Building a cell-free metabolic model using variational autoencoders

Nicholas L. Larus-Stone
Queens' College

UNIVERSITY OF CAMBRIDGE

A dissertation submitted to the University of Cambridge in partial fulfilment of the requirements for the degree of Master of Philosophy in Advanced Computer Science

University of Cambridge
Department of Computer Science and Technology
William Gates Building
15 JJ Thomson Avenue
Cambridge CB3 0FD
United Kingdom

Email: nl363@cl.cam.ac.uk

June 8, 2018

Acknowledgements

I could not have done this dissertation without the support of my two wonderful advisors Pietro Lio and Jim Haseloff. Pietro has been consistent encouragement throughout this entire process and he never ceases to amaze me with the depth of his knowledge and perpetual enthusiam. Jim was extraordinarily kind to let a computer scientist run around his lab for a few months and I appreciate that he let me get my hands dirty. Both of their groups have also been extremely welcoming despite me not being in either lab full time.

Emma Talbot was my mentor in the Haseloff group and taught me everything I needed to know about the wet lab in order to get this dissertation done on time. In Pietro's lab, Marco Barsacchi provided extremely helpful advice on how to combine VAEs and FBA and was always willing to chat. I really appreciate both of their help.

To my Cambridge friends Tom, Harriet, Emily, and Ian: thank you for putting up with my craziness and joining me for many meals during this entire process. Seren, thank you for the copious amounts of pesto you made me as well as your emotional support throughout this process.

Thank you also to Juanky Perdomo and my dad, James Larus, for reading drafts of this dissertation and giving me feedback.

Finally, I would like to thank all of my friends and family who encouraged me on this journey and helped me get to where I am today. In particular, I need to thank my parents, James Larus and Diana Stone, who have nurtured my love of learning and research and put me on the track that led me here.

Declaration

I Nicholas L. Larus-Stone of Queens' College, being a candidate for the M.Phil in Advanced Computer Science, hereby declare that this report and the work described in it are my own work, unaided except as may be specified below, and that the report does not contain material that has already been used to any substantial extent for a comparable purpose.

Total word count: 10698

Signed:

Date:

Building a cell-free metabolic model using variational autoencoders

Abstract

Cell-free protein synthesis (CFPS) systems are a promising platform for a variety of synthetic biology applications. These systems are simpler, faster to experiment with, and cheaper than comparable *in vivo* approaches. Despite their relative simplicity, CFPS systems lack the computational metabolic models of cell-based systems. This dissertation provides the first end-to-end system, called Cell-Free Flux Balance Analysis (CF-FBA), that reduces a cell-based metabolic model into a CFPS model based on experimental data. This type of model is useful for biologists who want to understand what reactions are important in their CFPS systems.

Through the creation of a system to generate these reduced models, I contribute a number of advances to the field of computational metabolic modeling and deep learning. As one of the first applications of deep learning to metabolic modeling, CF-FBA provides a groundwork for further exploration at the intersection of these fields. In particular, I show that although Principal Component Analysis (PCA) is a popular dimensionality reduction technique in computational biology, Variational Autoencoders (VAEs) can find non-linear relationships that PCA cannot. I also describe a VAE called Corr-VAE that uses a custom loss function to discover better low dimensional representations than a naive VAE. This loss function enables CF-FBA to generate metabolic models that are sensitive to different experimental conditions.

This dissertation explores the use of CF-FBA to create *Escherichia Coli* (*E. coli*) CFPS models, but has broader implications in the field of biology. While Corr-VAE was designed to solve a problem relating to CFPS systems, it can be used in any biological application that generates experimental data. Additionally, though CF-FBA was implemented using data from *E. coli* CFPS systems, it can be applied to other CFPS systems derived from any organism that has a Genome scale Model (GEM). This work will lead to new ways to configure and monitor cell-free systems and will be useful to wet lab biologists working in the space of CFPS systems.

Contents

List of Figures

List of Tables

Acronyms

*μ***L** microliter. 4, 26, 27, 55

E. coli *Escherichia Coli.* 6, 7, 17–20, 24, 30, 47, 51

AA Amino acid. 25, 53, 56

AE Autoencoder. 14, 20, 21, 51

cAMP cyclic adenosine monophosphate. 25, 56

CF-FBA Cell-Free Flux Balance Analysis. 1, 2, 4–7, 23, 28–33, 35, 37, 42, 45, 48–51

CFPS Cell-free protein synthesis. 1–7, 9–12, 19, 20, 24–31, 37–39, 42, 48, 49, 51, 56

CoA Coenzyme-A. 56

COBRA Constraint-Based Reconstruction and Analysis. 3, 30

CSV Comma Separated Value file. 28, 29, 31

DNA deoxyribonucleic acid. 4, 24–27, 56

FBA Flux Balance Analysis. 2–4, 6, 7, 10–12, 17, 19, 20, 30–35, 48, 49

GEM Genome scale Model. 3, 5–7, 18–20, 35, 37, 42, 43, 45, 48

HMP Hexose monophosphate. 26, 56

K Potassium. 26, 56

KL Kullback-Leiber. 14

LB Luria Broth. 24

LP Linear Programming. 11, 17

ME-Model Metabolic and gene Expression Model. 18

Mg Magnesium. 26, 56

MILP Mixed Integer Linear Programming. 18

mL millileter. 24, 25

mM Micromolar. 56

NAD nicotinamide adenine dinucleotide. 25, 56

nL nanoliter. 27, 55

NTP nucleoside triphosphate. 25, 26, 53

OD optical density. 24

PCA Principal Component Analysis. 5, 7, 12, 19, 20, 37–39, 51

PEG Polyethylene glycol. 26

PEMA Principal Element Mode Analysis. 19

ReLU Rectified Linear Unit. 34

RFP red fluorescent protein. 27

RNA ribonucleic acid. 4

TL translation. 10, 26

tRNA transfer RNA. 25, 56

tSNE t-Distributed Stochastic Neighbor Embedding. 39, 41, 42

TX transcription. 10, 26

VAE Variational Autoencoder. 2, 4–7, 14, 15, 17, 20, 21, 32–34, 37–39, 44, 48, 49, 51

Chapter 1

Introduction

Despite more than a century of active biology research, many biological systems lack good computational models. Computational models are useful if they accurately capture how the underlying biological system behaves. When these models are faithful to the underlying biology, experiments can be run *in silico* instead of *in vivo*, speeding up the pace of research. Moreover, the best computational models are useful not only as biology simulators, but also as facilitators of new hypotheses. Computational biology models should both help researchers improve their understanding of the modeled system and expose novel insights into the underlying biology.

Biological systems are complex and can be examined at various levels of abstraction. For example, one model could use a physical description of a cell to predict its shape as it grows. A different model could use -omic data—genomic, transcriptomic, and metabolomic measurements—to model the production of compounds that allow for cellular growth. My efforts focus on modeling the metabolism of a biological system to help biologists better understand what reactions are occurring in that system.

The goal of this work is to provide a metabolic model of a Cell-Free Protein Synthesis (CFPS) system that is useful for biologists working with CFPS systems. To that end, I introduce CF-FBA, an end-to-end system that produces

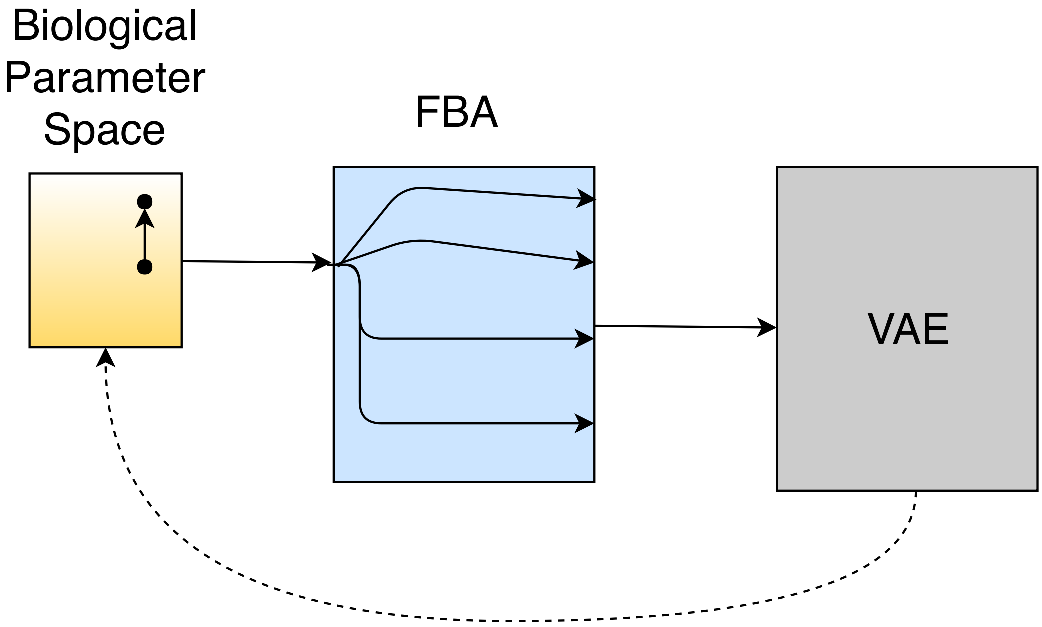


Figure 1.1: High-level idea behind CF-FBA. Small changes in the parameter space are amplified across many different fluxes in a FBA model. A VAE can help elucidate how the changes in the fluxes in the FBA model reflect the changes in the parameter space.

metabolic models for CFPS systems from experimental data. CF-FBA learns which reactions are most important for a given cell-free system and produces a reduced metabolic model that fits the experimental data. These reduced models are produced through the use of a Variational Autoencoder (VAE) on top of a common metabolic modeling technique called Flux Balance Analysis (FBA). In order to incorporate experimental data into the model, I also introduce a new VAE loss function based on correlation that I call a Corr-VAE. The Corr-VAE perturbs the latent space of a VAE to better fit the experimental data. CF-FBA can be used by biologists to navigate the parameter space of different starting conditions, cope with batch variation, and gain a deeper understand of the composition of cell-free systems. The code for CF-FBA can be found at `https://github.com/nlarusstone/cf\_fba`.

Metabolic models

Metabolism is defined as all of the chemical reactions that occur within an entity—usually an organism or a cell—in order to sustain its life and growth. Biologists are interested in understanding how different parts of metabolism function, as metabolic reactions are the end result of the other biological process of interest (e.g. transcription and translation). While ongoing work continues to develop a more robust understanding of metabolism, recent advances have begun to explore applications that involve manipulating the metabolism. Metabolic engineering has the potential to create novel advances in medicine and energy production such as increased bioproduction of the antimalarial drug artemisin [1]. Metabolic models are an important tool for metabolic engineers because they can be used to predict how the phenotype of an organism will respond to manipulation.

Many metabolic models represent the chemical reactions in a system with a mathematical representation of the reaction coefficients. Individual reactions are denoted as equations in these models. Mathematical constraints are placed on these reactions based on an understanding of the biological pathways they represent. Models of this type are collected under the heading of Constraint-Based Reconstruction and Analysis (COBRA) models [2] and are very common in the metabolic engineering community [3]. One of the most popular constraint-based metabolic models is FBA. It has been used in hundreds of different works to model metabolism [4]. These applications range from optimizing metabolic pathways [5] to investigating gene networks [6]. FBA is based on a genome-scale metabolic models (GEMs) reconstructed from the genome of an organism of interest. However, these GEMs describe living organisms, not cell-free systems, so typical FBA models cannot be applied out of the box to a cell-free system.

In addition, since metabolism is a heterogenous mixture of reactions, small changes in the parameter space can lead to large changes in the metabolic output of a system. The parameter space for CFPS systems primarily consists of final concentrations of energy substrates. Current models are unable to explain how changes in the concentration of a certain reactant will affect the output of a CFPS system. Even using a modeling technique such as FBA will not solve this issue because FBA acts as a non-linear amplifier on small changes in the parameter space. The use of a dimensionality reduction technique such as a VAE can help explain how the changes in the original parameter space are reflected in the output of a FBA model. Figure 1.1 demonstrates this idea, a key motivation behind the creation of CF-FBA.

Building this type of learning system requires datasets that explore across a biologically relevant parameter space. These datasets are not common, so this dissertation required extensive experimental work in order to generate data that could be used to learn a model. I will be also be releasing the CFPS datasets I generated to aid with further modeling work of CFPS systems. Additionally, performing both the experimental work and the modeling allowed me to acquire a deep understanding of the data I was generating. Without this understanding, building biologically relevant models would be difficult.

Cell-free systems

A CFPS system is a reduced version of a cell that produces proteins without being alive. The Central Dogma of biology, as enunciated by Francis Crick, is that deoxyribonucleic acid (DNA) makes ribonucleic acid (RNA) makes protein [7]. While this may be a slight oversimplification, the key idea is that the production of proteins is the end goal for most biological systems. CFPS systems take that to an extreme by removing all endogenous DNA, RNA, and membranes while retaining the machinery for transcription and translation.

