}_i$ denotes an uncertainty set. The optimization of $c^T x$ is taken over the set of all vectors x such that $A_i x \leq b_i$ for every vector A_i in the uncertainty set $\mathcal{U}_i$.

We will show that RAMP is equivalent to a Robust LP with ellipsoidal uncertainty sets. An ellipsoid is defined to be the set

$$\mathcal{U}_i = \{\bar{A}_i + u^T R_i : u^T u \leq 1\},$$

where $\bar{A}$ is a fixed vector and R is a matrix. The definition of an ellipsoid can be understood geometrically in the following way. The set $u^T u \leq 1$ represents the unit sphere. The matrix R distorts the unit sphere through multiplication into an elliptical shape. Addition of the vector $\bar{A}$ then re-centers the ellipsoid at $\bar{A}$.

We use a visual example to demonstrate the relationship between an LP and a Robust LP.

Observe the simple LP problem:

$$\begin{aligned}
\max\, &x_1 + x_2 \\
&\text{subject to} \\
&x_1 + x_2 \leq 1 \\
&x_1 \geq 0, x_2 \geq 0.
\end{aligned} \tag{1}$$

Note that the constraint $x_1 + x_2 \leq 1$ can be formulated as $Ax \leq b$, in the standard LP where A is the vector $[1, 1]$ and $b = 1$. To convert (1) to a Robust LP, we allow A to be chosen from an uncertainty set. The Robust LP is

$$\max x_1 + x_2$$
$$\text{subject to}$$
$$Ax \leq 1 \, \forall \, A \in \mathcal{U}$$
$$\mathcal{U} = \left\{ [1,1] + u^T \begin{bmatrix} 0.1 & 0 \\ 0 & 0.2 \end{bmatrix} : u^T u \leq 1 \right\} \tag{2}$$
$$x_1 \geq 0, x_2 \geq 0.$$

We present a graph of the feasible solution sets for each problem to demonstrate how the two compare.
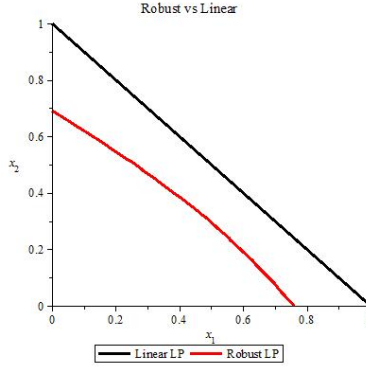
The black line represents the boundary of the set of feasible vectors for the LP. The red curve represents the same for the Robust LP. Everything to the left of each curve represents feasible points for each problem. Notice that the Robust LP is more restrictive because there exists some x that are LP feasible and Robust LP infeasible.

We now discuss how RAMP can be converted to a Robust LP problem.

2.4.2 Converting RAMP to a Robust LP

The conversion of RAMP to a Robust LP depends upon an approximation of the normal distribution of S_i by a finite set of discrete scenarios, each assigned some probability. Little information is lost if a sufficient number of scenarios are taken, and with this approximation it becomes possible to express every possible case and analyze them mathematically.

So suppose each S_i is assigned q possible scenarios, and let S_{ik} denote the row vector of stoichiometric coefficients for the ith metabolite under scenario k. Each scenario is assigned a probability p_k.

The mean and variance of $S_i v$ are then:

$$\mu_i = \sum_{k=1}^{q} p_k S_{ik} v$$

$$\sigma_i^2 = \sum_{k=1}^{q} p_k (S_{ik} v - \mu_i)^2.$$

These expressions of μ_i and σ_i derived from the scenario approximation enable the proof of the following theorem:

Theorem. The robust FBA model with its probabilistic constraint

$$\max \left\{ v_{\text{growth}} : \sigma_i \leq \frac{1}{\delta_{1-\epsilon}} (M_i \pm \mu_i),\ i = 1, 2, \ldots, m \right\}$$

is equivalent to the robust LP problem

$$\max \left\{ v_{\text{growth}} :\ -M_i \leq S_i v \leq M_i\ \forall\, S_i \in \mathcal{U}_i,\ i = 1, 2, \ldots, m \right\}.$$

$\mathcal{U}_i$ is the ellipsoidal uncertainty set

$$\mathcal{U}_i = \left\{ p^T S_i' + u^T \left(\delta_{1-\epsilon} \sqrt{P} (I - ep^T) S_i' \right) :\ u^T u \leq 1 \right\}$$

where p is a vector containing the q scenario probabilities, S_i' is a $q \times n$ matrix whose k^{th} row contains the kth scenario for S_i, P is a diagonal matrix containing the scenario probabilities, and e is a vector of 1's.

Proof. We first establish an alternate form for a robust LP problem. We show that

$$\max \left\{ c^T x :\ A_i x \geq b_i\ \forall\, A_i \in \mathcal{U}_i,\ \forall\, i \right\},\quad \mathcal{U}_i = \{ \bar{A}_i + u^T R_i : u^T u \leq 1 \}$$

is equivalent to

$$\max \left\{ c^T x :\ ||R_i x|| \leq \bar{A}_i x - b_i,\ \forall\, i \right\}.$$

Observe the following:

$$
\begin{aligned}
A_i x \geq b_i\ \forall\, A_i \in \mathcal{U}_i\ &\Longleftrightarrow\ 0 \leq A_i x - b_i\ \forall\, A_i \in \mathcal{U}_i \\
&\Longleftrightarrow\ 0 \leq \min \{ A_i x - b_i : A_i \in \mathcal{U}_i \} \\
&= \min \{ (\bar{A}_i + u^T R_i) x - b_i : u^T u \leq 1 \} \\
&= \bar{A}_i x - b_i + \min \{ u^T R_i x : u^T u \leq 1 \} \\
&= \bar{A}_i x - b_i + \max \{ u^T R_i x : u^T u \leq 1 \} \\
&= \bar{A}_i x - b_i - ||R_i x||.
\end{aligned}
$$

Therefore $A_i x \geq b_i\ \forall\, A_i \in \mathcal{U}_i\ \Longleftrightarrow\ ||R_i x|| \leq \bar{A}_i x - b_i$, as required.

We now convert the probabilistic FBA model to a robust LP model by expressing it in this alternate robust LP form. From the expressions of μ_i and σ_i developed in the preceding section, we have that:

$$\mu_i = \sum_{k=1}^{q} p_k S_{ik} v = p^T S_i' v.$$

(Recall that $p^T = (p_1, p_2, \ldots, p_q)$ and S_i' is the $q \times n$ matrix whose k^{th} row holds the coefficients of the ith metabolite under scenario k.) Therefore

$$\sigma_i^2 = \sum_{k=1}^{q} p_k (S_{ik} v - \mu_i)^2 = \sum_{k=1}^{q} p_k (S_{ik} v - p^T S_i' v)^2.$$

10

Note that an expression of the form $\sum_{k=1}^{q} p_k x_k^2$ can be written

$$\sum_{k=1}^{q} p_k x_k^2 = (x_1, x_2, \ldots, x_q) \begin{bmatrix} p_1 & 0 & \ldots & 0 \\ 0 & p2 & \ldots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \ldots & p_q \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_q \end{pmatrix} = X^T P X$$

and so

$$\sigma_i^2 = (S_i'v - ep^T S_i'v)^T \sqrt{P}\sqrt{P}(S_i'v - ep^T S_i'v)$$
$$= \left(\sqrt{P}(S_i'v - ep^T S_i'v)\right)^T \left(\sqrt{P}(S_i'v - ep^T S_i'v)\right)$$
$$= \left\|\sqrt{P}(S_i'v - ep^T S_i'v)\right\|^2$$
$$= \left\|\sqrt{P}(I - ep^T)S_i v\right\|^2 .$$

Therefore,

$$\sigma_i \leq \frac{1}{\delta_{1-\epsilon}}(M_i \pm \mu_i) \iff \left\|\sqrt{P}(I - ep^T)S_i v\right\| \leq \frac{1}{\delta_{1-\epsilon}}(M_i \pm \mu_i)$$
$$\iff \left\|\sqrt{P}(I - ep^T)S_i v\right\| \leq \frac{1}{\delta_{1-\epsilon}}(M_i \pm p^T S_i'v)$$
$$\iff \left\|\delta_{1-\epsilon}\sqrt{P}(I - ep^T)S_i v\right\| \leq \pm p^T S_i'v + M_i.$$

This establishes that the condition

$$\sigma_i \leq \frac{1}{\delta_{1-\epsilon}}(M_i \pm \mu_i)$$

is equivalent to

$$\left\|(\delta_{1-\epsilon}\sqrt{P}(I - ep^T)S_i')v\right\| \leq \pm p^T S_i'v + M_i.$$

Note that this statement embodies two conditions:

$$\left\|(\delta_{1-\epsilon}\sqrt{P}(I - ep^T)S_i')v\right\| \leq p^T S_i'v + M_i$$

and

$$\left\|(\delta_{1-\epsilon}\sqrt{P}(I - ep^T)S_i')v\right\| \leq -p^T S_i'v + M_i,$$

which are both in the alternate form of a robust LP constraint, with $\bar{A}_i = p^T S_i'$, $R_i = \delta_{1-\epsilon}\sqrt{P}(I - ep^T)S_i'$, and $b_i = -M_i$. They are therefore equivalent to the following two constraints:

$$S_i v \geq -M_i \, \forall \, S_i \in \mathcal{U}_i, \, \mathcal{U}_i = \left\{p^T S_i + u^T(\delta_{1-\epsilon}\sqrt{P}(I - ep^T)S_i') : u^T u \leq 1\right\}$$

$$S_i v \geq -M_i \, \forall \, S_i \in \mathcal{U}_i, \, \mathcal{U}_i = \left\{-p^T S_i + u^T(\delta_{1-\epsilon}\sqrt{P}(I - ep^T)S_i') : u^T u \leq 1\right\}$$

and this last constraint can be re-expressed:

$$-S_i v \geq -M_i \, \forall \, S_i \in \mathcal{U}_i, \, \mathcal{U}_i = \left\{p^T S_i - u^T(\delta_{1-\epsilon}\sqrt{P}(I - ep^T)S_i') : u^T u \leq 1\right\}$$
$$= \left\{p^T S_i + u^T(\delta_{1-\epsilon}\sqrt{P}(I - ep^T)S_i') : u^T u \leq 1\right\}$$

11

so that the two constraints together are equivalent to

$$-M_i \leq S_i v \leq M_i \, \forall S_i \in \mathcal{U}_i, \ \mathcal{U}_i = \left\{ p^T S_i + u^T (\delta_{1-\epsilon} \sqrt{P}(I - ep^T) S_i') : u^T u \leq 1 \right\}.$$

We have now derived two useful alternative forms for the probabilistic FBA model:

$$\max \left\{ v_{\text{growth}} : \ \left\| (\delta_{1-\epsilon} \sqrt{P}(I - ep^T) S_i') v \right\| \leq \pm p^T S_i' v + M_i, \ \forall \, i \right\}$$

and

$$\max \left\{ v_{\text{growth}} : \ -M_i \leq S_i v \leq M_i \, \forall \, S_i \in \mathcal{U}_i, \ \forall \, i \right\},$$
$$\mathcal{U}_i = \left\{ p^T S_i + u^T (\delta_{1-\epsilon} \sqrt{P}(I - ep^T) S_i') : u^T u \leq 1 \right\}$$

This completes the proof. □

While the equivalence of the the probabilistic and robust LP forms of RAMP highlights the relationship between linear FBA and RAMP, the norm inequality developed in the proof above is the most convenient way to computationally implement the RAMP constraint. An optimization problem subject to a constraint in this form is called a second-order cone program (SOCP). Before proceeding further in our discussion of RAMP, we take a brief foray into the world of conic optimization. The following section explains what a second-order cone program is and details key aspects of SOCP theory.

2.5 Theory of Second-Order Cone Programming

Definition. A cone is a set K of vectors such that if $k \in K$ and $\alpha > 0$, then $\alpha k \in K$.

Second-order cone programming in particular is concerned with quadratic cones.

Definition. The second-order (quadratic) cone of dimension n is the set

$$K_q^n = \left\{ \left(\begin{array}{c} z_0 \\ z \end{array} \right) \middle| z_0 \in \mathbb{R}, z \in \mathbb{R}^{n-1}, ||z|| \leq z_0 \right\}.$$

Note that there is only one second-order cone of each dimension. The statement $||z|| \leq z_0$ is important for visualizing the cone. As an example, consider $z = (z_1, z_2)^T$, so $||z|| = \sqrt{z_1^2 + z_2^2}$. K_q^3 is then the set of (z_0, z_1, z_2) satisfying $\sqrt{z_1^2 + z_2^2} \leq z_0$. If $\sqrt{z_1^2 + z_2^2}$ is $0, z_0$ can be any positive value. As we increase z_1 and z_2, z_0 is restricted more. As we vary z_1 and z_2, we obtain the ice-cream cone shape that we imagine for a cone.

