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 7, 2018

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: 0

Signed:

Date:

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 that reduces a cell-based metabolic model into a CFPS model based on experimental data.

Through the creation of a system to generate these reduced models, we 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, we provide a groundwork for further work at the intersection of these fields. In particular, we 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. We also describe a VAE that we call Corr-VAE that uses a custom loss function to discover better low dimensional representations than a naive VAE. This loss function enables our system to generate metabolic models that are sensitive to different experimental conditions.

We designed this system to create general CFPS models. 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. Although our system was built using data from *Escherichia Coli* (*E. coli*) CFPS systems, our system can be applied to 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 who want to understand what reactions are important in their CFPS systems.

Contents

1 Introduction **1**

2 Background **7**
2.1 Cell-free systems 7
2.2 Flux Balance Analysis 8
2.3 Autoencoders 10

3 Related Work **15**
3.1 Flux Balance Analysis 15
3.2 Reduction of metabolic models 16
3.3 Modeling cell-free systems 17
3.4 Variational autoencoders 18

4 Design and Implementation **21**
4.1 Data gathering 21
4.1.1 Cell-free systems 22
4.1.2 Datasets 24
4.2 Data ingestion and incorporation 26
4.2.1 Data ingestion 26
4.2.2 Data Incorporation 27
4.3 Dataset generation 28
4.3.1 Model generation 28
4.3.2 Flux sampling 29
4.4 VAE 29
4.4.1 Corr-VAE 30
4.5 Flux Balance Analysis (FBA) model reduction 31

5 Results **33**
5.1 Can a VAE distinguish between fluxes generated from different experimental conditions? 34
5.2 Noise experiments 36

5.3 Comparison of reduced models 37

6 Future Work **41**
6.1 Future datasets 41
6.2 Different Organisms 41
6.3 RL system for reduction 43
6.4 Batch variation 43

7 Conclusion **45**

List of Figures

1.1 High-level idea behind our system 2
1.2 The overall pipeline for our system 6

2.1 Example of a single layer autoencoder 11

4.1 Process for creating a CFPS system 22
4.2 Data ingestion part of our pipeline 25

5.1 Comparison of the latent dimensions PCA and VAE 35
5.2 Latent dimensions of our Corr-VAE network 36
5.3 Projection of the latent space of a 10-dimensional Corr-VAE trained on the Manual dataset 37
5.4 Projection of the latent space of a 10-dimensional Corr-VAE trained on the Karim dataset 38
5.5 Reconstructing fluxes reduces the amount of noise 39
5.6 Noise added to input data reduces the efficacy of a Corr-VAE 40

6.1 Our system can generate cell-free models for any organism . . 42
6.2 Future applications for our system 44

List of Tables

4.1 List of reactants for a 50 Microliter (μL) CFPS reaction. Note that MDX is a mixture of substrates (see the complete list in **??**. Amino Acids (AAs) includes all 20 of the amino acids. . . 23

5.1 Comparison of models. Models that have been produced by our pipeline end in 'reduced'. We compare the full GEMs to our reduced models and show that the using our system improves the correlation with real data 40

Acronyms

μL Microliter. 4, 23, 24

E. coli *Escherichia Coli*. 5, 6, 15–18, 22, 41, 42, 45

AA Amino Acid. 23

AE Autoencoder. 18, 19, 45

cAMP Cyclic Adenosine Monophosphate. 23

CFPS Cell-free protein synthesis. 1, 3–8, 10, 17, 18, 22–24, 26, 27, 33–35, 38, 42, 43, 45

COBRA Constraint-Based Reconstruction and Analysis. 3, 27

CSV Comma Separated Value file. 25, 26, 28

DNA Deoxyribonucleic Acid. 4, 22–24

FBA Flux Balance Analysis. 2–4, 6, 8, 10, 15, 17, 18, 27, 28, 31, 42, 43

GEM Genome scale Model. 3, 5, 6, 16, 18, 31, 33, 38, 40, 42

HMP Hexose Monophosphate. 23

K Potassium. 23

KL Kullback-Leiber. 13

LB Lysogeny Broth. 22

LP Linear Programming. 9, 10, 15

ME-Model Metabolic and gene Expression Model. 16

Mg Magnesium. 23

MILP Mixed Integer Linear Programming. 16

NAD Nicotinamide Adenine Dinucleotide. 23

nL Nanoliter. 24

NTP Nucleoside Triphosphate. 23

OD Optical Density. 22

PCA Principal Component Analysis. 5, 11, 17, 18, 33–35, 45

PEG Polyethylene Glycol. 23

PEMA Principal Element Mode Analysis. 17

ReLU Rectified Linear Unit. 30

RFP Red Fluorescent Protein. 24

RNA Ribonucleic Acid. 4

TL Translation. 8, 23

tRNA Transfer RNA. 23

tSNE t-Distributed Stochastic Neighbor Embedding. 35, 37, 38

TX Transcription. 8, 23

VAE Variational Autoencoder. 2, 3, 5, 6, 12, 13, 18, 19, 28–31, 33–37, 40, 43, 45

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. Our efforts focus on modeling the metabolism of a biological system to help biologists better understand what reactions are occurring in that system.