A CFPS system is therefore a type of programmable matter, a veritable biological computer. The CFPS system acts like a computer because it can "execute" any DNA "program" that is added to the mixture. The output of a CFPS system is determined by which DNA program is loaded into a cell-free system. This simplicity has made cell-free systems popular for a wide array of applications [8, 9, 10]. Additionally, cell-free systems have been shown to be extremely cheap, costing less than \$0.01 per μL of reaction [11, 12].

CFPS systems have two key aspects that make them an ideal platform to model for this dissertation. First of all, running experiments in a CFPS system is much faster than typical *in vivo* methods because performing an experiment does not require growing cells. This meant that I was able to start from scratch and generate relevant data within several weeks instead of the many months or years for a cell-based system. CFPS systems are also simpler than a full cellular system and should therefore be easier to model accurately. By definition, CFPS systems are composed of blended cells and contain a strict subset of the reactions that occur in a cell-based system. The primary problem is that which reactions are included in that subset is currently unknown.

The unknown composition of cell-free systems means that despite their reduced complexity, CFPS systems lack good models. However, prior attempts at modeling CFPS systems have used metabolic models to examine CFPS systems. Metabolic models are a natural fit for modeling CFPS systems. Since CFPS are not living organisms, their effectiveness is entirely based on reaction constraints and energy regeneration. Thus, understanding the metabolism of these systems is extremely important. The few existing metabolic models for CFPS systems have been constructed by hand, limiting their general applicability [13, 14]. CF-FBA is an automated system to reduce standard GEMs to metabolic models of CFPS systems.

Contributions

This dissertation makes a number of contributions in the field of metabolic modeling and dimensionality reduction for CFPS systems.

- CF-FBA is one of the first applications of deep learning to metabolic models. I show that using a VAE is more effective than PCA when building a reduced metabolic model. Since PCA is a common dimensionality reduction technique in biology, this motivates other work to consider using VAEs instead of PCA.

- Corr-VAE is a new loss function for VAEs that incorporates a correlation loss term. Although Corr-VAE was developed specifically for use with experimental data generated from CFPS systems, the structure of the network does not require it to be used only with cell-free systems. Corr-VAE can be used in other biological application that generate

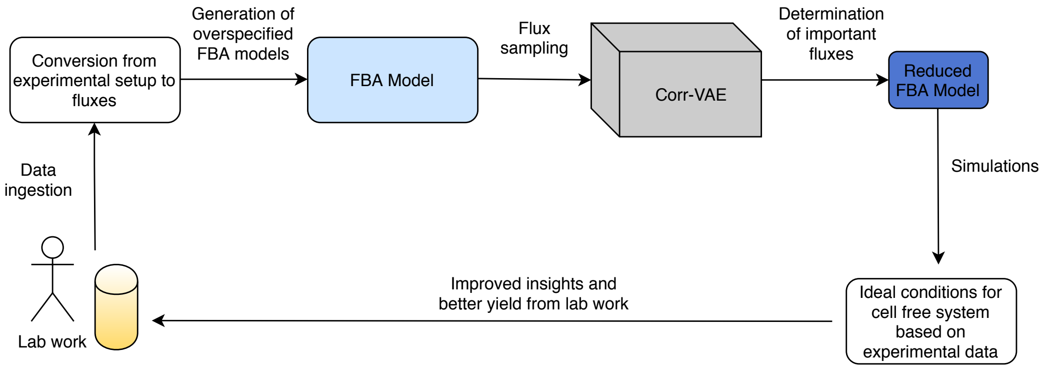

Conversion from experimental setup to fluxes
Generation of overspecified FBA models
FBA Model
Flux sampling
Corr-VAE
Determination of important fluxes
Reduced FBA Model
Simulations
Ideal conditions for cell free system based on experimental data
Improved insights and better yield from lab work
Lab work
Data ingestion

Figure 1.2: The overall pipeline for CF-FBA. Given an experimental setup and experimental data, CF-FBA generates a reduced cell-free metabolic model from the corresponding GEM.

experimental data and have a need for dimensionality reduction.

- Finally, I produce a number of new, reduced models for CFPS systems. These models more accurately describe CFPS metabolism than full-scale GEMs. The reduced models can be used by biologists and metabolic engineers working with these CFPS systems who want a FBA model of their system.

My approach was intentionally designed to be as general as possible, which has two auxiliary benefits. CF-FBA can be applied out-of-the-box to other types of cell-free systems; i.e. systems that use cells from organisms other than *E. coli*. Additionally, the techniques developed here for modeling cell-free systems can be applied to modeling other biological systems that uses experimental data and metabolic models.

Outline

The structure of this dissertation is as follows. Chapter 2 provides background information on cell-free systems, FBA, and VAEs. Chapter 3 introduces related work in the fields of metabolic models, reduction of metabolic models, modeling cell-free systems, and VAEs. Chapter 4 describes CF-FBA, the system I have built to generate CFPS models (shown in Figure 1.2). Chapter 5 examines the results of using a VAE to reduce FBA models.

Notably, I show that the reduced models CF-FBA produces are better at describing CFPS systems than both GEMs and already published reduced models. Chapter 6 discusses future work and provides suggestions of how this work may be utilized. The final chapter concludes with an overview of the project and its key advances.

Chapter 2

Background

2.1 Cell-free systems

CFPS systems have existed since the early 1960's as a way to express proteins that are otherwise difficult to express in a living cell [15]. These early CFPS systems were quite crude and had low protein yields, limiting their utility. More recently, work involving the removal of certain genes [16] and the improved stability of cofactor regeneration pathways [17] has improved the yield of CFPS systems.

When discussing CFPS systems, it is important to distinguish between the multiple types of cell-free systems. One type of cell-free system is created by combining individual purified proteins involved in transcription and translation. The most widely utilized of these reconstituted cell-free systems is the PURE system [18]. The benefit of these systems is that the system is completely known and controlled. Since the only things in the system are the proteins that have been deliberately added, users of the systems can be assured that there are no undesirable elements such as nucleases or proteases present. However, these systems have relatively high costs due to the time and effort required to isolate and purify each of the requisite proteins.

The type of CFPS system used in this dissertation is an extract-based system created using a crude cell extract. Creating an extract-based CFPS system involves growing cells and then lysing them to create a cell extract. This extract is then used as the basis of the CFPS system with the assumption that all the important transcription (TX)/translation (TL) machinery remains intact in the cell extract. These systems are also known as TXTL systems for this reason. Growing large quantities of cells is relatively cheap, so these types of systems have the potential to scale more effectively than reconstituted cell-free systems. The downside to this method of creating CFPS systems is that the exact composition of these cell extracts is unknown and can potentially vary between batches of CFPS systems. This motivates the need to provide an accurate model of these crude CFPS systems.

2.2 Flux Balance Analysis

Flux Balance Analysis (FBA) is one of the most common metabolic modeling tools. FBA relies on a mathematical representation of all of the metabolic reactions occurring in an organism. These reactions are represented using a matrix S, a $M \times N$ matrix, where each row represents a separate reaction and each column is a different metabolite. The entries in this matrix are the stoichiometric coefficient for each metabolite in a given reaction. Metabolites that are consumed in a reaction are represented using negative numbers. Flux through each of these reactions can be represented by a vector v, which is a 1-dimension vector with a length equal to the number of the reactions. A crucial assumption in FBA is that the system is at steady state, so the overall flux through all of the reactions is not changing. This is expressed mathematically as:

$$S \times v = \vec{0} \tag{2.1}$$

where S is the known stoichiometric matrix and v is the unknown flux vector. The goal of FBA is to solve for v.

This system is underspecified—there are typically more reactions than metabolites ($N >> M$)—so there is no unique solution to this equation. FBA solves this issue by specifying an objective function and turning this into an optimization problem. This objective function can be represented as a vector c, whose coefficients select which reactions are to be optimized for. In the case where the goal of a system is the production of a specific product, c is extremely sparse, having only a single entry. A common choice of objective function is the Biomass Objective Function, which is a pseudo-reaction that incorporates many essential metabolites necessary for cell growth [19].

FBA also uses a constraint matrix to incorporate biological information into the model. The constraint matrix specifies constraints on different entries of v in order to limit the amount of flux proceeding through a given reaction. This allows the model to add in biological information such as reaction irreversibility or metabolite uptake. For instance, an irreversible reaction is represented by creating a constraint that the lower bound on the flux for that reaction is 0.

Now the FBA model can be reformulated as a Linear Programming (LP) problem of the form:

$$\begin{aligned} \max\ Z = c^T v & \\ \text{s.t. } & S \times v = \vec{0} \\ & \text{and } -\inf < v_0 < \inf \\ & \text{and } 0 < v_1 < \inf \\ & \text{and } 0 < v_2 < 10 \\ \ldots & \end{aligned} \tag{2.2}$$

where v_i is the flux through reaction i and Z is the flux through the objective function. v_0 is an example of a reversible reaction because flux can go either way. v_1 and v_2 are both irreversible reactions, but v_2 is constrained due to some external biological knowledge. FBA models then use a LP solver to find the optimal distribution of fluxes.

FBA makes a key assumption that the system is at steady state—i.e. the system is stable. However, CFPS systems are dynamic systems and do not have a stable period where intake and output are equal until the system is no longer producing protein. To use FBA with CFPS, I consider the period of maximal production of a CFPS system to be its steady state. This is the most important interval for a CFPS system, so FBA can be used to analyze this pseudo-steady state.

2.3 Autoencoders

Autoencoders are a dimensionality reduction technique with a simple idea: given a dataset, try to reconstruct that dataset by passing it through a lower dimensional subspace. Deep autoencoders, which I will just refer to as autoencoders, use two neural networks to accomplish this. One network, called the encoder, performs the mapping from the original dataset to the lower dimensional subspace. A second network, the decoder, attempts to reconstruct the original dataset by mapping back from the lower dimension back to the dimension of the original dataset. The original idea behind autoencoders was that this lower dimensional representation of the data served as a compressed version of the data. Instead of hand designing compression/decompression algorithms, these functions can be learned by a neural network and can therefore be application specific. In practice, these autoencoders often perform worse than hand-designed compression algorithms. For instance, autoencoders have difficulty competing with classic algorithms on well-studied problems such as image compression [20]. However, autoencoders can also be used as a dimensionality reduction technique.

The lower dimensional subspace that the autoencoders map has been shown to be an effective form of unsupervised dimensionality reduction [21]. The transformations that autoencoders learn can sometimes uncover hidden relationships in the data that may be non-obvious in the original dataset. Biologists often use PCA to achieve similar goals. PCA also projects the data into a subspace, but it does so in a linear manner. Autoencoders have the advantage that they are able to learn non-linear transformations.

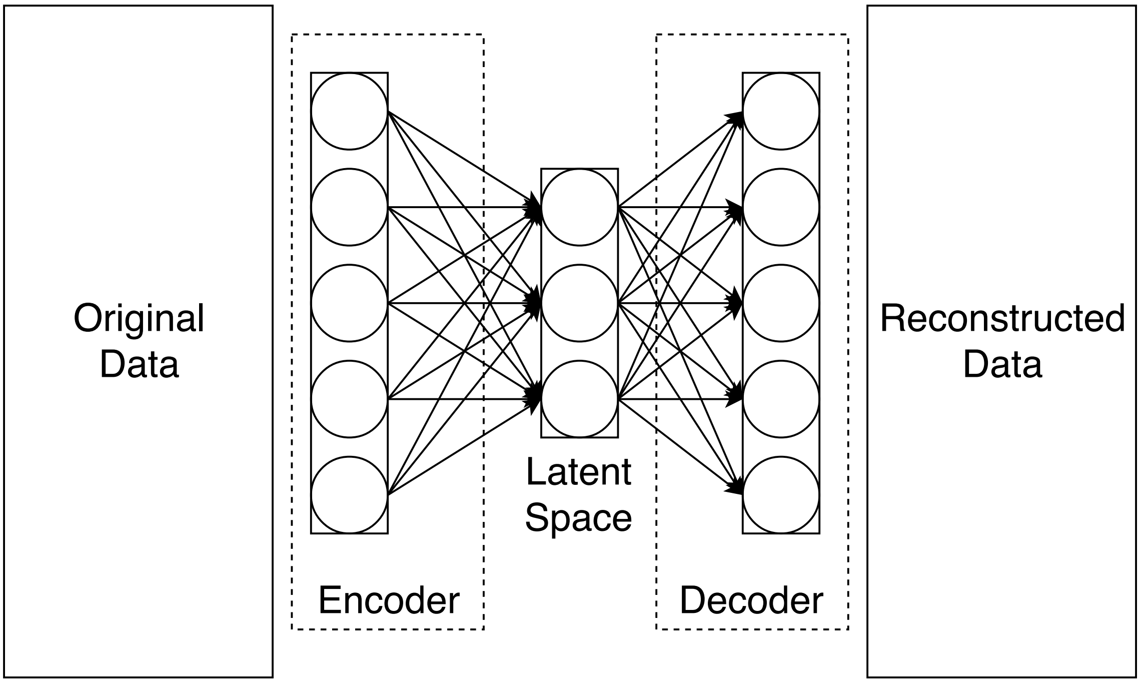


Figure 2.1: Example of a single layer autoencoder. In this case, both the encoder and the decoder are a single layer neural network. These layers can be stacked to create even more complex dimensionality reductions.