Given a cone K, the dual cone is defined as $K^* = \left\{ c : c^T x \geq 0 \, \forall \, x \in K \right\}$, i.e. a vector c is in the dual cone in and only if c is within ninety degrees of every vector in the cone. All second-order cones are self dual, $K^* = K$. $\mathbb{R}_+^n$ is also a self-dual cone.

A second-order cone program optimizes a linear function subject to conic constraints:

Definition. A second-order cone programming problem is an optimization problem of the form:

$$\min \left\{ \sum_{i=1}^{n} c_i^T x_i : \ \sum_{i=1}^{n} A_i x_i = b, x_i \in K_i, i = 1, 2, ..., n \right\},$$

where c_i and x_i are vectors, A_i is a matrix, and $K_1, \ldots, K_n$ are self-dual cones.

Like linear programs, every second-order cone program has a related dual problem. The primal and dual of an SOCP stand in the same relation to one another as the primal and dual of an LP, including the three conditions for optimality. We will now provide an example for deriving the dual problem from the primal with an SOCP.

2.5.1 Formulating the Dual from the Primal

Consider the primal SOCP:

$$\min \left\{ c_1^T x_1 + c_2^T x_2 : \ A_1 x_1 + A_2 x_2 = b, \ x_1 \in K_1, \ x_2 \in K_2 \right\}.$$

The Lagrangian of an optimization constraint is a special function useful for solving the optimization problem. The Lagrangian of the primal SOCP problem is the following:

$$\mathcal{L}(x, \lambda) = c_1^T x_1 + c_2^T x_2 - \lambda^T (A_1 x_1 + A_2 x_2 - b).$$

We claim that the primal is equivalent to the following problem:

$$\min_{x_1 \in K_1, x_2 \in K_2} \max_{\lambda \, \text{free}} c_1^T x_1 + c_2^T x_2 - \lambda^T (A_1 x_1 + A_2 x_2 - b)$$

This optimization problem is equivalent to the original primal problem for the following reasons. If the primal constraint $A_1 x_1 + A_2 x_2 = b$ is satisfied, $\lambda^T (A_1 x_1 + A_2 x_2 - b) = 0$ and so the objective function of the reformulated problem is the same as that of the primal for all x_1, x_2 satisfying the primal constraint. If the primal constraint is violated, $(A_1 x_1 + A_2 x_2 - b)$ is non-zero in at least one entry, so $\lambda^T (A_1 x_1 + A_2 x_2 - b)$ can be made arbitrarily large by choosing a suitable λ. Therefore the outside minimization problem is never solved by x_1 and x_2 that violate the primal constraint citeEmilyA.

The dual is the maximization problem obtained by switching the max and min above and separating those terms that contain some x and those that do not:

$$\max_{\lambda \, \text{free}} \min_{x_1 \in K_1 x_2 \in K_2} (c_1 - A_1^T \lambda)^T x_1 + (c_2 - A_2^T \lambda)^T x_2 + b^T \lambda.$$

The dual problem can be put in the form of a cone problem by reversing the process applied to the primal above. We will show that the dual is equivalent to the following cone problem:

$$\max \left\{ b^T \lambda : \ (c_1^T - A_1^T \lambda)^T \in K_1, \ (c_2^T - A_2^T \lambda)^T \in K_2 \right\}.$$

The dual objective function includes all parts of the dual Lagrangian that do not include an x term. All terms in the dual Lagrangian that do include an x will form cone constraints.

To make the problems equivalent, the dual constraints must make any λ infeasible that allows x_1 or x_2 to be chosen in such a way that $(c_1 - A_1^T \lambda)^T x_1 + (c_2 - A_2^T \lambda)^T x_2$ becomes arbitrarily small, because the max min problem cannot achieve optimality under such conditions.

Let's suppose $(c_1^T - A_1^T \lambda)^T \notin K_1^*$ (K_1^* is the dual of K_1.) Then, by the definition of a dual cone, $\exists x_1 \in K_1$ such that $(c_1^T - A_1^T \lambda)^T x_1 < 0$. Because K_1 is a cone, $\alpha x_1 \in K_1, (\alpha > 0)$. So we can choose αx_1 so that $(c_1^T - A_1^T \lambda)^T \alpha x_1$ returns negative infinity. Hence,

if $(c_1^T - A_1^T \lambda)^T \notin K_1^*$, our inner minimization problem returns negative infinity, and our maximization becomes infeasible. By this argument we have that one of our dual constraints is $(c_1^T - A_1^T \lambda)^T \in K_1^*$, and since all second-order cones are self-dual, $(c_1^T - A_1^T \lambda)^T \in K_1$. A similar argument holds for $(c_2^T - A_2^T \lambda)^T$, and our second constraint is $(c_2^T - A_2^T \lambda)^T \in K_2$.

We conclude that the dual is indeed

$$\max \left\{ b^T \lambda : \ (c_1^T - A_1^T \lambda)^T \in K_1, \ (c_2^T - A_2^T \lambda)^T \in K_2 \right\}.$$

This method for finding the dual problem can be used more generally, including finding the dual of an LP. We adapt the method specifically to second-order cones because second-order cones are prevalent in applications. The self-duality of a second-order cone is the lynch-pin of the method.

2.5.2   RAMP as SOCP

Recall that we established an alternate form of RAMP's probabilistic constraint:

$$\left\| \left(\delta_{1-\epsilon} \sqrt{P}(I - ep^T)S_i' \right) v \right\| \leq \pm p^T S_i' v + M_i, \ \forall \, i.$$

Directly from the definition of a quadratic cone, this norm inequality is true if and only if the following cone conditions hold:

$$\left(p^T S_i' v + M_i, \delta_{1-\epsilon}\sqrt{P}(I - ep^T)S_i' v \right) \in K_q \ \forall \, i,$$

$$\left(-p^T S_i' v + M_i, \delta_{1-\epsilon}\sqrt{P}(I - ep^T)S_i' v \right) \in K_q \ \forall \, i.$$

Expressing the probabilistic RAMP constraint in this form allows the RAMP model to be expressed in the standard primal form of an SOCP. The following section, devoted to explaining computational aspects of RAMP and their results, will begin with a detailed explanation of how RAMP can be expressed as an SOCP and solved by a computer algorithm.

3   Computational Methods, Applications, and Results

3.1   Introduction

The goal of RAMP is to improve traditional metabolic models by introducing robustness. This is accomplished in two primary ways: by accounting for uncertainty in experimentally-derived metabolic quantities, and by abandoning steady-state assumptions that produced more tractable mathematics but a less accurate portrayal of the biological reality being studied.

On the level of mathematical modeling, these changes convert linear optimization problems to second-order cone programs. While less easy to solve than linear programs, second-order cone programs (SOCP) are solvable in polynomial time by publicly available solvers using an interior-point algorithm. The increase in mathematical complexity is thus not immense and, we believe, is worth the increased accuracy in portraying the cellular cultures under study.

This section is devoted to explaining the purpose, implementation, and results of the computational methods of RAMP. We develop several programs that mimic the traditional applications of metabolic modeling, compare their successes, and then explain how RAMP can be used to simulate metabolic situations that could not be analyzed using traditional methods.

3.2 Computing Tools

Constraint-Based Reconstruction and Analysis (COBRA) is a well-established tool used to model organisms on a genome scale. COBRA works with a metabolic network model that is reconstructed based on the annotated gene sequences and known biochemical reactions of the organism of interest. Genome-scale metabolic networks have been reconstructed for many organisms, including E. *coli*. In the following computational experiments, we use metabolic reconstructions of E. *coli* obtained from the BiGG Database [7] and utilize COBRA Toolbox for MATLAB [1] as a tool for implementing RAMP.

We use two publicly available solvers, SeDuMi and sdpt3, to solve our second-order cone problem. SeDuMi is a MATLAB toolbox for optimizing over symmetric cones using interior point algorithm; sdpt3 is a MATLAB implementation of infeasible path-following algorithms for solving conic optimization problems. Although both solvers should ideally give the same solution for a given optimization problem, the differences in their implementation methods resulted in different solutions. Generally, SeDuMi showed more stability and accuracy in solving LPs while sdpt3 performed better in solving SOCPs. We therefore used SeDuMi in experiments that involve solving LPs and sdpt3 to solve SOCPs.

3.3 Implementing Robust Flux Balance Analysis

The fundamental computational tool of RAMP is a robust form of Flux Balance Analysis, denoted RFBA. This section is devoted to explaining the computational implementation of RFBA as a second-order cone program. This process involves two steps: first, the steady-state constraint must be formulated in a way compatible with the standard form of an SOCP and expressed in the proper form for the solver. Second, the environmental bounds must be formulated in a way consistent with a standard SOCP.

3.3.1 Implementing cone constraints

SOCP solvers solve problems of the specific form

$$\min \left\{ c^T x : Ax = b, x_i \in K_i \, \forall \, i \right\}.$$

c denotes a constant vector, A and b are respectively a matrix and a vector, and the constraint $x_i \in K_i \, \forall \, i$ requires that x be divisible into sub-vectors x_i each of which lies in a cone K_i. Each cone K_i may be $\mathbb{R}^n$, $\mathbb{R}^n_+$, or the quadratic cone $K_q = \left\{ (z_0, z) : z \in \mathbb{R}^{n-1}, z_0 \geq ||z|| \right\}$ (Refer to section 2.5 for details).

We divide the rows of the matrix S into two categories. Let S^c denote the rows of S whose entries are certain, arranged according to their ordering in S, and let S^u denote the rows of S containing any uncertain entries, ordered in the same way.

15

We account for the constraints on S^c and S^u separately. We first express the constraints $-M_i \leq S_i^c v \leq M_i$ to be compatible with standard SOCP. Since SOCP requires a matrix equality $Ax = b$, we convert the each inequality $-M_i \leq S_i^c v \leq M_i$ into a pair of equalities by introducing two slack variables and requiring that

$$S_i^c v + s_i^1 = M_i$$

$$S_i^c v - s_i^2 = -M_i$$

where the slack variables s_i^1 and s_i^2 are restricted to the positive cone $\mathbb{R}_+^n$. Applying this process to each row of S^c, we obtain the following system:

$$\begin{bmatrix} S_1^c & 1 & 0 & 0 & 0 & \ldots \\ S_1^c & 0 & -1 & 0 & 0 & \ldots \\ S_2^c & 0 & 0 & 1 & 0 & \ldots \\ S_2^c & 0 & 0 & 0 & -1 & \ldots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix} \begin{bmatrix} v \\ \hline s_1^1 \\ s_1^2 \\ s_2^1 \\ s_2^2 \\ \vdots \end{bmatrix} = \begin{bmatrix} M_1 \\ -M_1 \\ M_2 \\ -M_2 \\ \vdots \end{bmatrix},$$

$$(s_1^1, s_1^2, s_2^1, \ldots) \in \mathbb{R}_+^n.$$

Now consider the rows in S^u. We have already shown that the constrains on S^u can be formulated as a pair of cone constrains:

$$||Rv|| \leq -\bar{S}_i v + M_i.$$

$$||Rv|| \leq \bar{S}_i v + M_i.$$

In other words, $(-\bar{S}_i v + M_i, Rv)$ and $(\bar{S}_i v + M_i, Rv)$ must lie in the quadratic cone K_q.

In a standard primal SOCP, cone constrains may only be placed upon sub-vectors of the unknown vector x. To implement these cone constraints, we therefore expand x to include segments (z_0^{i1}, z^{i1}) and (z_0^{i2}, z^{i2}) for each uncertain row i. We then add to the system $Ax = b$ the requirements that $S_i^u v + M_i = z_0^{i1}$, $R_i v = z^{i1}$ and $-S_i^u v - M_i = z_0^{i2}$, $R_i v = z^{i2}$. Requiring that (z_0^{i1}, z^{i1}) and (z_0^{i2}, z^{i2}) lie in K_q then imposes the desired constraints on the model.

These additions to $Ax = b$ result in the following system:

$$\left[\begin{array}{c|ccc|cccc|cccc|cc} & 1 & 0 & \ldots & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots \\ S^c & 0 & -1 & \ldots & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots \\ & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \\ \hline \bar{S}_1^u & 0 & 0 & \cdots & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots \\ R_1 & 0 & 0 & \cdots & 0 & -Id & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots \\ \bar{S}_1^u & 0 & 0 & \cdots & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots \\ R_1 & 0 & 0 & \cdots & 0 & 0 & 0 & -Id & 0 & 0 & 0 & 0 & 0 & \cdots \\ \hline S_2^u & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & \cdots \\ R_2 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 & -Id & 0 & 0 & 0 & \cdots \\ \bar{S}_2^u & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & \cdots \\ R_2 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -Id & 0 & \cdots \\ \hline \vdots & \vdots & \vdots & \cdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \ddots \end{array}\right] \begin{bmatrix} v \\ \hline s_1^1 \\ s_1^2 \\ \hline \vdots \\ \hline z_0^{11} \\ z^{11} \\ z_0^{12} \\ z^{12} \\ \hline z_0^{21} \\ z^{21} \\ z_0^{22} \\ z^{22} \\ \hline \vdots \end{bmatrix} = \begin{bmatrix} M_1^c \\ -M_1^c \\ \vdots \\ \hline M_1^u \\ 0 \\ -M_1^u \\ 0 \\ \hline M_2^u \\ 0 \\ -M_2^u \\ 0 \\ \hline \vdots \end{bmatrix},$$

$$(s_1^1, s_1^2, \ldots) \in \mathbb{R}_+^n, (z_0^{i1}, z^{i1}), (z_0^{i2}, z^{i2}) \in K_q \ \forall \ i.$$

The constraints on both $S^c v$ and $S^u v$ have now been expressed in SOCP form. The constraints on v, however, are not yet compatible with SOCP, since the bounds given in L and U may take any form whatsoever and so are not initially in the form of cone constraints.