Figure 2.3 demonstrates the structure of a very simple autoencoder. While the main idea of an autoencoder is unchanged, there are a number of different versions of autoencoders that are extensions of this basic idea. One example is a sparse autoencoder. In order to force the autoencoder to learn a more general dimensionality reduction, the loss function can include a sparsity constraint to limit the number of activations among the network layers. This forces the autoencoder to learn a representation with fewer network weights, preventing overfitting. Another type of autoencoder is a denoising autoencoder. This type of autoencoder involves taking a noisy representation and mapping it to a clean representation of the data. These autoencoders add noise to the original data and then try to learn the original representation from the corrupted data. Despite their differences, almost all of these autoencoder types are trained using reconstruction loss—a measure of how far the autoencoded data is from the original data.

Sometimes it is important to force Autoencoders (AEs) to do more than just learn arbitrary encoding and decoding functions. For instance, VAEs are a class of autoencoder that imposes constraints on the encoded representation. Specifically, VAEs constrain the encoded representation to be a probability distribution. So, instead of learning deterministic functions to encode/decode the data, a VAE learns a mapping to parameters of an encoded probability distribution. This work uses a normal distribution as the encoded distribution, though other distributions can also be used with VAEs. In this case, the encoder learns to map samples to the mean and standard deviation of the normal distribution. Borrowing terminology from the field of probabilistic models, the encoded dimension is also called the latent space of the VAE. In order to decode a data point, VAEs then samples from the encoded probability distribution to generate a reconstructed data point in the original dimension.

VAEs measure loss using a combination of two loss terms. The first part of the error term measures how well the VAE can reconstruct the original data. This term is called the reconstruction loss and it measures the distance between the reconstructed data and the original data. The second loss term acts as a regularizer on the latent space and constrains it to be well formed. This term is calculated using Kullback-Leiber (KL) divergence, a common way of measuring distance between probability distributions. For a VAE, the loss is based on the KL divergence between the latent space and a prior distribution, usually the normal distribution. With the combination of those two loss terms, the VAE is then trained the same as any other neural network—by using gradient descent to minimize the loss function.

Define $x_i \in X$ to be the original dataset and z the latent representation learned by the VAE. The encoder network can be represented by a function p which has parameters θ. Similarly, the decoder can be represented as a posterior distribution q that is parameterized by ϕ. Then the loss function of a VAE can therefore be written out as follows:

$$\mathcal{L}(\theta, \phi) = -E[\log p_\theta(x_i|z)] + KL(q_\phi(z|x_i)||p(z)) \tag{2.3}$$

Minimizing this first part of the loss involves reducing the reconstruction error by maximizing the probability of reconstructing the original data given the latent representation. Minimizing the second part of the loss requires the latent representation to be similar to the normal distribution. The VAE learns the θ and ϕ terms that minimize this function.

Chapter 3

Related Work

This chapter introduces related work in the relevant fields of computer science and biological modeling. I begin by examining work in the field of building metabolic models using flux balance analysis. Next, I present a collection of automatic reduction systems for metabolic models, none of which have been applied to modeling cell-free systems. I then examine what types of models have been used to model cell-free systems. Finally, I investigate recent work on using variational autoencoders in biology and applying correlation loss terms to VAEs.

3.1 Flux Balance Analysis

Early metabolic models in the 1980s pioneered the use of stoichiometric equations to describe a biological system and predict the yield of a specific product [22]. These models soon adopted the use of Linear Programming (LP) to solve for the optimal result [23]. After the transition to LP solvers, this field was referred to as constraint-based analysis, while the primary technique used to solve these equations was called FBA. These techniques were soon applied to describe the main metabolic systems in *E. coli* [24]. Importantly, FBA has repeatedly been shown to predict phenotypes that corresponded to real-world data [25, 26, 27, 28].

Improvements on these early models have primarily focused on the addition of new genes, metabolites, and reactions [29]. After the first *E. coli* genome was sequenced and annotated, the size of these models grew rapidly. The earliest genome scale model (GEM) for *E. coli* was iJE660a, a model that described 627 unique reactions in a typical *E. coli* cell [30]. Over the years, more and more reactions have been progressively added to the model to better describe the biology. This work uses the most recent *E. coli* GEM, iJO1366 from 2011, which contains 2583 reactions [31]. Since 2011, work in this field has primarily focused on extending metabolic models to incorporate other types of biological information. For instance, recent work has involved the addition of gene expression data to create Metabolic and gene Expression Models (ME-Models) [32]. This has drastically increased the dimensionality and complexity of the model: the most recent ME-Model, iJL1678-ME has 79871 reactions.

3.2 Reduction of metabolic models

As these GEMs begin to incorporate even more information, the models begin to suffer from the curse of dimensionality [33]. To deal with this issue, a number of papers have tried to reduce these genome scale models to their most essential reactions. The Hatzimanikatis lab has developed multiple tools to reduce the dimensionality of these models. redGEM finds core metabolic models by performing a graph search where the objective is to minimize information loss [34]. lumpGEM also uses graph search techniques to identify and collapses entire reaction networks into a single balanced reaction [35]. Other important tools in this area are NetworkReducer and minNW. NetworkReducer iteratively prunes and compresses reaction networks while maintaining a set of protected reactions until a core model is found [36]. minNW uses Mixed Integer Linear Programming (MILP) techniques to compute minimal subnetworks with certain constraints [37]. All of these techniques reduce the dimensionality of a full GEM to a simpler representation of the system.

The methods above use some sort of search technique to find minimal core models from a stoichiometric description of the entire system. However, there has also been work that uses general dimensionality reduction techniques such as PCA. This idea has been incorporated into a reduction technique called Principal Element Mode Analysis (PEMA) [38]. PEMA identifies subnetworks that maximize the observed variance similar to how the principal components of PCA work. A more recent method called "Principal metabolic flux mode analysis" explicitly incorporates PCA and FBA [39]. It reduces the dimensionality of the system by running a form of PCA on the stoichiometric matrix that is regularized by the steady state assumption from FBA.

This work differs from these earlier reduction techniques in a few ways. Prior work so far has been constrained to looking at linear techniques of dimensionality reduction. Many biological relationships are non-linear, so there are limits to how well linear models can perform. I approach the problem of reducing metabolic models differently by using nonlinear techniques and deep learning. In addition, the techniques above all deal with the stoichiometric matrix of the system without using real experimental data. Finally, none of the techniques above have specifically targeted building models for CFPS systems.

3.3 Modeling cell-free systems

Models of cell-free systems have typically been based on kinetic models of transcription and translation. These models are often quite good at predicting the time-course of protein production, but they have a very narrow focus. Transcription and translation reactions can be described using differential equations when the rate constant of the reaction is known. This type of kinetic model has been applied to the commercial *E. coli* cell-free system TXTL, and was able to accurately describe the dynamics of their gene circuit of interest [40]. Other work has proposed a more general framework for modeling metabolic networks in cell-free systems using these types of kinetic models [41]. One issue with these types of kinetic models is that only reactions with known rate constants can be modeled accurately. Recent work attempts to solve that problem using Bayesian parameter inference to infer the kinetic parameters for these reactions [42].

There have only been a few attempts to use constraint-based metabolic models and FBA to model cell-free systems. One model adapted a full-scale *E. coli* FBA model to a cell-free system by hand [13]. The authors decided which parts of the full model were relevant for cell-free systems and then removed any irrelevant reactions from the model. A more recent attempt took a bottom-up approach to building a FBA model for cell-free systems [14]. Instead of removing reactions from the full *E. coli* GEM until they had a cell-free model, the authors instead selected the reactions they believed to be most important for protein synthesis. Since protein synthesis is the main goal of CFPS systems, they were able to use these reactions to build an accurate cell-free FBA model. These models show that it is possible to use FBA to describe cell-free systems. However, neither approach is able to construct cell-free systems in a systematic way that could scale to other labs or other organisms.

3.4 Variational autoencoders

Autoencoders have been around for many years, but VAEs were first introduced in 2013 [43, 44]. Since their introduction, VAEs have been used for everything from transferring image features [45] to molecule generation [46]. AEs, and VAEs in particular, have only just begun making their way into analyzing biological data. The typical dimensionality reduction technique in biology is PCA [47]. However, recent work has begun to explore the uses of VAEs on biological data. One study examined cancer transcriptomics data using a VAE to extract meaningful features of cancer gene expression [48]. Another work showed that a VAE was able to separate tissue types based on mass spectrometry data [49]. These papers have used the original structure and loss function for their VAEs, but there have also been development of new types of VAEs.

Specifically, work has also been done to incorporate correlation loss terms with AEs. Correlation Neural Networks were first introduced as a way to make AEs more effective on multi-modal data such as labeled images [50]. By maximizing the correlation between the latent representations for each view of the data, the authors were able to improve performance. This idea has been expanded in Relational Neural Networks, which also looks at correlation within the data to affect the AE loss term [51]. All of these papers use correlations within the dataset or latent representation of the AE. I introduce a new type of VAE, Corr-VAE, which uses a correlation loss term with respect to external experimental data.

Chapter 4

Design and Implementation

¡¡¡¡¡¡¡ HEAD =======

¿¿¿¿¿¿¿ d36f4c58ce53756c4208d542755ca45e64177c23 The primary deliverable of this dissertation is a system, CF-FBA (shown in Figure 1.2) that is able to generate metabolic models of cell-free systems. This chapter decomposes CF-FBA into its four main parts. First, CF-FBA ingests experimental data and converts it into a standardized format. Next, CF-FBA uses the data to construct a group of cellular metabolic models and sample fluxes from those models to create a dataset. Then, CF-FBA performs dimensionality reduction to elucidate underlying trends in the experimental data. Finally, CF-FBA leverages those insights to reduce the original, overspecified models to standalone cell-free metabolic models.

4.1 Data gathering

In order to build a robust model that reflects biological reality, I first had to generate data to use for training. Gathering high quality biological data is difficult and is crucial to the success of the system as a whole. I describe the high level process of creating a cell-free system and running the experiments below. The full experimental protocol can be found at dx.doi.org/10.17504/protocols.io.kz2cx8e.

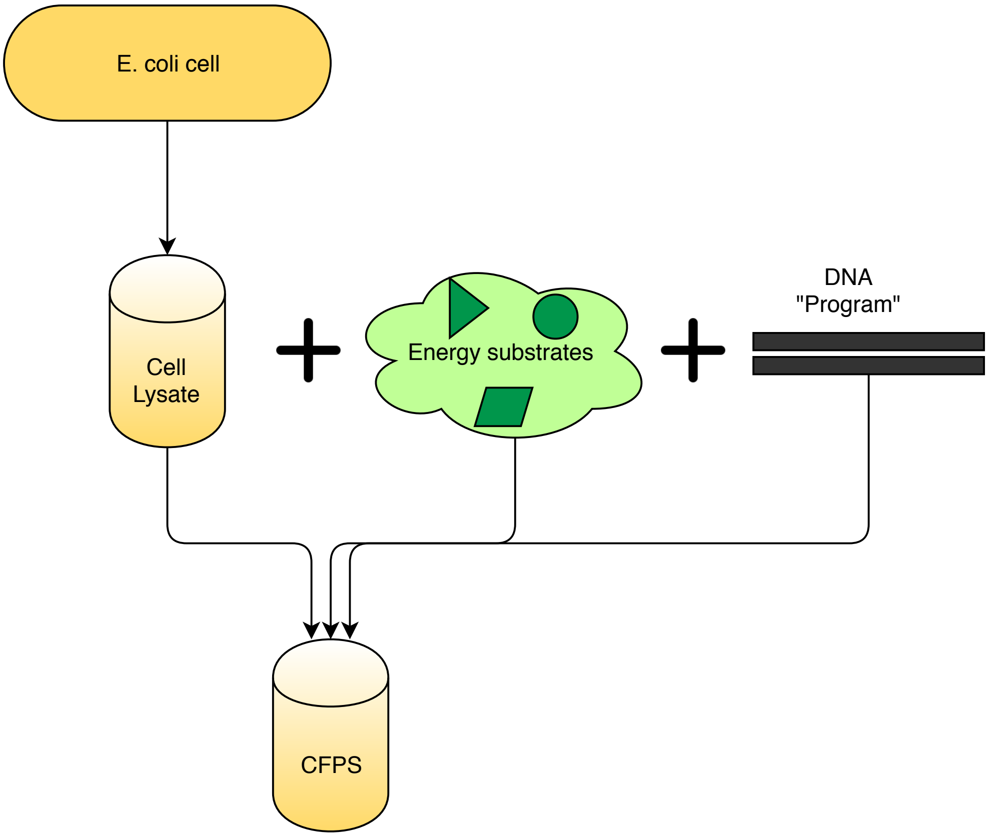


Figure 4.1: Process for creating a CFPS system. The *E. coli* cells can be grown up in bulk, allowing future preparation of CFPS reactions to be far quicker than typical *in vivo* methods.