3.4 Implementing environmental bounds

SeDuMi solves the optimization problem of the form

$$\min \left\{ c^T x : Ax = b \right\},$$

where each variable x_i is either free, non-negative, complex-valued, in a quadratic cone, or in a positive semi-definite cone.

For RAMP, we are only interested in variables that are either free, non-negative, or in quadratic cones. We define a structure K to give SeDuMi information about the types of input variables. K contains three fields: K.f, K.l, and K.q . K.f is the number of free variables; K.l is the number of non-negative variables; K.q is an array that contains the dimension of each quadratic cone.

The previous section describes how to set up the quadratic cone constraints. Here, we handle the constraint $L \leq v \leq U$ such that each variable v_i is either free or non-negative as required by the solver.

Given a problem of the form

$$\min \left\{ c^T v : Av = b, L \leq v \leq U \right\},$$

where L and U may take any values, we define A', b', and c' such that the given problem is reformulated as

$$\min \left\{ c'^T v' : A'v' = b', v_i' \in \mathbb{R} \text{ for } i = 1, \ldots, \text{K.f} , v_i' \geq 0 \text{ for } i = \text{K.f} + 1, \ldots, n \right\},$$

a form compatible with the solver.

We begin constructing A' column-by-column by handling the free variables in the original problem (i.e. any v_k where $-\infty < v_k < \infty$). SeDuMi requires that the first K.f columns of A' correspond to the free variables. We therefore find all columns of the original A that correspond to the free variables and place them in the first K.f columns of A'.

Next we handle the constant variables (i.e. any v_k where $v_k = \alpha_k$ for some constant α_k). The columns of A that correspond to constant variables do not appear in A', but are instead incorporated into b'. Notice that for any constant variable $v_k = \alpha_k$,

$$b_i = A_{i,1} v_1 +, \ldots, + A_{i,k-1} v_{k-1} + A_{i,k} \cdot \alpha_k + A_{i,k+1} v_{k+1} +, \ldots, A_{i,n} v_n$$

is equivalent to

$$b_i - A_{i,k} \cdot \alpha_k = A_{i,1} v_1 +, \ldots, + A_{i,k-1} v_{k-1} + A_{i,k+1} v_{k+1} +, \ldots, A_{i,n} v_n$$

for all i. This equivalence shows that any column of A that corresponds to a constant variable can be removed from A and be incorporated to the left hand side. So instead of placing such constant columns into A', we define b'_i as

$$b'_i = b_i - \sum_{constant\, v_k} A_{i,k} \cdot \alpha_k.$$

Now it remains to show how to handle variables that are neither free or constant. The remaining variables have bounds that fall into one of the three following cases:

1. $-\infty < v_k \leq \alpha_k$

2. $\alpha_k \leq v_k < \infty$

3. $\alpha_{k1} \leq v_k \leq \alpha_{k2}$, where $\alpha_{k1} < \alpha_{k2}$ are constants.

Although each case is handled slightly differently, an example of one sufficiently highlights the general idea of handling all the other cases. As an illustrative example, consider the case where $-\infty < v_k \leq \alpha_k$ for some constant α_k.

In order to transform v_k into a non-negative variable, we define a new variable $z_k = -v_k + \alpha_k$. Then the constraint $-\infty < v_k \leq \alpha_k$ becomes $0 \leq z_k$, a legal input for SeDuMi. Note that

$$\begin{aligned}
b_i &= A_{i,1}v_1 +, \ldots, A_{i,k}v_k +, \ldots, A_{i,n}v_n \\
&= A_{i,1}v_1 +, \ldots, A_{i,k}(-z_k + \alpha_k) +, \ldots, A_{i,n}v_n
\end{aligned}$$

is equivalent to

$$b_i - A_{i,k} \cdot \alpha_k = A_{i,1}v_1 +, \ldots, +(-A_{i,k})z_k +, \ldots, A_{i,n}v_n$$

for all i. So in order to preserve the system of equations with the new variable z_k, we place the negated column of A into A' and redefine b' as follows:

$$A'_{i,K.f+k} = -A_{i,k}$$

$$b' = b' - A'_{i,k} \cdot c_k.$$

Finally, we define c' to select the correct objective variable in A'.

If we handle each constraint $L_k \leq v_k \leq U_k$ in this manner for all k, then the original constraints can be re-expressed as

$$A'v' = b', v'_i \in \mathbb{R} \text{ for } i = 1, \ldots, K.f \,, v'_i \geq 0 \text{ for } i = K.f + 1, \ldots, n$$

as required.

Notice that the solution v' of the optimization problem is not equal to the solution to the original problem. In order to obtain the solution to the original problem, we map v' back to v by applying appropriate inverse operations.

3.5 Inferring Metabolic Parameters

A primary advantage of RAMP over linear FBA is the existence of additional modeling parameters in RAMP that either do not exist in linear FBA or are fixed at some value for mathematical convenience. These parameters are:

- ⋄ M, the deviation of metabolite concentrations from steady-state.

- ⋄ σ, the standard deviation of the growth coefficients.

- ⋄ $\bar{S}_i$, the average value of the ith row (i.e. of the growth coefficients).

All three of these parameters represent biologically meaningful information not modeled by linear FBA. Given biological information about any two parameters, RAMP can computationally infer information about the third and thus obtain information beyond the scope of linear FBA. Some examples are the following.

3.5.1 Steady-state deviation

The amount of deviation allowed from steady-state for each metabolite in the positive and negative directions, denoted by M_i^+ and M_i^- for the ith metabolite, controls the flexibility of the flux state of the cell. Increasing the vector M containing these bounds for each metabolite, therefore, increases the optimal growth rate whereas decreasing it has the opposite effect.

The minimal steady-state deviation necessary for cells to achieve their optimal growth rate is biologically relevant and computationally useful information. The following is an explanation of how a least-squares minimization of this steady-state deviation can be formulated as a second-order cone program.

3.5.1.1 Implementation *Notation*: Superscripts c and u are used to denote variables corresponding to certain and uncertain rows of the stoichiometric matrix, respectively. For example S^c denotes the matrix whose rows are the certain rows of S. j is used to index the uncertain rows of S, so that S_j^u denotes some uncertain row of S. Superscript $+$ and $-$ signs denote variables corresponding to upper and lower bounds. For instance, M_j^{u+} and M_j^{u-} denote the bounds on deviation from steady state for the jth uncertain metabolite.

The goal is to implement the following optimization problem as an SOCP:

$$\min \left\{ ||M|| : -M^{c-} \leq S^c v \leq M^{c+}, -M_j^{u-} + ||R_j v|| \leq \bar{S}_j v \leq M_j^{u+} - ||R_j v|| \ \forall j, L \leq v \leq U, v_{\text{growth}} \text{ fixed} \right\}.$$

The following steps convert this problem to an SOCP:

- ⋄ We incorporate the vector M into the unknown vector x rather than treating it as a set of known values.

- ⋄ Since an SOCP requires a linear objective function, we introduce an additional variable ϕ and minimize ϕ under the cone constraint $\phi \geq ||M||$. Minimization will of course force $\phi = ||M||$, so that minimizing ϕ becomes equivalent to minimizing $||M||$.

◇ The complete unknown vector thus becomes

$$x = \left(v, s_1^1, s_1^2, \ldots, \phi, M_1^{c+}, M_1^{c-}, \ldots, M_1^{u+}, M_1^{u-}, \ldots, z_0^{11}, z^{11}, \ldots\right).$$

◇ The entries in L and U corresponding to v_{growth} are fixed at the experimentally derived growth rate so that v_{growth} itself is fixed.

◇ Equations in the steady-state constraints involving M (formulated in the SOCP system $Ax = b$) must be rewritten to account for the fact that M is now an unknown. For instance, we can take the pair of equations

$$S_i^c v + s_i^1 = M_i^{c+}, \ \ S_i^c v - s_i^2 = -M_i^{c-}$$

and move the M terms to the left side

$$S_i^c v + s_i^1 - M_i^{c+} = 0, \ \ S_i^c v - s_i^2 + M_i^{c-} = 0$$

and then add a 1 and -1 to the rows enforcing these equations in the columns corresponding to M_i^{c+} and M_i^{c-}. When this process is executed for each equation containing an element of M, the resulting system $Ax = b$ is the following:

$$
\begin{bmatrix}
 & 1 & 0 & \cdots & 0 & -1 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots \\
S^c & 0 & -1 & \ddots & 0 & 0 & 1 & \ddots & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots \\
 & \vdots & \ddots & \ddots & \vdots & \vdots & \ddots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \\
\bar{S}_1^u & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & \cdots \\
R_1 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -Id & 0 & 0 & 0 & \cdots \\
\bar{S}_1^u & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & -1 & 0 & 0 & \cdots \\
R_1 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -Id & 0 & \cdots \\
 & \vdots & \vdots & \vdots & \cdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \ddots
\end{bmatrix}
\begin{bmatrix}
v \\ s_1^1 \\ s_1^2 \\ \vdots \\ \phi \\ M_1^{c+} \\ M_1^{c-} \\ \vdots \\ M_1^{u+} \\ M_1^{u-} \\ \vdots \\ z_0^{11} \\ z^{11} \\ z_0^{12} \\ z^{12} \\ \vdots
\end{bmatrix}
=
\begin{bmatrix}
0 \\ 0 \\ \vdots \\ 0 \\ 0 \\ 0 \\ 0 \\ \vdots
\end{bmatrix},
$$

$$(s_1^1, s_1^2, \ldots) \in \mathbb{R}_+^n, (z_0^{i1}, z^{i1}), (z_0^{i2}, z^{i2}) \in K_q \ \forall \, i, (\phi, M) \in K_q.$$

◇ We then impose the bounds L and U on v by formatting for the solver in the typical way.

◇ After obtaining a solution, we rearrange the elements of M so that they are ordered according to the original metabolite indices.
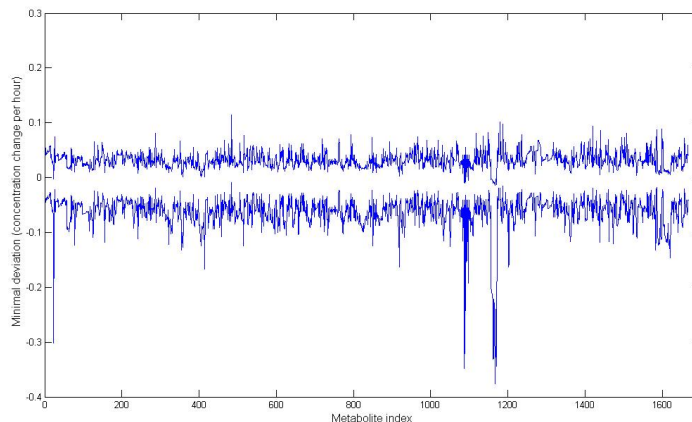
Figure 1: Minimal deviation from steady-state required to achieve the optimal growth rate

3.5.1.2 Results and Computational Uses While interesting and biologically mean-ingful in its own right, obtaining a minimal M is also useful for the purpose of further computations for it establishes a standardized M to use consistently in computations for a given model to generate comparisons across changes in other parameters. For instance, it establishes an M we can use to simulate gene knockouts with a guarantee that the cellular conditions imposed upon all the mutant cells are those that would yield the correct growth rate in non-mutant cells.

However, using the vector M obtained from the minimal M program often leads to com-putational instability, likely because it is the most restrictive M possible by design. We have found that taking the maximum element of the minimal M and allowing each metabolite that same maximal steady-state deviation in both directions yields the same growth rate with greater computational stability. The computational tests detailed throughout this sec-tion were for this reason executed using this method. We refer to the uniform M computed in this way as the "box M."

3.5.2 Growth as a function of probability

The parameter δ indicates the certainty at which the cells are near steady-state. Increasing δ, therefore, decreases the optimal growth rate by requiring a higher percentage of the cells to be near steady-state. Decreasing δ, on the other hand, allows more cells to deviate from steady-state, thereby giving them more flexibility to achieve a higher growth rate.