4.1.1 Cell-free systems

As shown in Figure 4.1, creating a cell-free protein expression system involves three main steps: growing and lysing cells, supplementing with energy substrates, and adding custom DNA constructs. This protocol is based on an open protocol from the Federici lab [52]. I began by growing up 1L of BL21 *E. coli* cells in an overnight culture of Luria Broth (LB) until they reached an optical density (OD) of 1.6. The cells were then split into 45 millileter (mL) aliquots centrifuged for 12 minutes at 4500g and the supernatant is poured off. The pellet was then be stored in a 4° refrigerator overnight if necessary. Next, the cells underwent 2 more sets of centrifugation with the same settings; the pellet was resuspended each time with 45 mL of S30A buffer (see Appendix A for buffer compositions). The cells were then centrifuged twice more at 2000g for 8 and 2 minutes. In between the final two centrifugations, the cells were resuspended with 37.5 mL of S30B buffer. After the final centrifugation step, the pellets were weighed and resuspended with 0.9 times their weight in S30A and 5 times their weight of 0.1 mm beads. Using a bead beater set to 46 rpm, the cells were beaten for 30 seconds, rested on ice for 1 minute, and then beaten for another 30 seconds. These lysed cells were spun one final time for 8 minutes at 4500g to remove the cellular debris and beads. The supernatant was then removed as a cell extract and stored in a -80° freezer.

This cell extract contains the core cellular machinery that is necessary for transcription and translation. However, the full CFPS system requires additional reactants that are important for the transcription and translation reactions. The addition of these reactants to the cell extract creates a full CFPS system. These substrates include cofactors such as cyclic adenosine monophosphate (cAMP), maltodextrin, and nicotinamide adenine dinucleotide (NAD), as well as vital building blocks of transcription and translation such as nucleoside triphosphates (NTPs), Amino acids (AAs), and transfer RNAs (tRNAs). Table 4.1 lists each of the reactants and their purpose in the cell-free system, while Table A.1 shows the final concentration for each reactant.

This cell-free expression system acts a biological computer. Whatever DNA "program" is added, the system will work to produce the appropriate protein result. This platform is powerful because it allows a user to easily insert these DNA programs and see results within a few hours. A typical biological timeline would take far longer because living cells would have to be transformed (a lengthy process) to uptake the DNA and incorporate it into their production processes.

Reactant	**Amount** (μL)	**Purpose**
Maltodextrin solution	5	Energy substrates
Amino acids	10	Building blocks for TL
Polyethylene glycol (PEG)	2.5	Molecular crowding
Hexose monophosphate (HMP)	1	Phosphate source
Magnesium (Mg)	0.74	Important cofactor
Potassium (K)	0.8	Charge homeostasis
DNA	0.8	Template for protein
Cell extract	25	TX/TL machinery
CFPS system	50	Protein production

Table 4.1: List of reactants for a 50 μL CFPS reaction. See the final concentrations of all reactants in Table A.1. Note that MDX is a mixture of substrates (also detailed in Table A.1). Amino acids includes all 20 of the amino acids.

Sugar	HMP	NTPs	K	Normalized output
0	0	1	0	0.57
0	1	0	0	0.52
1	0	0	0	0.59
0	0.5	0.5	0	0.97
0.5	0	0.5	0	0.76
0.5	0.5	0	0	0.20
0.25	0.25	0.5	0	0.40
0.25	0.5	0.25	0	1.00
0.5	0.25	0.25	0	0.96
0.25	0.25	0.25	0.25	0.36
0	0	0	1	0.19
0	0	0.5	0.5	0.22
0	0.5	0	0.5	0.16
0.5	0	0	0.5	0.16
0	0.25	0.25	0.5	0.37
0.25	0	0.25	0.5	0.47
0	0.5	0.25	0.25	0.26

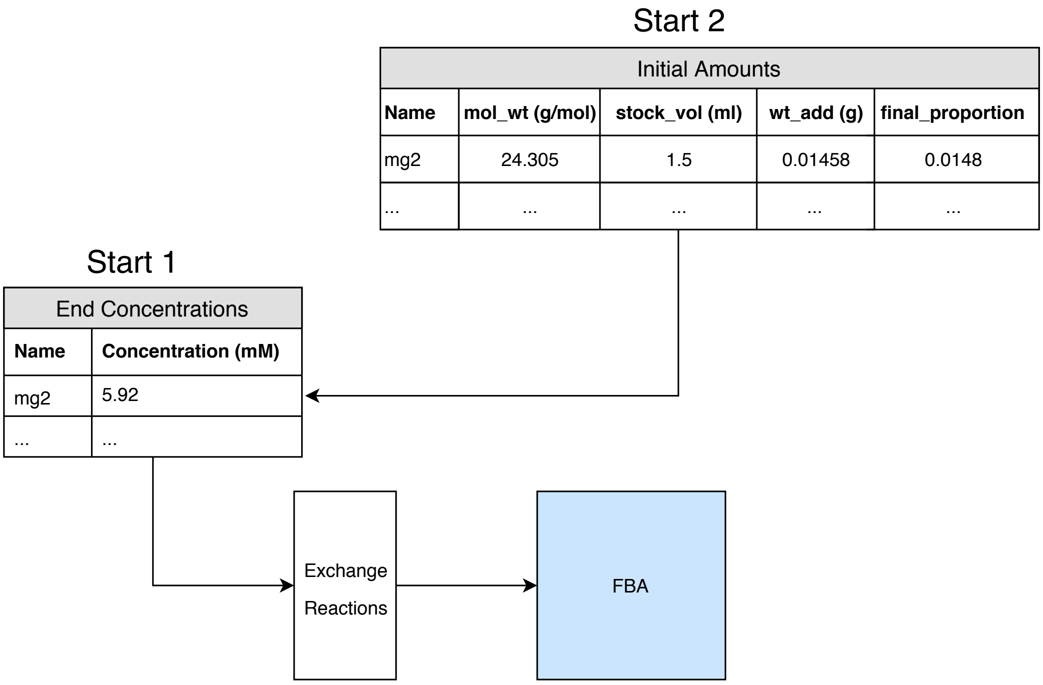
Table 4.2: Different reaction concentrations for our Manual dataset. Reactant columns are measured in μL, while the output column has been normalized by the maximum level of fluorescence observed. Each experiment was repeated $n = 2$ times at a total volume of 6μL.

4.1.2 Datasets

The protocol above was used to create two datasets for training. The first dataset was created by hand and followed standard lab procedures. Each of the reactants specified by Table 4.1 was combined in the appropriate ratio to create a 200 μL mastermix. That mastermix was then split into 5 μL aliquots. 1 μL of varying concentrations of the energetic substrates were added to each aliquot, bringing the total reaction volume to 6 μL . This dataset is composed of 17 differing ratios of sugar, phosphate, potassium, and nucleotides. I used a DNA circuit that encodes the red fluorescent protein (RFP) gene, so the amount of protein production was measured using fluorescence readout with a plate reader. The results for those reactions are shown in Table 4.2

One of the downsides to generating this data by hand is that creating large datasets by hand takes a long time. To increase the amount of data generated, I created a second dataset using a Labcyte Echo acoustic liquid handler. The Echo is an acoustic liquid handler that can programmed to combine different amounts of liquid. Using the Echo, I was able to generate a larger number of different experimental conditions for the CFPS reactions. Additionally, pipetting by hand has intrinsic error, while the Echo is able to transfer liquids in multiples of 2.5 nanoliter (nL) with less than 2% error. The Echo therefore has the dual benefits of generating more data and reducing the amount of noise in that data. The experiments on the Echo were performed using 2μL of CFPS system and 500 nL of additional reactants. The use of automation allowed us to test 51 different reaction conditions and shows that this could be expanded to an even larger scale.

Finally, I also used a third dataset from the recent Karim and Jewett paper that uses CFPS systems to perform metabolic engineering [53]. Their protocol is very similar to the one I showed above, though the final composition of the reactants differs. This dataset is useful because it was created in a different lab and they measured the output using liquid chromatography, not fluorescence. Since this dataset was created in different lab conditions, it can be used to check that our system is generalizable to cell-free systems as a



Initial Amounts

Name	mol_wt (g/mol)	stock_vol (ml)	wt_add (g)	final_proportion
mg2	24.305	1.5	0.01458	0.0148
...	...	...	...	...

End Concentrations

Name	Concentration (mM)
mg2	5.92
...	...

Figure 4.2: Two possible ways to incorporate the experimental setup in CF-FBA. In Start 1, the user already knows the final concentrations of each reactant and can upload that as a CSV. Otherwise, the user can upload information about the reactants that are added and our tools will automatically calculate the final concnetrations of each reactant.

whole and not just overfitting data generated from my CFPS system.

We will refer to these datasets as our "Manual", "Echo", and "Karim" datasets.

4.2 Data ingestion and incorporation

4.2.1 Data ingestion

Since this system is intended to aid biologists in their experiments, I designed it to be usable without any coding experience. With that in mind, I created two separate ways to incorporate the experimental setup into the models.

Figure 4.2 shows the acceptable structures of the experimental setup . The first structure is a CSV containing the final concentrations of the different reactants in the cell-free system. This is the easiest to incorporate because it is a simple conversion from concentrations to fluxes. Fluxes have the unit mol/min/gDW, so they can be converted from metabolite concentrations into fluxes using the following equation from MetaboTools [54]:

$$f = \frac{m}{c * t * 1000.0} \tag{4.1}$$

where f is the flux value, m is the concentration of our metabolite, and c is the overall concentration of the cell. I used $t = 24$ because our reactions ran for 24 hours. Using this equation, CF-FBA is able to ingest a CSV mapping the name of the reactant to the final concentration and turn them into flux constraints.

However, CF-FBA also supports users who might only know the relative amounts of each reactant that they are adding, but not the final concentrations of their reactants. This is important because many biologists may have a protocol that they follow that tells them how to make a CFPS system, but they may not know the final concentrations in their mixture. CF-FBA allows those users to use a CSV (or combination of CSVs) that contains information about each reactant used to make the system. Each row has the name of the reactant, its molecular weight, the weight used to create the stock, the volume used to create the stock, and the final volume of the stock added to the cell-free reaction. Using this information, CF-FBA calculates the final concentration of each reactant in the CFPS system and converts it to the same form as the other branch of the pipeline. Once the data is in its canonical form of final concentrations, CF-FBA proceeds with the a unified computational pipeline.

4.2.2 Data Incorporation

After ingesting the experimental setup, CF-FBA incorporates it into a FBA model. The FBA models are handled via the COBRApy python package [55]. CF-FBA uses the iJO1366 model for its base model of *E. coli*. iJO1366 consists of 1805 metabolites and 2583 reactions [31]. This model describes *E. coli* cells, and needs to be adapted to cell-free systems because they no longer have the same physical structure as *E. coli* cells. First, CF-FBA changes every reaction reflect the fact that there are no longer any membranes or cellular compartments. Every reaction that contains a periplasmic or extracellular component is moved into the cytosol. This is handled by replacing every metabolite in the model with its corresponding cytosolic version. If no corresponding cytosolic metabolite exists, CF-FBA creates a cytosolic version of the metabolic and then proceeds as above. After this step, all of the reactions in the model occurring in the same compartment, which reflects how a CFPS system works.

CF-FBA then handles the specific experimental data that was earlier ingested through the use of exchange reactions. Exchange reactions are pseudo-reactions that allow a FBA model to constrain the amount of a reactant present in the model. For each of the reactants in a CFPS system, CF-FBA creates an exchange reactions with bounds based on the fluxes that were calculated earlier. At the end of this process, CF-FBA has converted the full-scale *E. coli* model to better approximate a CFPS system. This type of model has not been generated before, but it is still overspecified. This model is what is later used to reduce into a base cell-free model.

CF-FBA also provides the option to explicitly incorporate the transcription and translation reactions that are so crucial to CFPS systems. Most FBA models do not explicitly model the transcription and translation reactions. However, earlier work from the Varner lab incorporated these reactions into a FBA model specifically to try to model a cell-free system [14]. This earlier work created the FBA model by hand and is written in Julia. CF-FBA provides auxiliary tools to, given the sequence of the gene of interest, automatically generate a relevant version of the Varner model. That model is converted to Python and the relevant transcription and translation reactions are extracted. Those reactions are then incorporated into the base cell-free model, creating a TXTL cell-free model. ¡¡¡¡¡¡¡ HEAD =======

¿¿¿¿¿¿¿ d36f4c58ce53756c4208d542755ca45e64177c23

4.3 Dataset generation

4.3.1 Model generation

Once CF-FBA has created these base models, it uses them to create a different model for each experimental condition. CF-FBA reads in a CSV consisting of the experimental conditions and a quantitative output. Each row represents a different experiment and each column contains the amount of additional reactant that was added. CF-FBA then re-calculates the final concentrations of each of the reactants for each experimental starting condition. Those new concentrations are used to update the constraints for the exchange reactions of each reactant. For each unique experimental condition, CF-FBA creates a new FBA model.