Figure 2 shows the optimal growth rate of E. *coli* core model as a function of δ. Here the box M was computed with $\delta = 3$ and 5 scenarios. The optimal growth rate was then computed with the fixed box M and varying δ values. The results are surprising in that the optimal growth rate drops significantly between $\delta = 4$ to $\delta = 5$, indicating that the constraint that the cells be near steady-state with 99.9999 percent probability is drastically more restrictive than 99.9937 percent. Further analysis is in progress.
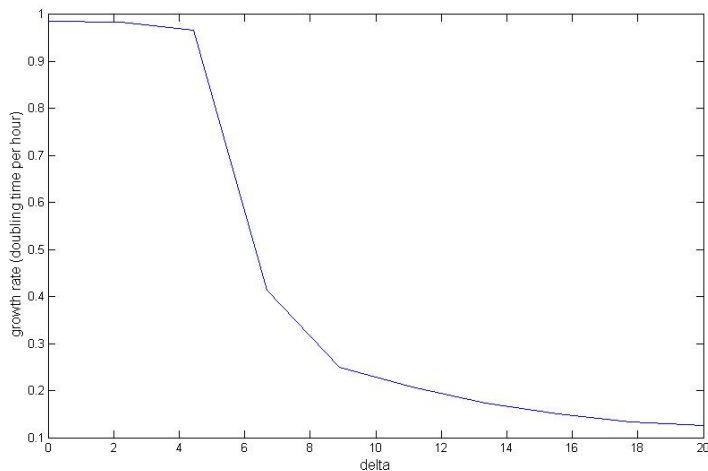
Figure 2: Optimal growth rate as a function of δ

For the following computational experiments, we let $\delta = 3$ as a default value, which means that the probabilistic constraint becomes $P(M^- \leq S_i v \leq M^+) \geq .997$.

3.5.3 Dependence upon scenario construction

Each growth coefficient in the stoichiometric matrix is treated as a normally distributed random variable with mean μ and standard deviation σ, where μ is the value of the coefficient as given by the original model and σ is its first non-stated decimal place.

The scenarios for the uncertain coefficient s are created by discretizing the normal distribution of the coefficient. For instance, we first compute the interval of possible values of s by considering values that lie within the 3 standard deviations from the mean: $[\mu - 3\sigma, \mu + 3\sigma]$. In order to discretize this interval, we sample n points uniformly distributed across the interval, where n is the number of scenarios.

Increasing the number of scenarios gives a better approximation of the distribution of possible s values. Setting the number of scenarios too high, however, increases the size of the problem, thus increases its computational time, significantly. We therefore seek an appropriate number of scenarios that approximates the distribution of s well, but small enough to keep the computational time relatively short.

The results are forthcoming.

3.6 Flux Variability

It may be interesting or useful to know what possible range of values a particular reaction rate may take under specified cellular conditions. The Linear and Robust FBA models can easily be adapted to compute these ranges. Our adaptation computes flux ranges in the following way:

⋄ Fix the growth rate at its true value. Accomplished by maximizing growth rate in the linear model and then setting the upper and lower bounds on the growth reaction equal to this optimal value.

⋄ In the robust case, fix some M, typically the minimal or box M that yields the true growth rate.

⋄ Construct c such that the objective function is the value of the flux of interest. The variable, c, should have a nonzero entry corresponding to the location of the reaction of interest in the solver-formatted ordering of reactions. This entry should be positive if the flux is to be minimized, and negative if it is to be maximized (so that the set of negatives is minimized). Sometimes in the course of formatting for the solver it is necessary to negate a flux value; we compensate by negating c if this happens to the flux of interest. Sometimes in formatting for the solver a flux is shifted by some constant, but this has no effect on the computation of the maximum and minimum so long as the optimal value is shifted back by the correct amount.

By maximizing and minimizing each flux value across a model's entire set of reactions, we obtain a profile of possible reaction rates for the cell. Such a result may look like this:
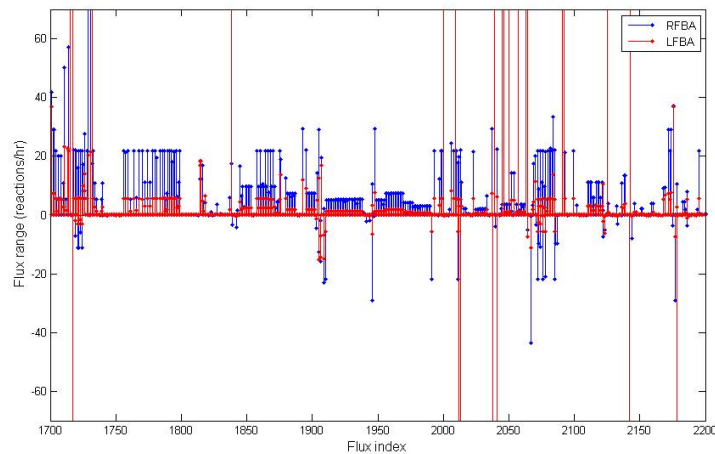
Figure 3: Range of possible flux values

3.6.1 Degeneracy

The flux ranges obtained in the linear and robust models are typically consistent, but some differences in results are notable.

Flux ranges may be classified into two groups: some fluxes are essentially unbounded, limited only by the L and U bounds imposed in the model. Such a result is biologically unrealistic and thus undesirable, failing to give meaningful information about possible flux values.

23

The existence of multiple sets of fluxes yielding the optimal growth rate is called the problem of degeneracy, and the existence of unbounded fluxes is its worst manifestation. We find that the robust model performs better than the linear model in computing flux ranges in the respect that it permits fewer unbounded fluxes, sometimes significantly fewer. In the largest and most accurate E. *coli* model, the linear model allows 108 unrestricted fluxes while the robust model allows only 67, a substantial decrease.

The other flux category is those that are bounded, restricted to falling within some fixed range. In the case of these fluxes, the linear model allows smaller ranges than the robust model, so that degeneracy is greater in the robust model than in the linear in this respect. However, it is debatable whether such degeneracy is bad: given the complexity of metabolic networks, it is likely that cells can achieve optimality through activation of different subsets of the metabolic network. The mathematical existence of many flux sets achieving optimality may reflect that biologically cells may achieve the optimal growth rate in many different ways.

It should be noted that robust flux ranges were computed using a box M constraint, and that some initial tests using a minimal M constraint showed decreased degeneracy in flux ranges. We also note that linear flux ranges are typically contained within the robust ranges, so that the two models are consistent with one another.

3.6.2   Experimental Validation

Although most metabolic flux values cannot presently be determined by experiment (hence the need for models such as FBA), wet-lab data do exist for a small set of reactions. A validation of RAMP would be incomplete without a comparison of these experimental values to the flux ranges allowed in RAMP. We found that the experimental values always lie within the ranges permitted by RAMP. However, this is not surprising because RAMP allows such wide ranges in flux values.

The following table summarizes the data analyzed and corresponding flux ranges permitted by the linear and robust models. All reactions rates were normalized to the glucose uptake rate, with a normalization factor of approximately 11.3.

3.7   Gene Knockout Simulation

3.7.1   Motivation

The term gene knockout refers to the removal of a gene from a species and the consequent loss of functions controlled by that gene. Organisms which have lost a gene are called mutants, while the original species is referred to as the wild-type. Mutants may arise naturally, or may be created in the lab, where researchers are capable of removing a specific gene from an organism.

It is often desirable to know the effect the loss of a particular gene would have on a species. This is particularly the case in drug development, where researchers can create drugs designed to remove essential genes from diseased cells, such as cancerous cells.

However, because of the great number of genes in a cell it is time-consuming and expensive to experimentally determine which gene-knockouts are lethal to a species and which

Reaction	iAF index	exp. flux value	linear range	robust range
GLC + ATP→G6P	1376	100 ± 1.2	0 to 132	0 to 317
G6P→6PG + NADPH	1165	29.5 ± 1.3	0 to 190	0 to 276
G6P→F6P	1919	70.1 ± 1.5	-170 to 141	-141 to 330
PEP→PYR+ATP	2037, 2099	119.7 ± 2.9	-64 to 157	-243 to 378
PYR→ACCOA + CO2 + NADH	1890	102.6 ± 2.9	0 to 155	0 to 216
OAA + ACCOA→ICT	482	159.7 ± 2.4	9 to 127	7 to 284
ICT→OGA + CO2 + NADPH	1418	24.2 ± 3.1	9 to 87	7 to 129
FUM→MAL	1130	13.8 ± 3.1	9 to 271	-48 to 575
MAL→OAA + NADH	1622	10.7 ± 2.4	-22 to 255	-291 to 167
MAL→PYR + CO2 + NADH	1626	3.1 ± 1.8	0 to 55	0 to 192
OAA + ATP→PEP + CO2	2022	0.2 ± 0.3	0 to 63	0 to 248
PEP + CO2→OAA	2020	27.1 ± 1.9	0 to 94	0 to 277
ACCOA→AC + ATP	195	59.7 ± 2.4	0 to ∞	0 to ∞
NADPH→NADH	1725, 2222	-52.5 ± 9.3	-155 to 264	-508 to 568
ETH→ETHxt	700	0.0 ± 1.0	0 to 8	0 to 34

Table 1: Experimental and Computed Flux Data

are not. FBA provides a computational alternative. Each gene is associated with some number of metabolic reactions which are lost under removal of that gene. These reactions can be deleted from the FBA model and a new optimal growth rate computed under these conditions. If the ratio of mutant to wild-type growth rates falls below a certain threshold, the gene knockouts is considered lethal.

3.7.2 Robust FBA Method

Implementation of a gene knockouts using FBA or RFBA simply involves fixing the flux values of affected reactions at zero, accomplished by adjusting the bounds L and U, and then optimizing growth rate.

Gene knockouts were simulated on three E. *coli* models called iJR904, iAF1260, and iJO1366. The computational predictions for gene essentiality were then compared to the experimentally derived gene knock-out results in studies by Joyce [4], Gerdes [2], and Orth [3].

Each model's environment was tuned to mimic the growth media used in these studies by adjusting the bounds on the intake reaction rates of the carbon source, oxygen, and other necessary nutrients. For nutrient-rich growth media, the cell was allowed free exchanges of all possible nutrients. For minimal growth media, the cell was only allowed to intake its carbon source, NH4, SO4, O2, Pi, H+, H2O, and CO2.

The gene was classified as essential if the ratio of the mutant to wild-type growth rates fell below 0.5.

3.7.2.1 Results The experimental data for the essentialness of genes were obtained from studies by Joyce, et al, where in vivo gene-knock outs were conducted in glycerol-supplemented minimal medium. Our computational gene knockout studies were carried out in a slightly altered cell environment; the cell was allowed to intake potassium and iron in

addition to all nutrients available in a minimal growth medium, and glucose as its carbon source.

The following tables show the the comparisons between experimental and model predictions of gene essentiality of genes in iJR904.

FBA Results:

	experimental essential	experimental non-essential
predicted essential	78	100
predicted non-essential	31	695

RFBA Results:

	experimental essential	experimental non-essential
predicted essential	63	48
predicted non-essential	46	747

RFBA showed an increased accuracy rate compared to FBA in predicting the essentialness of the genes. Although FBA predicted a higher number of correct essential genes, its accuracy rate (44 percent) was markedly lower than that of RFBA (57 percent).

Increasing the threshold value for determining the lethality of genes to .7 produced a higher number of correct essential genes determined by RFBA (63 to 75), but also increased the number false positives by a proportional amount (48 to 58), thus leaving the accuracy rate relatively stable around 57 percent. Similarly, decreasing the threshold value to .3 decreased the number of correct essential genes as well as false positives, and produced only a marginal improvement on the accuracy rate (57 percent to 58 percent).

Gene knockout simulations with models iAF1260 and iJO1366 are still work in progress.

3.7.3 Robust MOMA Method (RMOMA)

3.7.3.1 MOMA: an Alternative Approach Minimization of Metabolic Adjustment (MOMA) is an alternative method of simulating gene knockouts. Unlike FBA-based gene knockout simulations, which assume that mutant cells achieve an optimal growth rate, MOMA hypothesizes that mutant cells minimize changes in flux state in response to the gene deletion. [6] Specifically, MOMA minimizes the Euclidean distance between the flux values of the wild-type cell and the mutant cell. More precisely, the goal is to minimize $||v' - v||$, where v is any optimal flux vector of the wild-type cell satisfying the steady-state conditions and environmental bounds, and v' is any feasible flux vector for the mutant, meaning it satisfies the typical steady-state constraint subject to new bounds L' and U' which fix the lost reactions at zero flux. The growth flux of the mutant is then taken to be the growth flux of the new flux vector v' which yields optimality in the MOMA minimization problem. The lethality of the gene knockout is then defined by some threshold below which the ratio of mutant to wild-type growth rate must fall, just as under the FBA method.

3.7.3.2 Implementation MOMA can be formulated as an SOCP in the following way:

⋄ MOMA seeks to minimize the norm of a vector, just as in the minimization of M discussed previously. We again introduce a new variable ϕ, and impose the conic constraint $\phi \geq ||\Delta v||$. Minimizing ϕ is then equivalent to minimizing $||\Delta v||$.