These experimental condition models are still overspecified and therefore have the same optimal solution. Thus, these naive models poorly describe the biological reality of the datasets (see Section 5.3 for a comparison). Clearly, FBA does not describe what is occurring in a CFPS system. These FBA models contain every reaction that is occurring in a steady state bacterium. While the bacteria are harvested at steady state, not all reactions will maintain their importance in a cell-free system. I hypothesized that these base models failed to describe cell-free systems because the differences between a cell-based system and a cell-free system cannot be encapsulated solely through different exchange reactions and the creation of a single-compartment system.

FBA acts as a coarse, non-linear amplifier, so points that vary only slightly in the input parameter space may not differ very much in the output space.

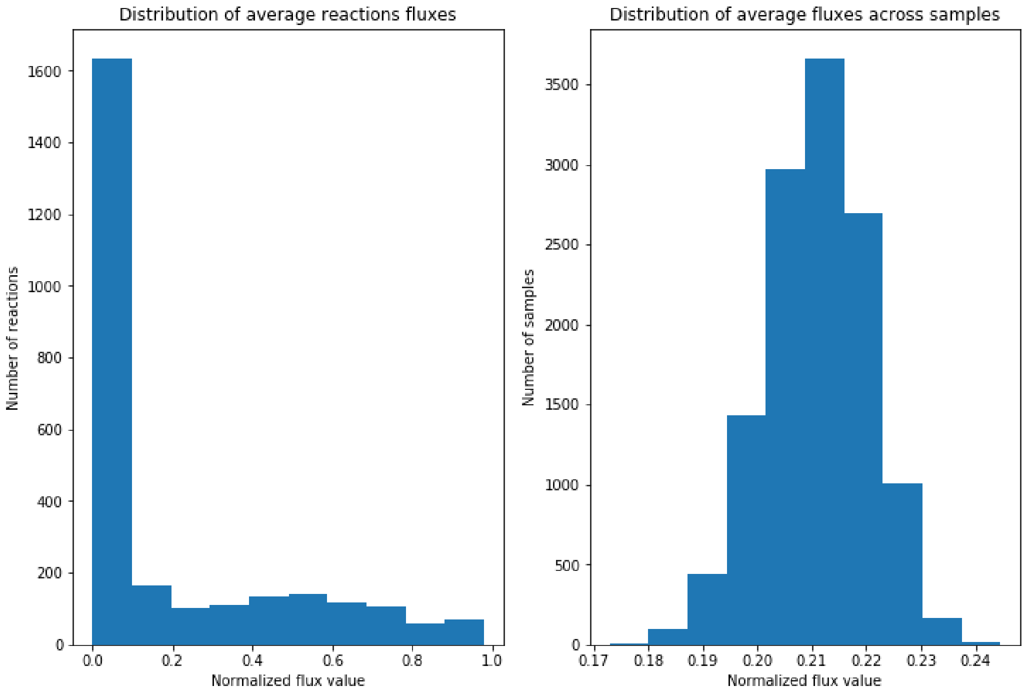


Figure 4.3: Distribution of fluxes generated by flux sampling. X axes represent the value of the fluxes and have been scaled to between 0 and 1. The plot on the left shows a histogram of the average flux value for each reaction. Most reactions have an average flux that is close to 0. The right plot shows the average flux across samples. The average flux across all samples clusters around 0.21.

I wanted to improve the granularity of FBA response by passing the fluxes through a dimensionality reduction technique such as a VAE. In order to do this, I needed a large dataset of fluxes to train on. Although the experimental conditions led to only have a small number of models, I was able to create large datasets by flux sampling from each model.

4.3.2 Flux sampling

Flux sampling is the process of sampling possible fluxes through each reaction. CF-FBA performs flux sampling on each model of a different experimental condition. For each model, this generates a distribution of fluxes for every reaction in that model. Figure 4.3 shows the average distribution of fluxes for each sample and each reaction. Flux sampling is implemented using the optGpSampler routine [56].

Each model has different experimental conditions and therefore has slightly different flux distributions for each reaction. So, after generating flux samples for each model, these sampled fluxes are combined to generate two new datasets. One dataset is flat—meaning we simply concatenate all of the sampled fluxes into a single dataset. This dataset has size $(N * E) \times R$ where N is the number of samples, E is the number of experiments, and R is the number of reactions. This flat dataset is useful to investigate the fluxes using typical dimensionality reduction techniques. However, CF-FBA also generates a stacked dataset of size $N \times E \times R$. This involves sampling with replacement from the distribution of fluxes of each model and then stacking the fluxes to create a new dataset. Using this version of the dataset facilitates the training of a Corr-VAE. I used this part of CF-FBA to generate flat and stacked versions of all three of the datasets described above.

4.4 VAE

As mentioned earlier, the base cell-free FBA models did not describe our experimental data well. To solve that issue, CF-FBA implements a VAE to reduce the dimensionality of our models. The VAE is implemented using the Keras framework [57] and the implementation was designed to be as modular as possible. Because the Corr-VAE has applications in other parts of biology, I implemented all VAEs to make it easy to change their parameters. Thus, my implementation takes in command-line arguments to specify either a flat or stacked dataset, a list of layer sizes, the number of latent dimensions, how to scale the inputs, and whether or not to use the custom correlation loss function. I also provide sensible default values for those users who have less experience in the field of deep learning.

The structure of the VAEs we used are as follows. I experimented with two different layer structures, a 2-layer VAE with layer sizes 1024 and 256 and a 3-layer VAE with layer sizes 1024, 1024, and 1024. The 3-layer VAE performed much better, so that was the structure I used for all subsequent experiments. I also tried using latent dimensions of size 2 and 10. Both 2 and 10 dimensions are able to capture relevant information in their latent space, though future work could use even more latent dimensions. Activations between layers of both the encoder and the decoder were Rectified Linear Units (ReLUs) [58]. The final layer of the VAE used a sigmoidal activation to produce scaled fluxes between 0 and 1.

4.4.1 Corr-VAE

Placing a regular VAE on top of a FBA model will simply attempt to reconstruct the fluxes. However, I did not just want to reconstruct fluxes in an unbiased way—I wanted the reconstructed fluxes to correspond to the real-world biological data. The key insight of a Corr-VAE is that the experimental data can be used to perturb the latent space to better reflect biological reality. A Corr-VAE incorporates the experimentally determined output values into its loss function. It does so by adding a loss term consisting of the correlation between the experimental data and the reconstructed flux that represents the objective function. Recall Equation 2.3 that described the normal VAE loss function. The Corr-VAE loss function is now written as:

$$\mathcal{L}(\theta, \phi) = -E[\log p_\theta(x_i|z)] + KL(q_\phi(z|x_i)||p(z)) + corr(x'_i, d) \tag{4.2}$$

where x'_i is the reconstructed data, d is the experimentally determined biological data, and *corr* is the Pearson correlation between the two. I also experimented with weighting the correlation loss term to force the representation to be closer to the real world data, but this did not appear to have a large effect. Corr-VAEs required a stacked dataset in order to calculate the correlation between the experimental data and the output fluxes. By default CF-FBA uses a Corr-VAE to reduce the dimensionality of a generated dataset.

4.5 FBA model reduction

The Corr-VAE is able to generate a lower dimensional representation of the base cell-free models. CF-FBA uses the reconstructed representation to create better cell-free models. After examining the latent space of the Corr-VAE to ensure that it was learning the differences between the models, I was able to use the reconstructed fluxes to reduce the original models. For each experimental condition model, CF-FBA uses the reconstructed fluxes from the Corr-VAE to determine which reactions are unimportant. It does this by creating a threshold and removing any reactions that have a flux under that threshold. By doing this for each model, it builds up a list of reactions that are not used in any of the experimental models.

CF-FBA then uses removes all of those reactions from the base model to generate a new cell-free model. The value of this model is tested by comparing how well the optimal fluxes correlate with the experimental data. Section 5.3 shows that these reduced models with the differential conditions give us better explanations of experimental data than full GEMs. The thresholding is a simple method of identifying which reactions are not important in cell-free systems. Future work could generate improved reduced models by using more sophisticated reduction techniques in conjunction with the Corr-VAE.

Chapter 5

Results

CF-FBA is able to produce biologically relevant dimensionality reductions that allow us to make better reduce models for CFPS systems. This chapter demonstrates the following:

- PCA and naive VAEs cannot learn good representations of different models, while a Corr-VAE can.

- The separation between the different models improves as the number of latent dimensions increases. This is replicated across different CFPS systems.

- As noise is added to the input dataset, Corr-VAE performance falls off, showing that it is learning real trends.

- A Corr-VAE can be used to reduce GEMs to create CFPS models that correlate better with experimental data.

5.1 Can a VAE distinguish between fluxes generated from different experimental conditions?

I first investigated whether or not a VAE can learn how changes in the experimental conditions affect the underlying CFPS system. To be useful for extracting information about the underlying cell-free system, a VAE would need to be able to distinguish between different starting experimental conditions. In order to assess how well a VAE is learning the differences between experimental conditions, I examined its latent space. If the VAE performs well, we should see a clear separation between fluxes generated from different experimental conditions. This set of experiments was run on the set of fluxes generated from our manual dataset. The PCA and regular VAE are trained on the flat version of the manual dataset, while our Corr-VAE is trained on the stacked version. All three techniques then project the same test dataset into 2 dimensions for visualization purposes. The VAE and the Corr-VAE have identical network structures as described in Section 4.4.

Figure 5.1 shows the latent space of a basic VAE as well as the first two principal components of the PCA. Both dimensionality reduction techniques fail to uncover the underlying structure of the fluxes. This demonstrates that naive applications of a VAE or PCA cannot learn the differences between the various starting experimental conditions.

However, a Corr-VAE is able to recover clear distinctions between the different experimental conditions. Figure 5.2 plots the latent space of our Corr-VAE. Whereas the experimental conditions cannot be recovered in the latent space of the VAE, the latent space of the Corr-VAE clearly distinguishes between the different starting conditions. While PCA is a popular dimensionality reduction technique in biology, these results suggest that Corr-VAE may be a more effective way to discover relationships in biological data.

The original flux datasets have around 2600 reactions, so using a latent space

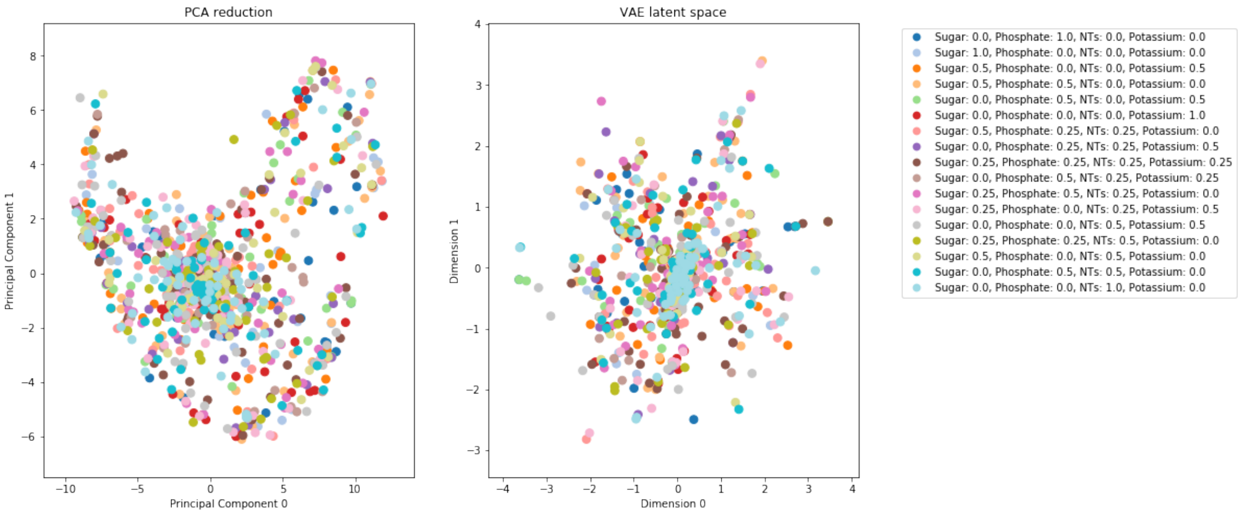


Figure 5.1: Left: first two principal components of a PCA on our test dataset. Right: the two latent dimensions of a trained VAE with the classic loss function. Different colors represent different experimental conditions. Neither PCA nor a basic VAE can learn a clear representation of the different models we are using.

of only 2 dimensions is a very severe dimensionality reduction. In addition to the Corr-VAE with latent dimension 2, shown above, I also trained a Corr-VAE with a 10 dimensional latent space. In order to visualize the latent space of this Corr-VAE, the 10-dimensional latent space was projected down to 2 dimensions using the t-Distributed Stochastic Neighbor Embedding (tSNE) method [59]. Figure 5.3 shows that using a 10-dimensional latent space with a Corr-VAE is also able to clearly separate different starting experimental conditions. The separation between models is even more clear in this case, and suggests that more latent dimensions allow for richer latent representations of each model.