◇ The unknown vector is consequently defined to be the vector
$$x = (v, s, z, v', s', z', \phi, \Delta v).$$

◇ Recall that the original RFBA program constructed an SOCP in two steps, first by constructing a system of equations $Ax = b$ which imposed the steady-state conditions, and then by manipulating variables and imposing cone constraints such that the condition $L \le v \le U$ was satisfied by any v feasible in the SOCP. To impose the steady-state conditions on v and v', we form the matrices A and b that impose this constraint and begin with the system

$$\left[\begin{array}{c|c} A & 0 \\ \hline 0 & A \end{array} \right] \left[\begin{array}{c} v \\ s \\ z \\ v' \\ s' \\ z' \end{array} \right] = \left[\begin{array}{c} b \\ b \end{array} \right].$$

◇ It will later be advantageous to have the unknowns v, v' located consecutively within x. By suitably rearranging columns of the above system we obtain this reordering of variables.

◇ We then expand x to include the variables $\phi, \Delta v$, and enforce the equation $\Delta v = v' - v$ by adding the following rows to the system:

$$\left[\begin{array}{c|c|c|c} Id & -Id & 0 & -Id \end{array} \right] \left[\begin{array}{c} v \\ \hline v' \\ \hline s \\ s' \\ z \\ z' \\ \phi \\ \hline \Delta v \end{array} \right] = \left[\begin{array}{c} 0 \end{array} \right].$$

◇ Since v and v' are adjacent, we can construct lower and upper bound vectors on the joint flux vector $[v; v']$ by concatenating L with L' and U with U', setting fixing the growth rate in the bounds on v and setting deleted reactions to 0 in the bounds on v'. With a single flux vector a single vectors of upper and lower bounds, we can then format the system in the normal way for the solver.

3.7.3.3   Computational Difficulties   We have confirmed that RMOMA has been implemented correctly and yields reasonable results through testing on the E. *coli* core model. However, the system of equations on which RMOMA is based is more than double the size of a typical

3.8   Wild-Type Modeling

3.8.1   RFBA and Non-optimized Cultures

The traditional FBA model suffers from two limiting assumptions: that cells exist in steady-state, and that cells are identical across a culture, particularly with regard to biomass

27

composition (reflected in growth coefficients). While these assumptions represent an over-simplification of biological reality in any situation, one can conceive of a cellular culture as converging towards such a state over lengthy periods of time if it remains in a stable environment with a steady supply of the same nutrients. Indeed, while the improvements shown by RAMP support the notion that LFBA's assumptions are faulty, the successes of linear FBA (LFBA) suggest that the steady-state and uniformity assumptions are not gross misrepresentations of optimized cultures in a stable environment, such as a labratory.

However, when one turns from a stable cellular environment to the ever-changing environments in nature, the LFBA assumptions no longer represent a reasonable approximation of the biological situation. Cells in nature, referred to as wild-type cells, exhibit greater variability across a culture and can be expected deviate farther from steady-state as they adjust their flux state and metabolite concentrations to adapt to changes in nutrients available. While LFBA cannot possibly model such situations, RAMP may be capable of modeling non-optimized wild-type cultures. Parameters representing deviation from steady state (M) and variability across a culture (σ in the distribution of S_i) can be increased to represent the fact that wild-type cultures do not exist near the steady-state, uniform condition imposed by LFBA.

This section is devoted to discussing preliminary attempts to model wild-type cells, their results, and future possibilities in the use of RAMP to study non-optimized cultures.

3.8.2   Preliminary Attempts

Preliminary attempts have centered upon isolating possible values for the parameters M (departure from steady-state) and σ (variability of growth coefficients) over time as a wild-type culture adapts to a stable environment and approaches optimality. We expect M to decrease and σ to increase over time in such a way that the growth rate increases towards LFBA optimal value. Based upon experimental data from [5], we know the growth rate that *E. coli* should have at each point in time as it approaches optimality over a 40-day period. The goal is to then back-calculate an (M, σ) pair for each point in time to infer information about steady-state deviation and growth coefficient variability as the culture adapts.

The most significant hurdle in inferring such information has been degeneracy in possible (M, σ) pairs for any given growth rate. In fact, for a given growth rate, we can calculate for *any* σ a minimal M such that the growth rate is achieved. This process simply involves constructing the scenario sets of the uncertain rows S^u according to a given σ and then executing the minimal M function. The result of executing this process for a fixed growth rate over a range of σ values is a linear relationship between $||M||$ and σ.

Since an infinite number of possible (M, σ) pairs can be computed for each growth rate, we sought additional constraints to isolate realistic pairs. An initial idea was this: we expect that as the cellular culture progresses from one growth rate to another, changes in M and σ are continuous. Consequently, (M, σ) pairs for consecutive, sufficiently close growth rates should be close to each other, i.e. near the intersection of the M-vs.-σ graphs for the two growth rates. Taking each consecutive pair of growth rates and locating the point of intersection of their M-vs.-σ graphs, we hoped to identify a progression of (M, σ) points along which the culture must progress over time in order to adjust steady-state deviation and growth coefficient variance continuously as it adapts to a stable environment.

While theoretically promising, this approach did not yield realistic results. Taking a least-squares linearization of the M-vs.-σ relation for each growth rate and locating points of intersection, we found that the intersections points were all very near the origin and in no orderly progression. Further consideration suggests these lines should in fact not intersect, and that their apparent intersection near the origin was the result of computational error and rounding. For any two growth rates $g_1 < g_2$ and a fixed σ, a greater M (greater steady-state deviation) should always be required to achieve the higher growth rate. In other words, $M(g_1, \sigma) \leq M(g_2, \sigma)$ for every σ, so that the M-vs.-σ graph for g_1 should never cross the M-vs.-σ graph for g_2. Note that this does not imply that (M, σ) pairs must change discontinuously with growth rate; a small increase in growth rate could produce only a slight shift of the M-vs.-σ graph upward, so that the new (M, σ) pair moves only slightly from the old without the two lines needing to intersect.

In another approach, we sought experimental biological data that might limit possible (M, σ) pairs. [5] contained experimental data about glucose and oxygen uptake rates for the culture over time. We speculated that of the set of (M, σ) pairs for each growth rate, only some bounded subset of pairs might be consistent with the known glucose and oxygen uptake rates. Selecting these subsets for each growth rate and connecting them in a reasonable way might then yield the desired path of (M, σ) through time.

However, this approach also met with difficulty. We determined consistency of a given (M, σ) pair with experimental data by maximizing and minimizing the glucose and oxygen uptake rates subject to a maximal steady-state deviation of M and growth coefficient standard deviation of σ. (M, σ) pairs for M and σ nearly zero were consistent with the experimental data, and as M and σ increased, the range of theoretically possible uptake rates did not shift in one direction or the other, but expanded outward in both directions, remaining consistent with the data. The wide ranges on flux values permitted by the robust model and the expanding rather than shifting of these ranges with increases in M and σ combined to make all (M, σ) pairs consistent with experimental data, so this method also failed to isolate probable (M, σ) values.

In addition to these computational difficulties, we encountered broader, biological questions about how a wild-type culture should be modeled. One persistent question was this: what aspect of wild-type cells prevents them from growing optimally when not acclimated to a stable environment? We certainly expect steady-state deviation to be greater for a wild-type culture in a changing environment, and this has the effect of enabling increased growth rates. If σ is the only other parameter being varied, it must then increase to account for the diminished growth rate, representing greater variability of the growth culture.

While increasing σ has the effect of reducing growth rate in RAMP because RAMP optimizes over some high percentage of growth coefficient scenarios, a change in σ does not actually represent an overall decrease in growth efficiency in the culture: increasing σ represents greater deviation of the growth coefficients in both directions, both the direction unfavorable toward growth and the direction favorable toward it, with the result that some portion of the culture is represented as having more ideal growth coefficients than any eventually will in the optimized culture. A more accurate representation of the wild-type growth coefficients should therefore involve a change in the average growth coefficients as well as growth coefficient variability to avoid this misrepresentation. However, computational results may be unaffected because RAMP optimizes over the least ideal growth coefficients in either case.

Another approach, however, might say that wild-type cultures fail to reach optimality not because of less favorable growth coefficients within the culture (reflecting a less ideal biomass composition), but because cells fail to reach the maximum growth rate possible because it takes time to express the genes and create the enzymes necessary to adjust their flux state the ideal flux state when they enter a new environment. Such a scenario would represent a greater modeling challenge for RAMP, since RAMP is inherently an optimization model.

However, we have developed a method to represent in RAMP the gradual progression of a wild-type culture toward an optimal growth rate. We represent time in a discrete series of points and assume that sub-optimal cells increase growth rate as quickly as possible over time, but are limited by some maximum rate at which they can adjust their flux state in time. This method thus optimizes growth rate at each point in time, but subject to some maximum deviation from the flux state of the previous point in time. It begins with some sub-optimal flux vector v_0 which yields a growth rate equal to the experimentally-known initial growth rate of the wild-type culture. Between any two points in time t_j and t_{j+1} it then solves the following optimization problem:

$$\max\left\{(v_{\text{growth}})_{t_{j+1}} : -M_i^- \leq S_i v_{j+1} \leq M_i^+ \, \forall \, S_i \in \mathcal{U}_i, L \leq v_{j+1} \leq U, ||v_{j+1} - v_j|| \leq \phi\right\}$$

for some constant ϕ representing the maximum change in flux state between two points in time.

The result of this program is a function giving growth rate over time. In the typical result, growth rate increases quickly at first and levels off gradually toward the ultimate optimal growth rate, as expected. The length of time is affected by ϕ, but it does not affect its overall qualitative shape.

Several disadvantages to this approach should be mentioned. First, it currently treats all other modeling parameters as constant. This could potentially be fixed, for example by making M some suitable function of time. The greater problem is that this approach assumes information about parameters such as M, and then infers information about growth. However, since growth information is already known about a culture over time, it would be more ideal to have some method of inferring information about other modeling parameters given information about growth over time.

To conclude, our work indicates that RAMP is capable of representing non-optimized wild-type cultures, but that additional biological information is necessary before an accurate model can be implemented. Succinctly put, many combinations of different parameters can be used in RAMP to induce the expected wild-type behavior, and so only with greater biological information might it be possible to determine which parameters should be used to account for this behavior and to then make specific quantitative predictions about the behavior of those parameters over time.

3.9   An Alternative Computing Approach

3.9.1   Motivation

Semidefinite and quadratic program solvers take problems of the specific form

$$\min\left\{c^T x : Ax = b\right\}$$

where the components of x are free or restricted to positive or quadratic cones. Although the Robust FBA optimization problem is not naturally in this form, we have explained how it can be expressed in the canonical form of a second-order cone program above and solved by SeDuMi and similar programs. However, this process involves the introduction of many slack and surplus variables and an expansion of the stoichiometric matrix into a new matrix many times its size. These changes required to express our problem in the standard primal form of an SOCP have had a significant effect on computational time and complexity, so that many programs take hours or days to solve.

The original form of the Robust FBA optimization problem resembles much more closely the dual of a canonical SOCP. A more efficient computational method than converting it to a primal problem would instead consist of leaving it as a dual problem, finding the primal problem to which it dual, and giving that primal problem to the solver. Since the primal and dual of an SOCP have the same solution, the same optimal value will be computed. The solution vector v to our (dual) problem will consist of the dual variables given by the solver.

3.9.2 Implementation

The original form of the RFBA/ SOCP problem is the following:

$$\max \left\{ d^T v : M^{c-} \leq S^c v \leq M^{c+}, M_j^{u-} + ||R_j v|| \leq S_j^u v \leq M_j^{u+} \, \forall \, j, L \leq v \leq U \right\}$$

where we have used d rather than c to avoid future confusion.

The canonical forms of a primal SOCP and its dual are the following:

$$\begin{array}{cc} \text{primal:} & \text{dual:} \\ \min \left\{ \sum_i c_i^T x_i : \sum_i A_i x_i = b, x_i \in K_i \, \forall \, i \right\}, & \max \left\{ b^T y : c_i - A_i^T y \in K_i \, \forall i \right\}. \end{array}$$

Our goal is to express RFBA in the canonical form of a dual SOCP, find the corresponding primal problem, and format it for solvers such as SeDuMi.