I also show that this technique can generalize across different CFPS systems. Figure 5.4 is generated using the same 10-dimensional structure as the Corr-VAE in Figure 5.3, but on the Karim dataset instead of the Manual dataset.

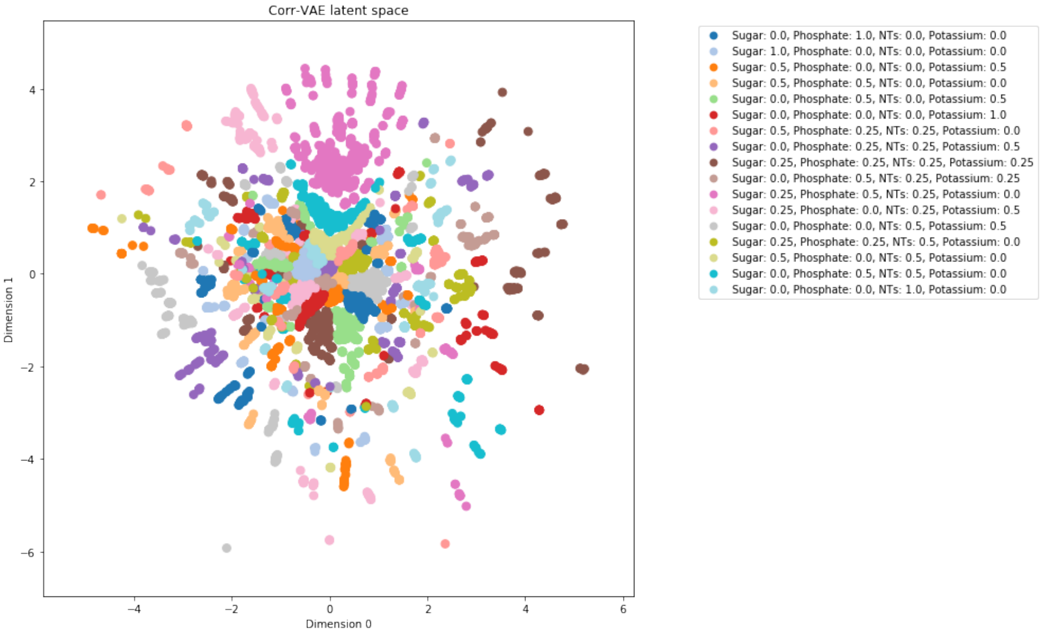


Figure 5.2: Different colors represent different experimental conditons. This plot shows the two latent dimensions of a trained Corr-VAE. Colors cluster together and show that a Corr-VAE can recover the original models.

5.2 Noise experiments

In order to validate that the Corr-VAE was performing as intended, I ran a number of experiments to ensure that these results were not due to noise. Figure 5.5 plots the standard deviations of each of the reactions for the test dataset as well as the reconstructed dataset. The reconstructed dataset maintains a very similar shape to the original dataset, while also reducing the standard deviation of fluxes with each reaction. This shows that the Corr-VAE is not drastically changing the structure of the fluxes to produce biologically impossible fluxes. Additionally, this plot implies that the Corr-VAE could also be used to de-noise our dataset.

Another potential issue was that the Corr-VAE was only learning how to artificially order the objective flux. Although the correlation error was smaller than the reconstruction error, the Corr-VAE could be generating correlations instead of discovering them, just to reduce its loss function. To show that

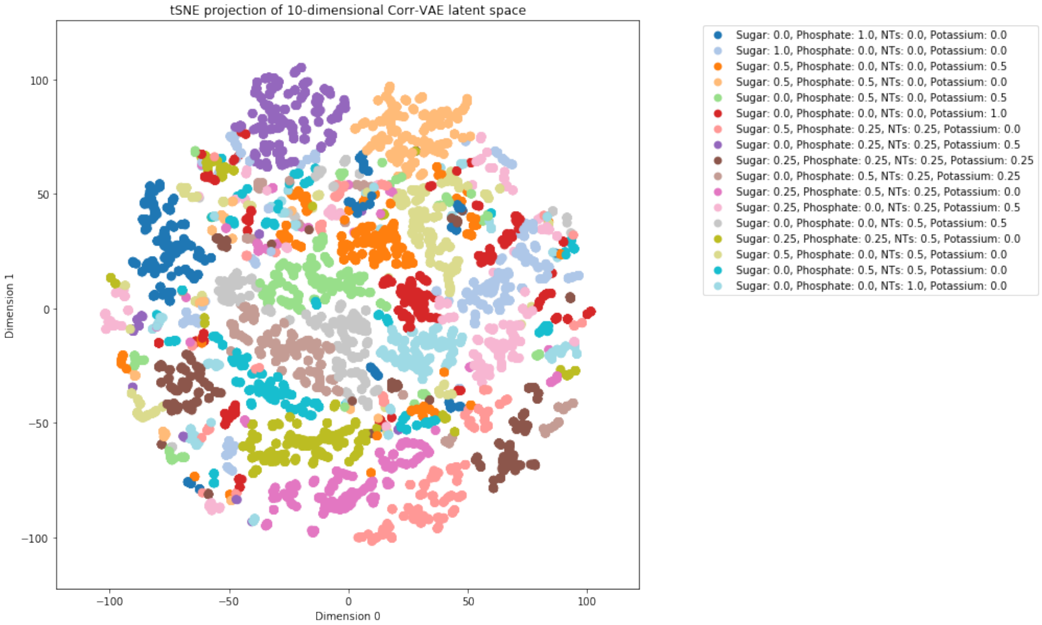


Figure 5.3: Different colors represent different experimental conditons. This plot shows a tSNE projection of the 10-dimensional latent space of a Corr-VAE trained on the Manual dataset. Adding more latent dimensions allows better reconstruction of the invidual models.

this was not the case, I generated a new, biased dataset where the objective fluxes were already highly correlated to our experimental data. Figure 5.6 shows what happens when we add noise to this biased dataset. The correlation between the starting data and the experimental data is around 0.78 and is shown by the green line. After running the biased fluxes through a Corr-VAE, the correlation is even higher, as evidenced by the blue line. But, as noise is added to the data by randomly drawing from a normal distribution and adding it to the flux for each reaction, the correlation begins to fall. This shows that the Corr-VAE architecutre does not create correlations with random data, but is learning the correlation from the underlying data.

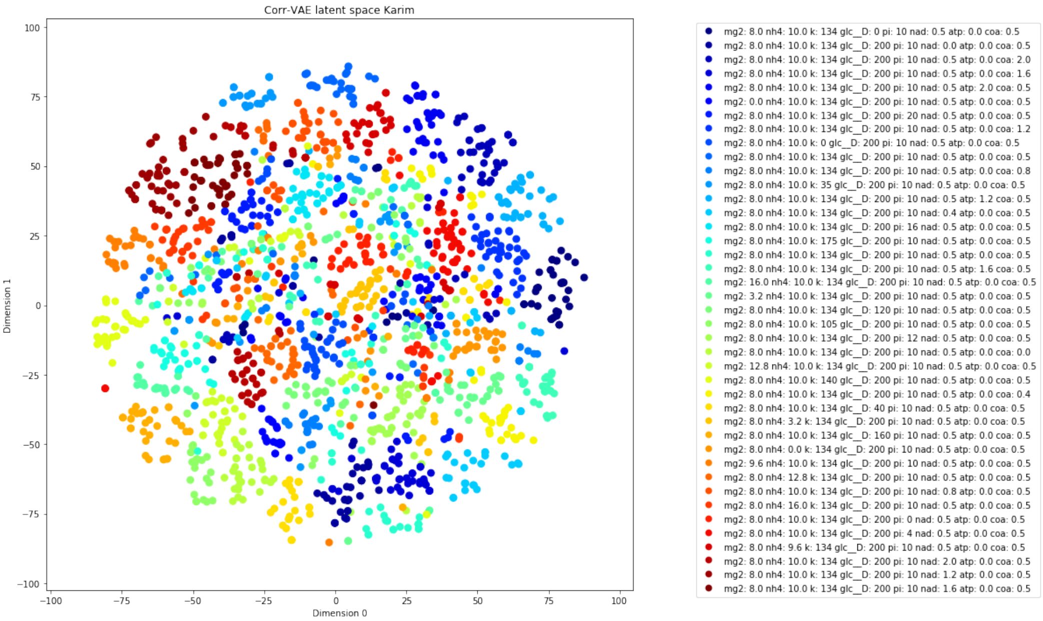


Figure 5.4: Different colors represent different experimental conditons. This plot shows a tSNE projection of the 10-dimensional latent space of a Corr-VAE trained on the Karim dataset. Our Corr-VAE is able to reconstruct the individual models for multiple types of CFPS systems.

5.3 Comparison of reduced models

A key goal for CF-FBA was to produce reduced models that could accurately explain experimental data. I tested this by solving for the optimal flux distribution for each model of our reaction conditions. A model's ability to explain the experimental data was then assessed using the correlation between the predicted output and the real biological output. Table 5.1 shows the results for each of our starting models that are at the GEM scale as well as some reduced models. As expected, we found that the full-scale GEMs have very low correlations, between 0.14 and 0.24. This models are intended to describe living *E. coli*cells and thus do a poor job of modeling CFPS systems.

We also compare our model to pre-existing pruned models that other algorithms have created. For instance, the ColiPruned model from NetworkReducer has a correlation of 0.34 with our biological data. This is better than

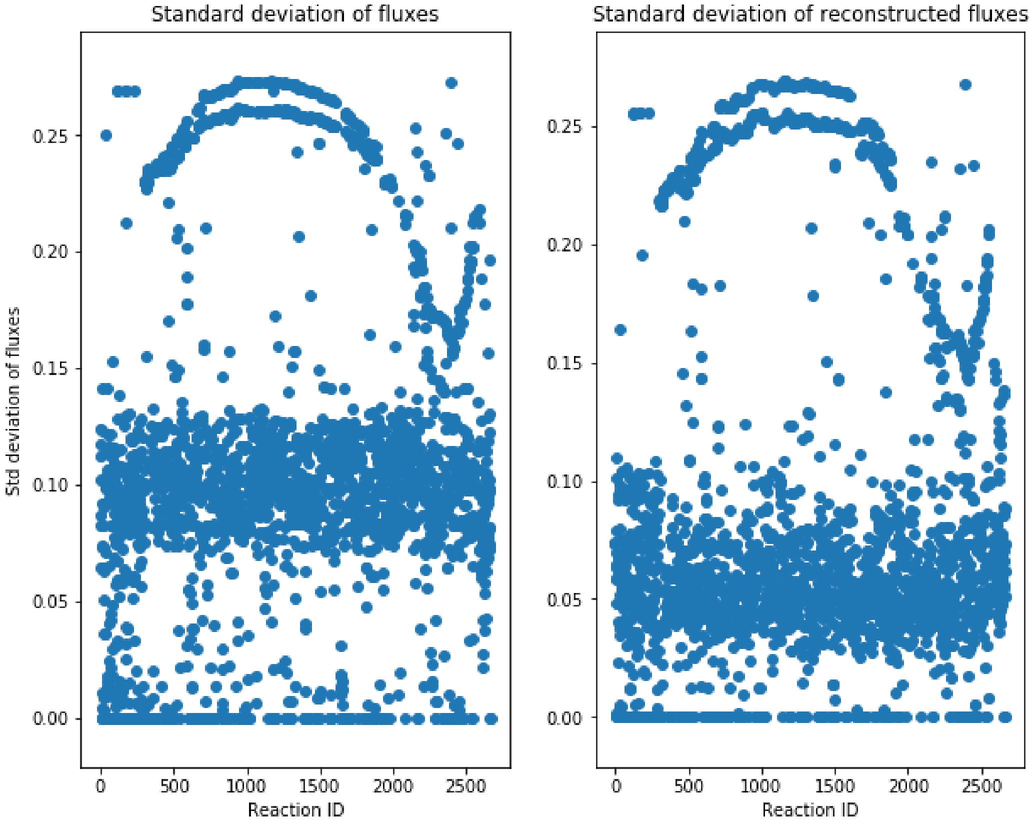


Figure 5.5: Reconstructed fluxes reduce the amount of noise while also maintaining the shape

the full-scale genome models, though it is not as good as our customized cell-free models. This make sense because the goal of NetworkReducer is to reduce to a minimal core model, not to specify it for cell-free systems. Our system produces models that with more than 2x better correlations than a full-scale GEM. While these correlations show that we still cannot perfectly describe the biological system, the reduction is far better than a full-scale model.