We first express RFBA as an SOCP. It is evident that d plays the role of b, and that the unknown y in the dual problem is the flux vector v in RFBA. It remains to express the RFBA constraints as cone constraints of the form $c_i - A_i^T y \in K_i$. For the constraints on S^c and the flux bounds L and U, observe the following:

$$
\begin{aligned}
M^{c-} \leq S^c v \leq M^{c+}, L \leq v \leq U \iff & \ M^{c-} \leq S^c v \leq M^{c+}, L \leq Id \cdot v \leq U \\
\iff & \ \begin{bmatrix} M^{c-} \\ L \end{bmatrix} \leq \begin{bmatrix} S^c \\ Id \end{bmatrix} v \leq \begin{bmatrix} M^{c+} \\ U \end{bmatrix} \\
\iff & \ \begin{cases} \begin{bmatrix} M^{c+} \\ U \end{bmatrix} - \begin{bmatrix} S^c \\ Id \end{bmatrix} v \in \mathbb{R}^n_+, \\ \begin{bmatrix} -M^{c-} \\ -L \end{bmatrix} - \begin{bmatrix} -S^c \\ -Id \end{bmatrix} v \in \mathbb{R}^n_+. \end{cases}
\end{aligned}
$$

Similarly, we express the constraints on each row of S^u as follows:

$$\begin{cases} ||R_j v|| \le M_j^{u+} - \bar{S}_j^u v \; \forall \, j, \\ ||R_j v|| \le -M_j^{u-} + \bar{S}_j^u v \; \forall \, j \end{cases} \iff \begin{cases} (M_j^{u+} - \bar{S}_j^u v, R_j v) \in K_q \; \forall \, j, \\ (-M_j^{u-} + \bar{S}_j^u v, R_j v) \in K_q \; \forall \, j \end{cases}$$

$$\iff \begin{cases} \left[\begin{array}{c} M_j^{u+} \\ 0 \end{array} \right] - \left[\begin{array}{c} \bar{S}_j^u \\ R_j \end{array} \right] v \in K_q \; \forall \, j, \\ \left[\begin{array}{c} -M_j^{u-} \\ 0 \end{array} \right] - \left[\begin{array}{c} -\bar{S}_j^u \\ -R_j \end{array} \right] v \in K_q \; \forall \, j. \end{cases}$$

RFBA now is the following optimization problem, in canonical dual form:

$$\max d^T v, \text{ subject to:}$$

$$\left[\begin{array}{c} M^{c+} \\ U \end{array} \right] - \left[\begin{array}{c} S^c \\ Id \end{array} \right] v \in \mathbb{R}_+^n,$$

$$\left[\begin{array}{c} -M^{c-} \\ -L \end{array} \right] - \left[\begin{array}{c} -S^c \\ -Id \end{array} \right] v \in \mathbb{R}_+^n.$$

$$\left[\begin{array}{c} M_j^{u+} \\ 0 \end{array} \right] - \left[\begin{array}{c} \bar{S}_j^u \\ R_j \end{array} \right] v \in K_q \; \forall \, j,$$

$$\left[\begin{array}{c} -M_j^{u-} \\ 0 \end{array} \right] - \left[\begin{array}{c} -\bar{S}_j^u \\ -R_j \end{array} \right] v \in K_q \; \forall \, j.$$

A comparison of the canonical primal and dual forms shows that in the corresponding primal problem, each matrix by which v is multiplied in the above cone constraints becomes the transpose of a matrix A_i in the primal constraint $\sum_i A_i x_i = d$, where the new primal unknown x_i must lie in the cone of the corresponding dual constraint. Moreover, each vector in the above cone constraints becomes a vector c_i in the primal objective function $\sum_i c_i x_i$. The primal problem to which RFBA corresponds is thus:

$$\min \left\{ \left[\begin{array}{c} M^{c+} \\ U \end{array} \right]^T x_1 + \left[\begin{array}{c} -M^{c-} \\ -L \end{array} \right]^T x_2 + \sum_j \left(\left[\begin{array}{c} M_j^{u+} \\ 0 \end{array} \right]^T x_{2j+1} + \left[\begin{array}{c} -M_j^{u-} \\ 0 \end{array} \right]^T x_{2j+2} \right) \right\},$$

subject to:

$$\left[\begin{array}{c} S^c \\ Id \end{array} \right]^T x_1 + \left[\begin{array}{c} -S^c \\ -Id \end{array} \right]^T x_2 + \sum_j \left(\left[\begin{array}{c} \bar{S}_j^u \\ R_j \end{array} \right]^T x_{2j+1} + \left[\begin{array}{c} -\bar{S}_j^u \\ -R_j \end{array} \right]^T x_{2j+2} \right) = d,$$

$$x_1, x_2 \in \mathbb{R}_+^n, x_3, \ldots, x_n \in K_q.$$

Since solvers such as SeDuMi require one vector c, one matrix A, and one unknown x, we give the solver the equivalent problem $\min \left\{ c^T x, Ax = b, x_i \in K_i \right\}$ where

$$c = \left[(M^{c+})^T, \; U^T, \; (-M^{c-})^T, \; -L, \; M_1^{u+}, \; 0, \; -M_1^{u-}, \; 0, \; \ldots, \; M_j^{u+}, \; 0, \; -M_j^{u-}, \; 0, \; \ldots \right]$$

$$A = \left[(S^c)^T, \; Id, \; (-S^c)^T, \; -Id, \; (\bar{S}_1^u)^T, \; R_1^T, \; (-\bar{S}_1^u)^T, \; -R_1^T, \; \ldots, \; (\bar{S}_j^u)^T, \; R_j^T, \; (-\bar{S}_j^u)^T, \; -R_j^T, \; \ldots \right]$$

$$x = \left[x_1^T, \; x_2^T, \; \ldots, \; x_i, \; \ldots, x_n \right]^T, \text{ each sub-vector } x_i \text{ in its cone as given above.}$$

3.10 Modularity and Other Alternative Forms

To increase coding flexibility, we divided Robustify into two separate functions. We created a function *dual2solver* which takes the A_i, c_i, and d matrices for an SOCP in canonical dual form and formulates the corresponding primal problem for solvers. Robustify now consists of a code which creates these matrices as detailed above and then calls *dual2solver*. The advantage of modularizing code in this way is that the *dual2solve* function has made it easy to create alternate forms of the Minimal M and RMOMA programs, which can also be expressed more simply as dual rather than primal SOCPs.

4 Theoretical Results

4.1 Stability Analysis

The robust FBA model maximizes growth flux over a set of feasible flux vectors v. The flux vector, v, is defined to be feasible if it satisfies the constraint $Av \leq b$ for all vectors A in an uncertainty set $\mathcal{U} = \{\bar{A} + u^T R : u^T u \leq 1\}$ determined by some vector $\bar{A}$ and some matrix R.

The choice of parameters $\bar{A}$ and R defining $\mathcal{U}$ is somewhat arbitrary, and therefore it is desirable to assure that feasible v are stable under sufficiently small perturbations in $\bar{A}$ and R.

We prove the following:

Theorem. Let $\mathcal{U} = \{\bar{A} + u^T R : u^T u \leq 1\}$ and let v be a vector such that $Av \leq b$ for every $A \in \mathcal{U}$, where $b > 0$ is a constant. Let $\mathcal{U}' = \{\bar{A}' + u^T R' : u^T u \leq 1\}$ be a second uncertainty set defined by $\bar{A}' = \bar{A} + \Delta A$ and $R' = R + \Delta R$. Then for any given $\epsilon > 0$, there exists $\delta > 0$ such that if $\left|\left|\Delta \bar{A}\right|\right| + \left|\left|\Delta R\right|\right| < \delta$, then there exists v' satisfying $A'v' \leq b$ for all $A' \in \mathcal{U}'$ and $\left|\left|v' - v\right|\right| < \epsilon$.

We make use of the following Lemma:

Lemma. $Av \leq b \ \forall \ A \in \mathcal{U}$ if and only if $\bar{A}v + ||Rv|| \leq b$.

Proof of Lemma.

$$
\begin{aligned}
Av \leq b \text{ for all } A \in \mathcal{U} \iff \max\{Av : A \in \mathcal{U}\} &\leq b \\
&= \max\left\{\left(\bar{A} + u^T R\right)v : u^T u \leq 1\right\} \\
&= \bar{A}v + \max\left\{u^T Rv : u^T u \leq 1\right\} \\
&= \bar{A}v + ||Rv||.
\end{aligned}
$$

Therefore $Av \leq b$ for all $A \in \mathcal{U} \iff \bar{A}v + ||Rv|| \leq b$. $\qquad\square$

Proof of Theorem: Let $\epsilon > 0$. We will find $c \in (0, 1)$ so that $v' = cv$ satisfies $A'v' \leq b$ for all $A' \in \mathcal{U}'$ and $\left|\left|v' - v\right|\right| < \epsilon$ when $\left|\left|\Delta \bar{A}\right|\right| + \left|\left|\Delta R\right|\right| < \delta$ for some $\delta > 0$.

We show that $\max\{A'(cv) : A' \in \mathcal{U}'\} \leq b$ by showing $\max\{A'(cv) : A' \in \mathcal{U}'\} \leq \max\{Av : A \in \mathcal{U}\}$. Note that this inequality is equivalent to the following:

$$
\begin{aligned}
\max\{A'(cv) : A' \in \mathcal{U}'\} &\leq \max\{Av : A \in \mathcal{U}\} \\
\iff \bar{A}'cv + ||R'cv|| &\leq \bar{A}v + ||Rv|| \\
\iff \left(\bar{A} + \Delta\bar{A}\right)cv + ||(R + \Delta R)cv|| - \bar{A}v - ||Rv|| &\leq 0.
\end{aligned}
$$

The expression on the left side of the inequality can be bounded above as follows:

$$\left(\bar{A} + \Delta\bar{A}\right) cv + ||(R + \Delta R)\, cv|| - \bar{A}v - ||Rv||$$
$$\leq \bar{A}cv + \left|\left|\Delta\bar{A}cv\right|\right| + ||Rcv|| + ||\Delta Rcv|| - \bar{A}v - ||Rv||$$
$$\leq c\bar{A}v + c\left|\left|\Delta\bar{A}\right|\right| ||v|| + c\, ||Rv|| + c\, ||\Delta R||\, ||v|| - \bar{A}v - ||Rv||$$
$$\text{(by the sub-multiplicity of the 2-norm)}$$
$$\leq c\left(\bar{A}v + \left|\left|\Delta\bar{A}\right|\right| ||v|| + ||Rv|| + ||\Delta R||\, ||v||\right) - \left(\bar{A}v + ||Rv||\right).$$

Therefore, if

$$c \leq \frac{\bar{A}v + ||Rv||}{\bar{A}v + ||Rv|| + \left(\left|\left|\Delta\bar{A}\right|\right| + ||\Delta R||\right) ||v||}$$

then

$$\max\left\{A'(cv) : A' \in \mathcal{U}'\right\} \leq b.$$

Choose

$$v' = cv = \left(\frac{\bar{A}v + ||Rv||}{\bar{A}v + ||Rv|| + \left(\left|\left|\Delta\bar{A}\right|\right| + ||\Delta R||\right) ||v||}\right) v.$$

Now we show that if $\left|\left|\Delta\bar{A}\right|\right| + ||\Delta R||$ is sufficiently small then $||v' - v|| = ||cv - v|| = (1 - c)\, ||v|| < \epsilon$, which is true if and only if $1 - c < \frac{\epsilon}{||v||}$. Note that the function $f(x) = \frac{k}{k + x||v||}$ (k constant) satisfies $\lim_{x \to 0} f(x) = 1$. Therefore if $k = \bar{A}v + ||Rv||$, $\exists\, \delta > 0$ such that if $\left|\left|\Delta\bar{A}\right|\right| + ||\Delta R|| < \delta$, then $\left|f\left(\left|\left|\Delta\bar{A}\right|\right| + ||\Delta R||\right) - 1\right| = 1 - c < \frac{\epsilon}{||v||}$, which implies $||v' - v|| < \epsilon$. $\qquad\square$

4.2 Extensions and Corollaries of Stability Analysis

The actual FBA model requires not just that $Av \leq b$ for all A in a single uncertainty set $\mathcal{U}$, but that $A_i v \leq b$ for all $A_i \in \mathcal{U}_i$ where $\left\{\mathcal{U}_i = \left\{\bar{A}_i + u^T R_i : u^T u \leq 1\right\} : i = 1, 2, \ldots, m\right\}$ is a collection of uncertainty sets. To show that the model is stable, it is therefore necessary to extend the theorem with the following corollary:

Corollary. Under a perturbation of each $\bar{A}_i$ and R_i, a single v' can be constructed such that $A_i' v' \leq b\ \forall\ A_i' \in \mathcal{U}_i'$ for each $i = 1, 2, \ldots, k$.

Proof. Let $\epsilon > 0$ be given. We must find v' and $\delta > 0$ so that if $\left|\left|\Delta\bar{A}_i'\right|\right| + ||\Delta R_i'|| < \delta\ \forall\ i$, then $A_i' v' \leq b\ \forall\ A_i' \in \mathcal{U}_i'$ for each i, and $||v' - v|| < \epsilon$.

By the theorem proven above, for each i there exist c_i and δ_i such that $||c_i v - v|| < \epsilon$ and $A_i' c_i v \leq b\ \forall\ A_i' \in \mathcal{U}_i'$ provided $\left|\left|\Delta\bar{A}_i\right|\right| + ||\Delta R_i|| < \delta_i$. Choose $c = \min\left\{c_i : i = 1, 2, \ldots, m\right\}$, so that $c = c_k$ for some $1 \leq k \leq m$, and choose $\delta = \min\left\{\delta_i : i = 1, 2, \ldots, m\right\}$.

Now suppose that $\left|\left|\Delta\bar{A}_i'\right|\right| + ||\Delta R_i'|| < \delta$ for all i. Then $\left|\left|\Delta\bar{A}_k'\right|\right| + ||\Delta R_k'|| < \delta_k$ and so $||cv - v|| = ||c_k v - v|| < \epsilon$. Furthermore, for each $i = 1, 2, \ldots, m$, $\left|\left|\Delta\bar{A}_i'\right|\right| + ||\Delta R_i'|| < \delta_i$ and so $\max\left\{A_i' cv : A_i' \in \mathcal{U}_i'\right\} = c\left(\bar{A}_i' v + ||R_i' v||\right) \leq c_i\left(\bar{A}_i' v + ||R_i' v||\right) \leq b$. So $v' = cv$ satisfies the requisite properties. $\qquad\square$

It may sometimes be desirable to tighten the restriction upon $A'v'$ from the initial restriction on Av, so that $A'v' \leq b - \delta_b\ \forall\ A' \in \mathcal{U}'$ for some $\delta_b > 0$. The following corollary shows that the theorem remains true under sufficiently small perturbations in b.

Corollary. For any $\epsilon > 0$, there exist $\delta_{AR}, \delta_b > 0$ so that if $\left|\left|\Delta\bar{A}\right|\right| + \left|\left|\Delta R\right|\right| < \delta_{AR}$, $\exists\, v'$ such that $||v' - v|| < \epsilon$ and $A'v' \leq b - \delta_b\ \forall\ A' \in \mathcal{U}'$.

Proof. The reasoning follows that of the proof of the theorem.

$$A'v' \leq b - \delta_b \,\forall\, A' \in \mathcal{U}' \text{ if } \bar{A}'cv + ||R'cv|| - \bar{A}v - ||Rv|| + \delta_b \leq 0.$$

This expression is bounded above by

$$c\left(\bar{A}v + \left|\left|\Delta\bar{A}\right|\right| ||v|| + ||Rv|| + ||\Delta R|| \, ||v||\right) - \bar{A}v - ||Rv|| + \delta_b$$

and so the inequality is satisfied if

$$c\left(\bar{A}v + \left|\left|\Delta\bar{A}\right|\right| ||v|| + ||Rv|| + ||\Delta R|| \, ||v||\right) - \bar{A}v - ||Rv|| + \delta_b \leq 0,$$

which is true if

$$c = \frac{\bar{A}v + ||Rv|| - \delta_b}{\bar{A}v + ||Rv|| + \left(\left|\left|\Delta\bar{A}\right|\right| + ||\Delta R||\right) ||v||}.$$

Clearly, c can be chosen arbitrarily close to 1 by taking δ_b and $\left|\left|\Delta\bar{A}\right|\right| + ||\Delta R||$ sufficiently small, which in turn makes $v' = cv$ arbitrarily close to v. $\qquad\square$

The robust FBA model places not only an upper but also a lower bound on Av. We now extend the theorem to cover the constraint $-b \leq Av \leq b$.

Corollary. Suppose v satisfies $-b \leq Av \leq b$ for all $A \in \mathcal{U}$, and let $\epsilon > 0$. Then $\exists\, \delta > 0$ so that if $\left|\left|\Delta\bar{A}\right|\right| + ||\Delta R|| < \delta$, there exists v' such that $||v' - v|| < \epsilon$ and $-b \leq A'v' \leq b\ \forall\ A' \in \mathcal{U}'$.

Proof. By the theorem, there exist $c_1 \in (0,1)$ and $\delta_1 > 0$ such that $||c_1 v - v|| < \epsilon$ and $A'c_1v \leq b\ \forall\ A' \in \mathcal{U}'$ if $\left|\left|\Delta\bar{A}\right|\right| + ||\Delta R|| < \delta_1$. Note that $-b \leq Av \implies A(-v) \leq b$. So applying the theorem to $-v$, there exist $c_2 \in (0,1)$ and $\delta_2 > 0$ such that $||c_2 v - v|| < \epsilon$ and $A'(c_2(-v)) \leq b\ \forall\ A' \in \mathcal{U}' \iff -A'(c_2v) \leq b\ \forall\ A' \in \mathcal{U}' \iff -b \leq A'(c_2v)\ \forall\ A' \in \mathcal{U}'$ if $\left|\left|\Delta\bar{A}\right|\right| + ||\Delta R|| < \delta_2$.

Let $c = \min\{c_1, c_2\} = c_k$ for some k, and $\delta = \min\{\delta_1, \delta_2\}$. Then $\left|\left|\Delta\bar{A}\right|\right| + ||\Delta R|| < \delta$ implies $\left|\left|\Delta\bar{A}\right|\right| + ||\Delta R|| < \delta_k$, which means $||cv - v|| = ||c_k v - v|| < \epsilon$. Furthermore, $\left|\left|\Delta\bar{A}\right|\right| + ||\Delta R|| < \delta_1 \implies A'cv \leq A'c_1v \leq b$ if $A'v$ is nonnegative, and $A'cv < 0 < b$ if $A'v$ is negative. Likewise, $\left|\left|\Delta\bar{A}\right|\right| + ||\Delta R|| < \delta_2$, which implies $A'cv \geq A'c_2v \geq b$ if $A'v$ is negative, and $A'cv > 0 > -b$ if $A'v$ is positive. Therefore $-b \leq A'cv \leq b\ \forall\ A' \in \mathcal{U}'$. Hence $v' = cv$ satisfies the requisite properties. $\qquad\square$

4.3 Lower Semi-Continuity

4.3.1 Lower Semi-Continuity of Robust LP

We attempt to establish a Lower Semi-Continuity argument to validate the legitimacy of the new RAMP model. Lower Semi-Continuity guarantees that as we decrease uncertainty in RAMP, the RAMP solutions converge back to a solution to FBA. We prove Lower Semi-Continuity for a general Robust LP and remember that RAMP is an application of a Robust LP.

Suppose we have a primal Robust LP problem of the form $\min\{c^T x : Ax = a, B_i x \geq b_i \,\forall\, B_i \in \mathcal{U}_i, x \geq 0\}$ where $\mathcal{U}_i = \{\bar{B}_i + u^T R_i : u^T u \leq 1\}$. Notice that the constraints denoted with A's contain no uncertainty in the Robust problem while the constraints denoted with B's *are* given uncertainty in the Robust problem. We write the dual to the robust LP, $\max\{a^T \lambda + b_i^T \rho_0 : A^T \lambda + \bar{B}_i^T \rho_{0i} - \sum_i (R_i)^T \rho_i \leq c, \rho_{0i} \geq ||\rho_i||, \lambda \text{ free}\}$. We create a function f, a point-to-set map, that takes in some $(A, a, R, \bar{B}, b)$ and outputs a set containing items of the form $(x, \lambda, \rho_0, \rho)$ where x belongs the set $\text{argmin}\{c^T x : Ax = a, B_i x \geq b_i \,\forall\, B_i \in \mathcal{U}_i, x \geq 0\}$ and λ, ρ_0, and ρ are the corresponding allowable dual variables for a given x. The cost function c is a constant. We assume Slater's interior condition, a condition that guarantees we have an optimal solution to our problem, and we show that f is lower-semi continuous.

Theorem. Lower Semi-Continuity: If the sequence $(A^k, a^k, R^k, \bar{B}^k, b^k) \to (A^*, a^*, R^*, \bar{B}^*, b^*)$ and the sequence $(x^k, \lambda^k, \rho^k, \rho_0^k) \to (x^*, \lambda^*, \rho^*, \rho_0^*)$ where $(x^k, \lambda^k, \rho^k, \rho_0^k) \in f(A^k, a^k, R^k, \bar{B}^k, b^k)$, then $(x^*, \lambda^*, \rho^*, \rho_0^*) \in f(A^*, a^*, R^*, \bar{B}^*, b^*)$. In other words, if we know the parameters defining the Robust LP converge and the corresponding solutions to the Robust LP also converge, the limit of the solutions to the Robust LP is equal to the solution to the Robust LP defined by the limiting values of the original parameters.

Proof. For any $(A^k, a^k, R^k, \bar{B}^k, b^k)$ in the sequence, $(A^k, a^k, R^k, \bar{B}^k, b^k)$ is an allowable input to f by assumption, so $(A^k, a^k, R^k, \bar{B}^k, b^k)$ must satisfy primal feasibility, dual feasibility, and complementarity in the robust problem in order to produce an optimal output for f. By constructing the terms of these three conditions, $(A^k, a^k, R^k, \bar{B}^k, b^k)$ and $(x^k, \lambda^k, \rho^k, \rho_0^k) \in f(A^k, a^k, R^k, \bar{B}^k, b^k)$ must satisfy the following set of equations for all k:

$$
\begin{aligned}
A^k x^k &= a^k \\
||R^k x^k|| &\leq \bar{B}^k x^k - b^k \\
(A^k)^T \lambda^k + \bar{B^k}^T (\rho_0^i)^k - \sum_i (R^k)^T (\rho^i)^k &\leq c \\
(\rho_0^i)^k &\geq ||(\rho^i)^k|| \\
c^T x^k - (a^k)^T \lambda^k - (b^k)^T (\rho_0)^k &= 0.
\end{aligned}
\tag{3}
$$

Since the product of two convergent sequences converges to the product of the limits of the two sequences, and the same notion holds for the sum of two convergent sequences, we get that every equation and inequality in (3) holds for $A^*, a^*, R^*, \bar{B}^*, b^*, x^*, \lambda^*, \rho^*$, and ρ_0^*. For example, we look at the last equality, $c^T x^k - (a^k)^T \lambda^k - (b^k)^T (\rho_0)^k = 0$. The right hand side remains 0 for every k and so $\lim_{k \to \infty}(c^T x^k - (a^k)^T \lambda^k - (b^k)^T (\rho_0)^k) = 0$. We use properties of limits to break this up into the equation $\lim_{k \to \infty} c^T x^k - \lim_{k \to \infty}(a^k)^T \lambda^k - \lim_{k \to \infty}(b^k)^T (\rho_0)^k = 0$. We know that the product of the limits of convergent sequences is the is equal to the limit of the entire product, so $\lim_{k \to \infty}(b^k)^T (\rho_0)^k = (b^*)^T \rho_0^*$. This notion holds for every term, and we have $c^T x^* - (a^*)^T \lambda^* - (b^*)^T \rho_0^* = 0$. The same argument holds for inequalities. The norm is a continuous operator so it also does not skew limiting values. Every equation holds when we plug in $A^*, a^*, R^*, \bar{B}^*, b^*, x^*, \lambda^*, \rho^*$, and ρ_0^*. Thus by the definition of the function f, $(x^*, \lambda^*, \rho^*, \rho_0^*) \in f(A^*, a^*, R^*, \bar{B}^*, b^*)$ and f is lower semi-continuous.

4.3.2 Robust Convergence to a Linear Optimal Solution

Observe the primal Linear Programming problem $\min\{c^T x : \hat{A}x = \hat{a}, \hat{B}x \geq \hat{b}, x \geq 0\}$ with a dual, $\max\{\hat{a}^T y_1 + \hat{b}^T y_2 : \hat{A}^T y_1 + \hat{B}^T y_2 \leq c, y_1 \text{ free}, y_2 \geq 0\}$. To satisfy complementarity

in the LP, x must solve $c^T x = \hat{a}^T y_1 + \hat{b}^T y_2$. All together, for x to be an optimal solution to the LP, it must satisfy the following necessary and sufficient conditions:

$$
\begin{aligned}
\hat{A}x &= \hat{a} \\
\hat{B}x &\geq \hat{b}, x \geq 0 \\
\hat{A}^T y_1 + \hat{B}^T y_2 &\leq c, y_2 \geq 0 \\
c^T x &= \hat{a}^T y_1 + \hat{b}^T y_2.
\end{aligned}
\tag{4}
$$

We look back to the Robust LP problem from Section 4.3.1: $\min\{c^T x : Ax = a, Bx \geq b \,\forall\, B \in U, x \geq 0\}$ where $U = \{\bar{B} + u^T R : u^T u \leq 1\}$. We use the idea of lower semi-continuity to show that as uncertainty decreases to 0 in the Robust LP and constraints from the Robust problem converge to constraints from the Linear problem, the optimal solution to the Robust problem will converge to an optimal solution to the Linear problem. More specifically, as a sequence $(A^k, a^k, R^k, \bar{B}^k, b^k)$ converges to $(\hat{A}, \hat{a}, 0, \hat{B}, \hat{b})$, uncertainty decreases to 0 and the robust constraints converge back to the linear constraints. If there exists a corresponding sequence $(x^k, \lambda^k, \rho^k, \rho_0^k)$ that converges to (x^*, y_1, ρ^*, y_2), then x^* satisfies the necessary and sufficient conditions for the linear program by the previous theorem. In this case, we have decreased uncertainty and shown that the optimal solution to our robust model converges to an optimal solution to the linear model. Notice that when assuming lower semi-continuity, the system of equations (3) must hold true for the given convergent values. In other words,

$$
\begin{aligned}
\hat{A}x^* &= \hat{a} \\
\|0 \cdot x^*\| &\leq \hat{B}x^* - \hat{b} \\
\hat{A}^T y_1 + \hat{B}^T y_2 - \sum_i (0)(\rho^*)^i &\leq c \\
y_2 &\geq \|\rho^i\| \\
c^T x^* - \hat{a}^T y_1 - \hat{b}^T y_2 &= 0.
\end{aligned}
\tag{5}
$$

must hold true. Ignoring the fourth equation in the system and simplifying, we know the following must be true:

$$
\begin{aligned}
\hat{A}x^* &= \hat{a} \\
\hat{b} &\leq \hat{B}x^* \\
\hat{A}^T y_1 + \hat{B}^T y_2 &\leq c \\
c^T x^* &= \hat{a}^T y_1 + \hat{b}^T y_2.
\end{aligned}
\tag{6}
$$

Looking back to the original posed Linear Program, we see that x^* now satisfies the necessary and sufficient conditions for an optimal solution in the LP (4). Thus, the system of equations for the robust model collapses to that of the linear model, and our optimal solutions to the robust model converge to an optimal solution to the linear model. Alternatively, instead of having R converge to 0, we could impose the more-relaxed conditions that $\|Rx^*\|$ converges to 0 and $R\rho^*$ converges to 0 and get the same result.