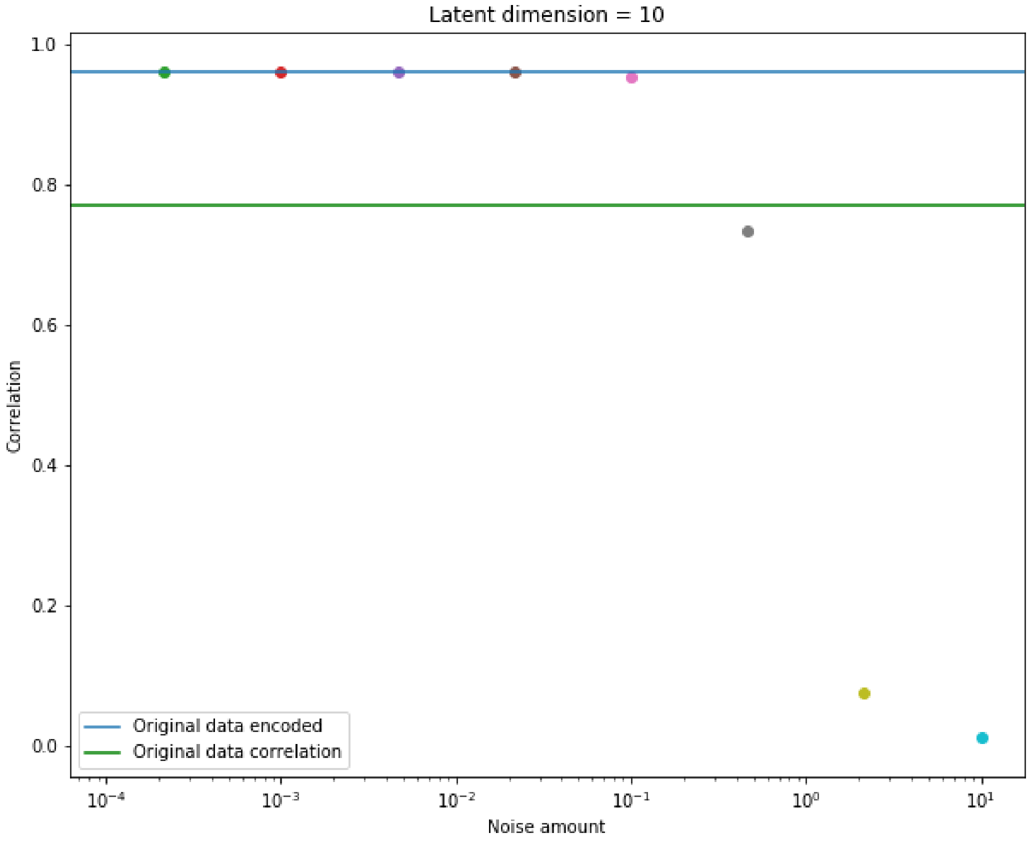


Figure 5.6: Adding noise to the data reduces the correlation from our VAE

Dataset	Uses Corr-VAE?	Correlation
echo_txtl	No	0.22
echo	No	0.14
manual_txtl	No	0.23
manual	No	0.24
karim	No	0.17
coli-pruned	No	0.34
echo-reduced	Yes	**0.54**
manual-reduced	Yes	**0.50**
karim-reduced	Yes	**0.43**

Table 5.1: Comparison of how well different models correlate with experimental data. Models that have been produced by CF-FBA end in 'reduced'. Full-scale GEMs perform the worst, while a reduced model such as the ColiPruned model from NetworkReducer performs slightly better. Although the reduced models genereated by CF-FBA cannot fully explain the experimental data, they are able to explain the data better than any other models.

Chapter 6

Future Work

6.1 Further datasets

The datasets that used in this dissertation are relatively small—on the order of 10s of different parameter combinations. The limited scope of these datasets means that they are only able to coarsely cover the entire parameter space. I hypothesize that one way to get better reduced models would be to train using data generated from more of the parameter space. Generating large biological datasets by hand is difficult and error prone. However, this dissertation demonstrated the potential for further lab automation by generating a dataset using the Labcyte Echo liquid handler. Future work will take this even further and generate datasets with hundreds or even thousands of starting conditions. Another logical extension is to use datasets generated from organisms other than *E. coli*.

6.2 Different Organisms

This work has only explored the generation of *E. coli* cell-free models. However, much ongoing work involves the use of non-*E. coli* organisms. This system was written in such a way that it can be applied to any organism

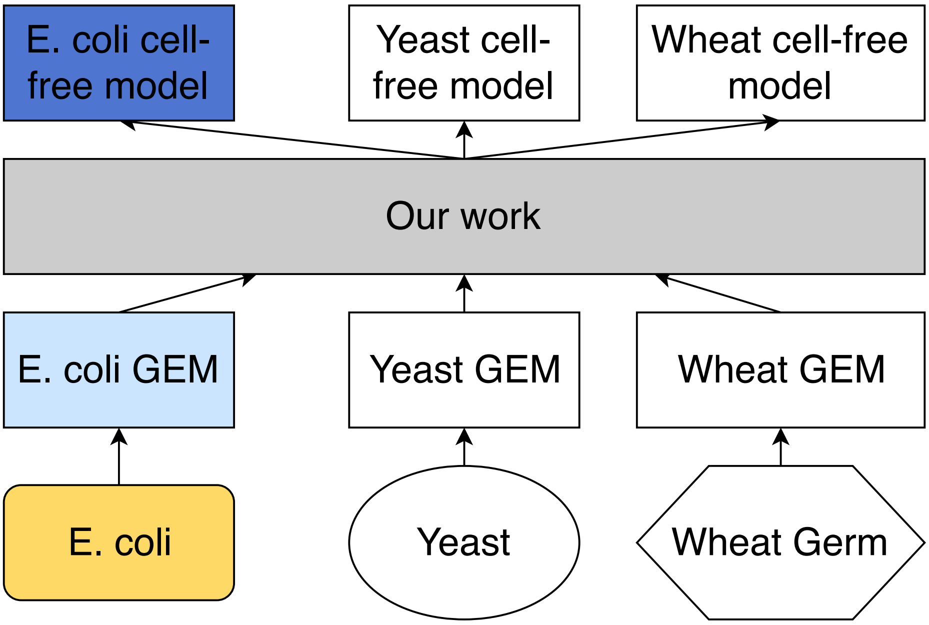


Figure 6.1: CF-FBA was designed to convert any organism's GEMs into an appropriate cell-free model.

with a GEM. Figure 6.1 demonstrates one potential vision of how CF-FBA could be used in the future. Instead of creating a model by hand that only describes our CFPS system, CF-FBA can be used to create cell-free metabolic models for many different organisms. In particular, I am also a part of an OpenPlant grant involving modeling wheat-germ cell-free systems. The data is currently being generated, and I plan to apply CF-FBA to this new CFPS system as soon possible.

6.3 RL system for reduction

The use of a VAE as a dimensionality reduction tool proved to be quite powerful, but the part of the system that converted from the reduced representation back into a FBA model was quite simple. CF-FBA removes reactions based on a simple thresholding criterion. Instead of deciding which reactions to remove using a threshold, future version of CF-FBA could reframe this problem as a reinforcement learning problem. Using a combination of experimental data, a starting FBA model, and a latent representation of the problem space, these more sophisticated search methods could likely produce better reduced models. To that end, I have also built a framework on top of OpenAI's Gym environment for performing reinforcement learning on FBA models. Given a starting model with almost 2600 reactions, the search space of possible reduced models is enormous. However, a combination of the Corr-VAE and reinforcement learning could perform a targeted search and yield a better reduced model.

6.4 Batch variation

Finally, an important use case for CF-FBA is as a way to deal with batch variation in CFPS systems. Batch variation is an important problem facing the cell-free community [11, 60]. The current solution to this problem is to run everything that needs to be compared in the same batch. This is problematic, though, if the batch runs out or someone else wants to reproduce this result. The only way to deal with this problem would be to redo all of the experiments in a new batch.

Figure 6.3 demonstrates how CF-FBA provides a potential solution to this issue of batch variation. A researcher runs a calibration set of experiments and then uses those experimental results to generate a VAE model. Now, this VAE can be used to project any future experimental data into the same latent space as the first set of experiments. These experiments could be compared within the latent space, or they could be transformed back into the original dimensionality of the data and then compared. This transformed data should allow for better comparisons across batches.

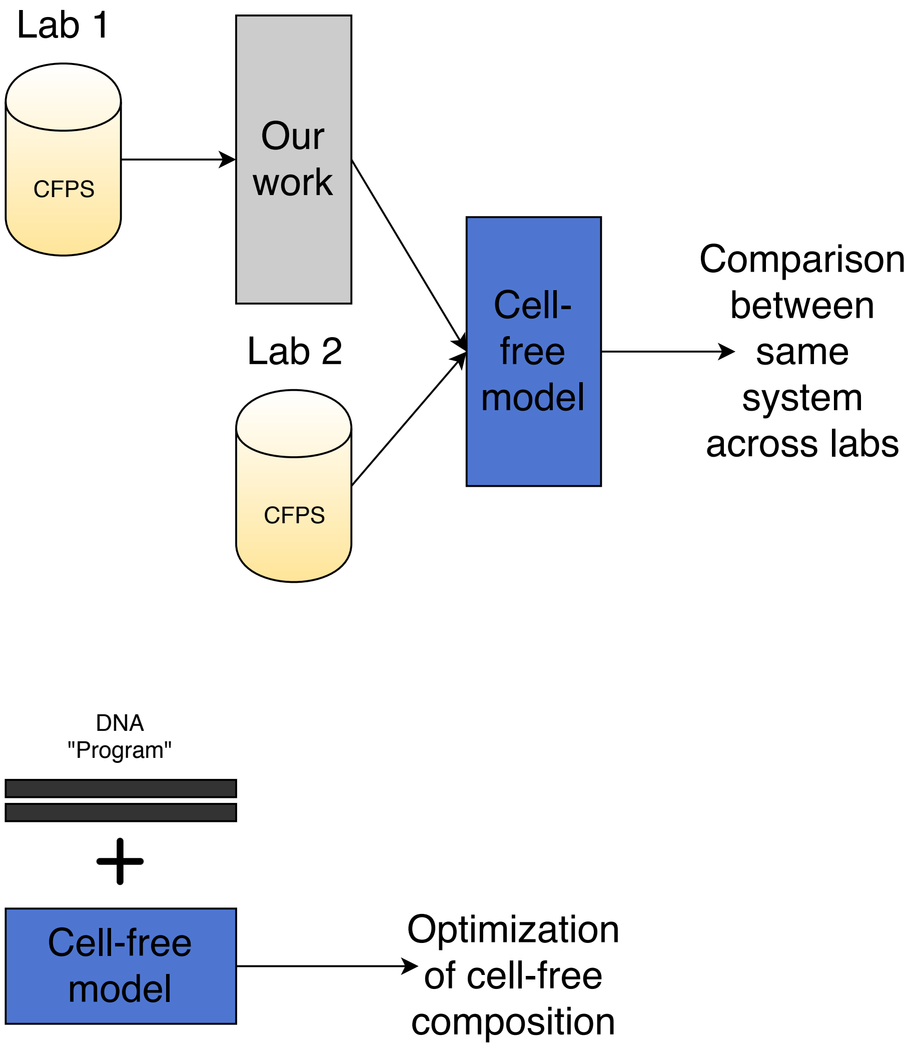

Lab 1
CFPS
Our work
Lab 2
CFPS
Cell-free model
Comparison between same system across labs
DNA "Program"
+
Cell-free model
Optimization of cell-free composition

Figure 6.2: Two future applications of CF-FBA. Above, multiple experiments can be projected in the same subspace to better compare experimental results. Below, the system can help optimize the energetic composition of a cell-free system

Chapter 7

Conclusion

The goal of this dissertation was to provide a metabolic model for CFPS systems. I produced a system, CF-FBA, that accomplished that goal by generate these reduced metabolic models from already existing, genome-scale models. The models generated by CF-FBA explain real experimental data better than currently existing models. The system I created was used to generate *E. coli* CFPS models, but can be applied to other types of organisms. Along the way, I developed a new type of loss function for VAE and showed that it is more effective at discovering patterns in the experimental data than PCA.

In conclusion, I have shown that deep learning techniques can be successfully applied to biology, specifically metabolic modeling. Computational biologists should start thinking about using AEs (especially VAEs) for dimensionality reduction instead of defaulting to using PCA. In particular, the new loss function that I developed for my VAE is general enough to be used with any biological data and provides a good starting point for future work. I hope this work encourages other computational biologists to continue work integrating deep learning with biology and metabolic models.

Appendix A

Experimental Setup

A.1 Buffers

To create S30A buffer, combine 10.88g Mg-glutamate, 24.39g mM K-glutamate, 50 mL Tris and titrate with acetic acid to reach pH 7.7. Add water to 2L and autoclave. Before using, add 4 ml of 1 M DTT.

For S30B buffer, combine 10.88g mM Mg-glutamate, 24.39g mM K-glutamate, and titrate with 2M Tris until pH 8.2. Add water to 2L and autoclave. Add 2 ml of 1 M DTT before using.