We return to the example of a comparative Robust LP and LP used in Section 2.4.1 to illustrate this result. Recall that the boundary of the feasible region for each problem can be represented in the graph:

Robust vs Linear
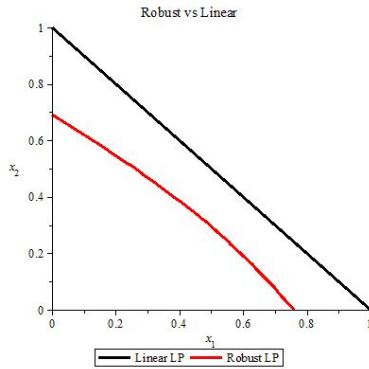
Lower Semi-Continuity guarantees a pair of boundaries for feasible solutions that look like the following:
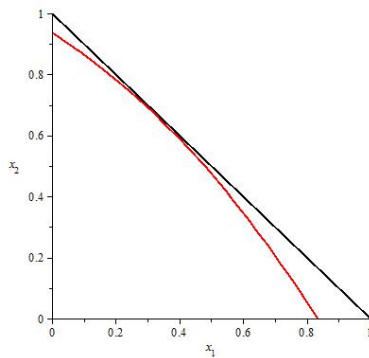
We see here that uncertainty has decreased, so the robust boundary lies close to the linear boundary. Lower Semi-Continuity tells us that the robust LP behaves as we would expect-this visual indeed does describe how robust solutions behave as we decrease uncertainty to 0.

4.3.3 Application to RAMP and FBA

Refer back to Section 2 and recall that the Robust Aanalysis of Metabolic Pathways model is a Robust LP and the FBA model is a related LP. In applying this result to Flux Balance Analysis and RAMP, we validate the legitimacy of our new robust model. The FBA model has been used extensively and it is reassuring that if there is small uncertainty in the robust model, the optimal solution to the robust model is close to that of the linear model. Also, as robust uncertainty decreases, if we can assume that the robust optimal solution converges, we know that the robust optimal solution gets closer to a linear optimal solution. We can thus replicate the linear FBA model if we use specific parameters in the RAMP model.

4.4 Upper Semi-Continuity

4.4.1 Upper Semi-Continuity of Robust LP

We have shown that as we decrease uncertainty to zero, the optimal solution to the Robust LP converges an optimal solution of the original LP if it converges somewhere. It is also of interest to assure that we can converge our Robust LP back to *every* optimal solution to the linear model if we construct our convergent sequence in a specific way. This argument describes our function f as upper semi-continuous. Recall the Robust LP stated in Section 4.3.1: $\min\{c^T x : Ax = a, B_i x \geq b_i \,\forall\, B_i \in \mathcal{U}_i, x \geq 0\}$ where $\mathcal{U}_i = \{\bar{B}_i + u^T R_i : u^T u \leq 1\}$. We first prove a lemma.

Lemma. Given x_B^*, which describes a single optimal LP solution, and y_{1B} and y_{2B}, the corresponding dual LP variables, then $(x_B^*, y_{1B}, 0, y_{2B}) \in f(\hat{A}, \hat{a}, R_B, \hat{B}, \hat{b})$ where R_B is defined as a non-zero matrix such that $||R_B x_B^*|| = 0$ and $||R_B x_B|| \neq 0$, where x_B is any other optimal linear solution.

Proof. The proof follows an extension of ideas from Section 4.3.2. Since x_B^* is an optimal LP solution, it solves the equations (4) for $y_{1B}, y_{2B}, \hat{A}, \hat{a}, \hat{B}, \hat{b}$. To assure $(x_B^*, y_{1B}, 0, y_{2B}) \in f(\hat{A}, \hat{a}, R_B, \hat{B}, \hat{b})$, the inputs must satisfy (3). So we must have:

$$
\begin{aligned}
\hat{A} x_B^* &= \hat{a} \\
||R_B x_B^*|| &\leq \hat{B} x_B^* - \hat{b} \\
\hat{A}^T y_{1B} + \hat{B}^T y_{2B} - \textstyle\sum_i (R_B)^T \cdot 0 &\leq c \\
y_{2B} &\geq ||0|| \\
c^T x_B^* - \hat{a}^T y_{1B} - \hat{b}^T y_{2B} &= 0.
\end{aligned}
\tag{7}
$$

The fourth equation follows from constraints on the dual variable in the original LP. All of the other equations are equivalent to those in (4), which are given as true.

Now we prove the theorem.

Theorem. Upper Semi-Continuity: Given $(x_B^*, y_{1B}, 0, y_{2B}) \in f(\hat{A}, \hat{a}, R_B, \hat{B}, \hat{b}) \,\exists$ a sequence $(A^k, a^k, R^k, \bar{B}^k, b^k) \to (\hat{A}, \hat{a}, R_B, \hat{B}, \hat{b})$ and $(x^k, \lambda^k, \rho^k, \rho_0^k) \in f(A^k, a^k, R^k, \bar{B}^k, b^k)$ such that $(x^k, \lambda^k, \rho^k, \rho_0^k) \to (x_B^*, y_{1B}, 0, y_{2B})$.

Proof. Consider the sequence $(\hat{A}, \hat{a}, tR_B, \bar{B}, \bar{b})$ where we let $t \to 0$. We know that $R_B x_B^* = 0$, or that x_B^* is in the null space of R_B. Then x_B^* must also be in the null space of αR_B, where α is a scalar. Thus, for the sequence given above, USC Lemma verifies that $(x_B^*, y_{1B}, 0, y_{2B})$ is in the mapping of every member of the sequence. Therefore, the constant sequence $(x_B^*, y_{1B}, 0, y_{2B})$ fits the conditions of the theorem and trivially converges to $(x_B^*, y_{1B}, 0, y_{2B})$.

We look back again to our example of an LP and a Robust LP. Upper Semi-Continuity guarantees that by skewing R in a specific way, we can converge the robust solutions back to *any* solution to the linear model. So, we don't only have a convergence like this:
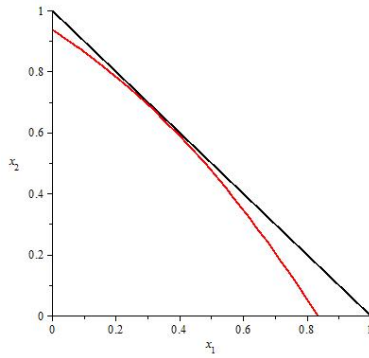
but also one like this:
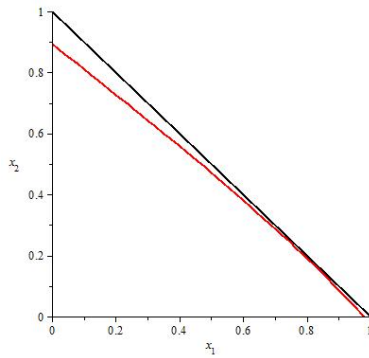
and in fact, a convergence of the red curve to every point on the black curve. Our specific method for obtaining upper semi-continuity relied on a scaling of R, which would lead to a converge that looks something like the following in our example:
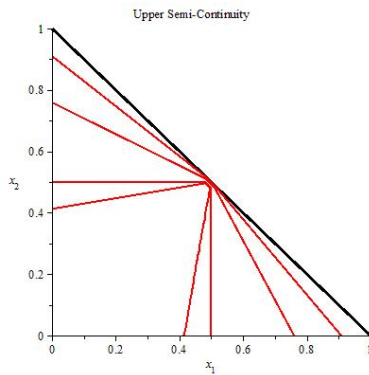
The graphs here apply to the particular toy example used above, note that usually we are dealing with high-dimensionality, which does not lend itself to this geometry.

40

4.4.2   Application to RAMP and FBA

In applying the result to FBA and RAMP, we have that every optimal solution to a posed linear FBA problem can be described as the limit of a specific set of solutions to RAMP. The parameters that define the uncertainty set used in RAMP, mainly R_B here, must satisfy very specific rules. In FBA, we have the constraint $S_i v = 0$, while in RAMP, we have $-M_i \leq S_i v \leq M_i \, \forall \, S_i \in \mathcal{U}_i$. In order to converge back to the linear model, we have to relax the inequality in the robust model, so we need $S_i v = 0 \, \forall \, S_i \in \mathcal{U}_i$ where $\mathcal{U} = \{\bar{B} + u^T R : u^T u \leq 1\}$. To get Upper Semi-Continuity, we also need $R_B x_B^* = 0$. To satisfy the equality under an uncertainty set condition, our robust LP reduces to a linear problem with the added linear constraint $R_B x_B^* = 0$. R_B must be one-dimensional for this to be true. To apply Upper Semi-Continuity to FBA and RAMP, we must specify R_B as a particular matrix, but one which is not wholly unlikely for our problem.

5   Conclusion

Inspired by the systems biology problem Flux Balance Analysis (FBA), we have introduced Robust Analysis of Metabolic Pathways (RAMP) to provide a more theoretically sound and computationally accurate model of cellular metabolism. RAMP recognizes the faulty assumptions of traditional FBA and corrects them by allowing deviation from steady-state and accounting for variability across a cellular culture. RAMP is computationally more successful in predicting the essentiality of genes and reducing degeneracy in optimal flux values. Theoretical results describe the stability of RAMP under perturbations in modeling parameters and the convergence of FBA to RAMP when uncertainty decreases. The inclusion of new modeling parameters in RAMP opens the possibility of modeling a wider range of cellular conditions, including non-optimized cultures. We conclude that RMAP is an improvement over traditional FBA and deserves further study.

References

[1] UCSD Bioengineering, Sigillum Universitatis Islandiae, and Systems biology research group. The opencobra project.

[2] S. Y. Gerdes, M. D. Scholle, J. W. Campbell, G. Balzsi, E. Ravasz, M. D. Daugherty, A. L. Somera, N. C. Kyrpides, I. Anderson, M. S. Gelfand, A. Bhattacharya, V. Kapatral, M. D'Souza, M. V. Baev, Y. Grechkin, F. Mseeh, M. Y. Fonstein, R. Overbeek, A.-L. Barabsi, Z. N. Oltvai, , and A. L. Osterman. Genomics and proteomics: Experimental determination and system level analysis of essential genes in escherichia coli mg1655. *J. Bacteriol*, 185:5673–5684.

[3] Orth JD, Conrad TM, Na J, Lerman JA, Nam H, Feist AM, and Palsson BO. A comprehensive genome-scale reconstruction of escherichia coli metabolism–2011. *Mol Syst Biol*.

[4] Andrew R. Joyce, Jennifer L. Reed, Aprilfawn White, Robert Edwards, Tomoya Baba, Hirotada Mori, Scott A. Lesely, Bernhard Palsson, and Sanjay Agarwalla. Experimental

and computational assessment of conditionally essential genes in escherichia coli. *Journal of Bacteriology*, 188(23):8259–8271.

[5] Robert Schuetz, Nicola Zamboni, Mattia Zampieri, Matthias Heinemann, and Uwe Sauer. Supplementary materials for multidemsional optimality of microbial metabolism. *Science*, 336(6081):601–604.

[6] Daniel Segrè, Dennis Vitkup, and George M. Church. Analysis of optimality in natural and perturbed metabolic networks. *PNAS*, 99(23).

[7] San Diego Systems Biology Research Group at the University of California. Bigg database.

5.1 Acknowledgments