A.2 Supplementary Tables

Sugar	AAs	NTPs	Normalized output
0	0	200	0.43
12.5	12.5	187.5	0.48
25	25	175	0.53
37.5	37.5	162.5	0.95
50	50	150	0.68
62.5	62.5	137.5	0.83

75	75	125	0.54
87.5	87.5	112.5	0.45
100	100	100	0.43
112.5	112.5	87.5	0.48
125	125	75	0.48
137.5	137.5	62.5	0.87
150	150	50	0.77
162.5	162.5	37.5	0.86
175	175	25	0.81
187.5	187.5	12.5	0.69
200	200	0	0.43
0	200	100	0.79
12.5	187.5	112.5	0.65
25	175	125	0.52
37.5	162.5	137.5	0.68
50	150	150	0.49
62.5	137.5	162.5	0.53
75	125	175	0.82
87.5	112.5	187.5	0.92
100	100	200	0.65
112.5	87.5	0	0.47
125	75	12.5	0.47
137.5	62.5	25	0.66
150	50	37.5	0.54
162.5	37.5	50	0.44
175	25	62.5	0.71
187.5	12.5	75	0.89
200	0	87.5	1.00
0	100	0	0.58
12.5	112.5	12.5	0.42
25	125	25	0.60
37.5	137.5	37.5	0.70
50	150	50	0.57

62.5	162.5	62.5	0.53
75	175	75	0.75
87.5	187.5	87.5	0.68
100	200	100	0.65
112.5	0	112.5	0.44
125	12.5	125	0.57
137.5	25	137.5	0.59
150	37.5	150	0.72
162.5	50	162.5	0.69
175	62.5	175	0.42
187.5	75	187.5	0.61
200	87.5	200	0.75

Table A.2: Different reaction concentrations for our Echo dataset. Reactant columns are measured in nL, while the output column has been normalized by the maximum level of fluorescence observed. Each experiment was repeated $n = 2$ times at a total volume of 2.5 μL

Compound	Concentration (Micromolar (mM))
HMP	0.981
Mg	5.92
K	48.0
NAD	0.454
Coenzyme-A (CoA)	0.353
cAMP	1.01
Folinic acid	0.092
Spermidine	0.181
Glucose	5.42
ATP/GTP	2.09
CTP/UTP	1.26
AAs	35.6
tRNA	0.010
DNA	0.071

Table A.1: Final concentrations of reactants in our CFPS reaction

Bibliography

[1] Jay D Keasling. Synthetic biology and the development of tools for metabolic engineering. *Metabolic engineering*, 14(3):189–195, 2012.

[2] Jan Schellenberger, Richard Que, Ronan MT Fleming, Ines Thiele, Jeffrey D Orth, Adam M Feist, Daniel C Zielinski, Aarash Bordbar, Nathan E Lewis, Sorena Rahmanian, et al. Quantitative prediction of cellular metabolism with constraint-based models: the cobra toolbox v2. 0. *Nature protocols*, 6(9):1290, 2011.

[3] Jeffrey D Orth, Ines Thiele, and Bernhard Ø Palsson. What is flux balance analysis? *Nature biotechnology*, 28(3):245, 2010.

[4] Adam M Feist and Bernhard Ø Palsson. The growing scope of applications of genome-scale metabolic reconstructions using escherichia coli. *Nature biotechnology*, 26(6):659, 2008.

[5] E_ Almaas, B Kovacs, T Vicsek, ZN Oltvai, and A-L Barabási. Global organization of metabolic fluxes in the bacterium escherichia coli. *Nature*, 427(6977):839, 2004.

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

[7] Francis Crick. Central dogma of molecular biology. *Nature*, 227(5258):561, 1970.

[8] C Eric Hodgman and Michael C Jewett. Cell-free synthetic biology: thinking outside the cell. *Metabolic engineering*, 14(3):261–269, 2012.

[9] Joseph A Rollin, Tsz Kin Tam, and Y-H Percival Zhang. New biotechnology paradigm: cell-free biosystems for biomanufacturing. *Green chemistry*, 15(7):1708–1719, 2013.

[10] Erik D Carlson, Rui Gan, C Eric Hodgman, and Michael C Jewett. Cell-free protein synthesis: applications come of age. *Biotechnology advances*, 30(5):1185–1194, 2012.

[11] Zachary Z Sun, Clarmyra A Hayes, Jonghyeon Shin, Filippo Caschera, Richard M Murray, and Vincent Noireaux. Protocols for implementing an escherichia coli based tx-tl cell-free expression system for synthetic biology. *Journal of visualized experiments: JoVE*, (79), 2013.

[12] Richard Murray. Cell-free circuit breadboard cost estimate. `https://openwetware.org/wiki/Biomolecular_Breadboards:Protocols:cost_estimate`, 2013.

[13] Matthias Bujara and Sven Panke. In silico assessment of cell-free systems. *Biotechnology and bioengineering*, 109(10):2620–2629, 2012.

[14] Michael Vilkhovoy, Nicholas Horvath, Che-hsiao Shih, Joseph Wayman, Kara Calhoun, James Swartz, and Jeffrey Varner. Sequence specific modeling of e. coli cell-free protein synthesis. *bioRxiv*, page 139774, 2017.

[15] Marshall W Nirenberg and J Heinrich Matthaei. The dependence of cell-free protein synthesis in e. coli upon naturally occurring or synthetic polyribonucleotides. *Proceedings of the National Academy of Sciences*, 47(10):1588–1602, 1961.

[16] Kara A Calhoun and James R Swartz. Total amino acid stabilization during cell-free protein synthesis reactions. *Journal of biotechnology*, 123(2):193–203, 2006.

[17] Michael C Jewett, Kara A Calhoun, Alexei Voloshin, Jessica J Wuu, and James R Swartz. An integrated cell-free metabolic platform for protein production and synthetic biology. *Molecular systems biology*, 4(1):220, 2008.

[18] Yoshihiro Shimizu, Akio Inoue, Yukihide Tomari, Tsutomu Suzuki, Takashi Yokogawa, Kazuya Nishikawa, and Takuya Ueda. Cell-free translation reconstituted with purified components. *Nature biotechnology*, 19(8):751, 2001.

[19] Adam M Feist and Bernhard O Palsson. The biomass objective function. *Current opinion in microbiology*, 13(3):344–349, 2010.

[20] Lucas Theis, Wenzhe Shi, Andrew Cunningham, and Ferenc Huszár. Lossy image compression with compressive autoencoders. *arXiv preprint arXiv:1703.00395*, 2017.

[21] Pierre Baldi. Autoencoders, unsupervised learning, and deep architectures. In *Proceedings of ICML Workshop on Unsupervised and Transfer Learning*, pages 37–49, 2012.

[22] Eleftherios Terry Papoutsakis. Equations and calculations for fermentations of butyric acid bacteria. *Biotechnology and bioengineering*, 26(2):174–187, 1984.

[23] David A Fell and J Rankin Small. Fat synthesis in adipose tissue. an examination of stoichiometric constraints. *Biochemical Journal*, 238(3):781–786, 1986.

[24] RA Majewski and MM Domach. Simple constrained-optimization view of acetate overflow in e. coli. *Biotechnology and bioengineering*, 35(7):732–738, 1990.

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

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

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

[28] Aarash Bordbar, Jonathan M Monk, Zachary A King, and Bernhard O Palsson. Constraint-based models predict metabolic and associated cellular functions. *Nature Reviews Genetics*, 15(2):107, 2014.

[29] Amit Varma and Bernhard O Palsson. Metabolic capabilities of escherichia coli: I. synthesis of biosynthetic precursors and cofactors. *Journal of theoretical biology*, 165(4):477–502, 1993.

[30] JS Edwards and BO Palsson. The escherichia coli mg1655 in silico metabolic genotype: its definition, characteristics, and capabilities. *Proceedings of the National Academy of Sciences*, 97(10):5528–5533, 2000.

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

[32] Colton J Lloyd, Ali Ebrahim, Laurence Yang, Zachary Andrew King, Edward Catoiu, Edward J O'Brien, Joanne K Liu, and Bernhard O Palsson. Cobrame: A computational framework for models of metabolism and gene expression. *bioRxiv*, page 106559, 2017.

[33] Richard Bellman. *Dynamic programming.* Courier Corporation, 2013.

[34] Meric Ataman, Daniel F Hernandez Gardiol, Georgios Fengos, and Vassily Hatzimanikatis. redgem: Systematic reduction and analysis of genome-scale metabolic reconstructions for development of consistent core metabolic models. *PLoS computational biology*, 13(7):e1005444, 2017.

[35] Meric Ataman and Vassily Hatzimanikatis. lumpgem: Systematic generation of subnetworks and elementally balanced lumped reactions for the biosynthesis of target metabolites. *PLoS computational biology*, 13(7):e1005513, 2017.

[36] Philipp Erdrich, Ralf Steuer, and Steffen Klamt. An algorithm for the reduction of genome-scale metabolic network models to meaningful core models. *BMC systems biology*, 9(1):48, 2015.

[37] Annika Röhl and Alexander Bockmayr. A mixed-integer linear programming approach to the reduction of genome-scale metabolic networks. *BMC bioinformatics*, 18(1):2, 2017.

[38] Moritz von Stosch, Cristiana Rodrigues de Azevedo, Mauro Luis, Sebastiao Feyo de Azevedo, and Rui Oliveira. A principal components method constrained by elementary flux modes: analysis of flux data sets. *BMC bioinformatics*, 17(1):200, 2016.

[39] Sahely Bhadra, Peter Blomberg, Sandra Castillo, and Juho Rousu. Principal metabolic flux mode analysis. *bioRxiv*, page 163055, 2017.

[40] Zoltan A Tuza, Vipul Singhal, Jongmin Kim, and Richard M Murray. An in silico modeling toolbox for rapid prototyping of circuits in a biomolecular breadboard system. In *Decision and Control (CDC), 2013 IEEE 52nd Annual Conference on*, pages 1404–1410. IEEE, 2013.

[41] Joseph A Wayman, Adithya Sagar, and Jeffrey D Varner. Dynamic modeling of cell-free biochemical networks using effective kinetic models. *Processes*, 3(1):138–160, 2015.

[42] Simon J Moore, James T MacDonald, Sarah Wienecke, Alka Ishwarbhai, Argyro Tsipa, Rochelle Aw, Nicolas Kylilis, David J Bell, David W McClymont, Kirsten Jensen, et al. Rapid acquisition and model-based anal-

ysis of cell-free transcription–translation reactions from nonmodel bacteria. *Proceedings of the National Academy of Sciences*, 115(19):E4340–E4349, 2018.

[43] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[44] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. *arXiv preprint arXiv:1401.4082*, 2014.

[45] Anders Boesen Lindbo Larsen, Søren Kaae Sønderby, Hugo Larochelle, and Ole Winther. Autoencoding beyond pixels using a learned similarity metric. *arXiv preprint arXiv:1512.09300*, 2015.

[46] Rafael Gómez-Bombarelli, Jennifer N Wei, David Duvenaud, José Miguel Hernández-Lobato, Benjamín Sánchez-Lengeling, Dennis Sheberla, Jorge Aguilera-Iparraguirre, Timothy D Hirzel, Ryan P Adams, and Alán Aspuru-Guzik. Automatic chemical design using a data-driven continuous representation of molecules. *ACS Central Science*.

[47] Markus Ringnér. What is principal component analysis? *Nature biotechnology*, 26(3):303, 2008.

[48] Gregory P Way and Casey S Greene. Extracting a biologically relevant latent space from cancer transcriptomes with variational autoencoders. *bioRxiv*, page 174474, 2017.

[49] Paolo Inglese, Zoltan Takats, and Robert Glen. Variational autoencoders for tissue heterogeneity exploration from (almost) no preprocessed mass spectrometry imaging data. *arXiv preprint arXiv:1708.07012*, 2017.

[50] Sarath Chandar, Mitesh M Khapra, Hugo Larochelle, and Balaraman Ravindran. Correlational neural networks. *Neural computation*, 28(2):257–285, 2016.

[51] Qinxue Meng, Daniel Catchpoole, David Skillicom, and Paul J Kennedy. Relational autoencoder for feature extraction. In *Neural Networks (IJCNN), 2017 International Joint Conference on*, pages 364–371. IEEE, 2017.

[52] Anibal A Medina, Contreras Juan P, and Fernan Federici. Preparation of cell-free rnapt7 reactions. `dx.doi.org/10.17504/protocols.io.kz2cx8e`, 2017.

[53] Ashty S Karim, Jacob T Heggestad, Samantha A Crowe, and Michael C Jewett. Controlling cell-free metabolism through physiochemical perturbations. *Metabolic engineering*, 45:86–94, 2018.

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

[55] Ali Ebrahim, Joshua A Lerman, Bernhard O Palsson, and Daniel R Hyduke. Cobrapy: constraints-based reconstruction and analysis for python. *BMC systems biology*, 7(1):74, 2013.

[56] Wout Megchelenbrink, Martijn Huynen, and Elena Marchiori. optgpsampler: an improved tool for uniformly sampling the solution-space of genome-scale metabolic networks. *PloS one*, 9(2):e86587, 2014.

[57] François Chollet et al. Keras. `https://keras.io`, 2015.

[58] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814, 2010.

[59] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

[60] Fabio Chizzolini, Michele Forlin, Noel Yeh Martin, Giuliano Berloffa, Dario Cecchi, and Sheref S Mansy. Cell-free translation is more variable than transcription. *ACS synthetic biology*, 6(4):638–647, 2017.