Our goal is to provide a metabolic model of a CFPS system that is useful for biologists working with CFPS systems. This dissertation details an end-to-end system that produces metabolic models for CFPS systems from

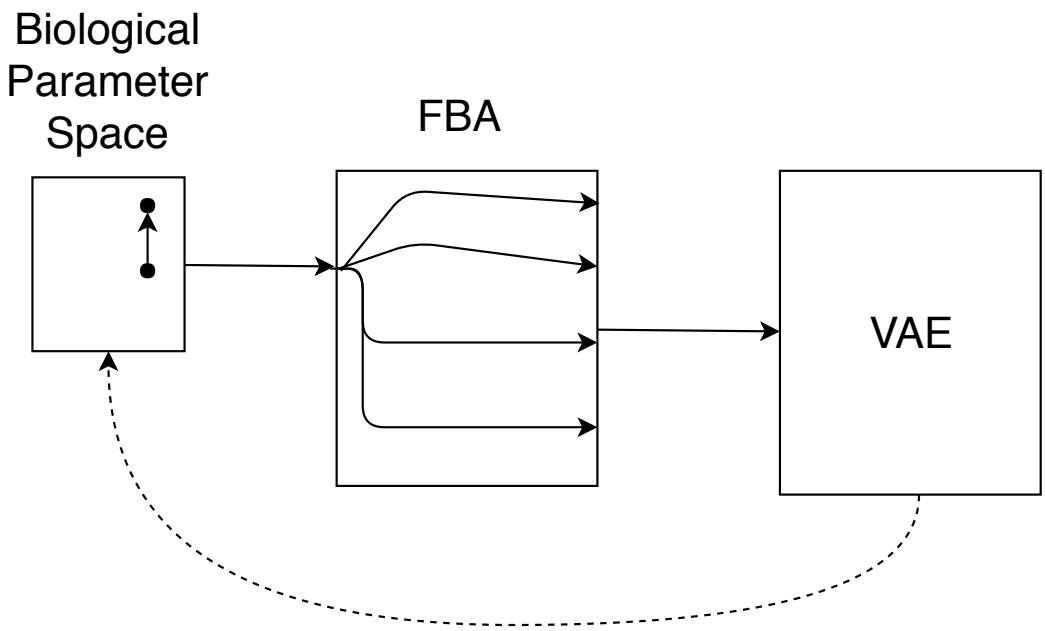

Figure 1.1: High-level idea behind our system. Small changes in the parameter space are amplified across many different fluxes in a FBA model. We used a VAE to understand how the changes in the fluxes reflect the changes in the parameter space.

experimental data. Our system learns which reactions are most important for a given cell-free system and produces a reduced metabolic model that fits the experimental data. We utilize a Variational Autoencoder (VAE) on top of a common metabolic modeling technique called Flux Balance Analysis (FBA) to produce these reduced models. In particular, we introduce a new loss function based on correlation with experimental data for our VAE. We call this new model a Corr-VAE. The Corr-VAE perturbs the latent space of a VAE to better fit our experimental data. Our system 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.

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.

Most metabolic models represent the chemical reactions in a cell 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 our knowledge of biological pathways. 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. We use a VAE to understand how the

changes in the original parameter space are reflected in the output of a FBA model. Figure 1.1 illustrates how our system can be used to

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. We will be also be releasing the CFPS datasets we generated to aid with further modeling work of CFPS systems. Additionally, performing both the experimental work and the modeling allowed us to acquire a deep understanding of the data we were generating. Without this understanding, building biologically relevant models would be difficult.

Cell-free systems

A Cell-free Protein Synthesis (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 we were 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 key issue is that we do not know which reactions are included in that subset.

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. We believe that 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 hand-constructed by a specific lab [13, 14]. We provide 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.

- We provide one of the first applications of deep learning to metabolic models. We show that using a VAE is more effective than PCA with regards to 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 experimental data and have a need for dimensionality reduction.

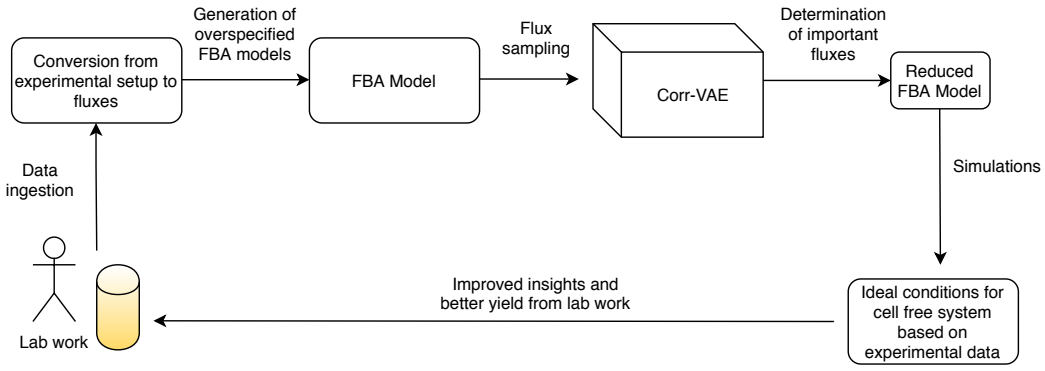

Figure 1.2: The overall pipeline for our system. Given experimental data on one end, we are able to generate cell-free metabolic models.

- Finally, we produce a number of new, reduced models for CFPS systems. These models more accurately describe CFPS metabolism than full-scale GEMs and can be used by biologists and metabolic engineers working with these CFPS systems.

We note that our approach was designed to be as general as possible, which has two key benefits. Our system 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 provides related work in the fields of metabolic models, reduction of metabolic models, modeling cell-free systems, and VAEs. Chapter 4 describes the system we have designed and built to model cell-free systems (shown in Figure 1.2). Chapter 5 describes the results of using a VAE to reduce FBA models. Notably, we show that our reduced models are better at describing CFPS systems than both GEMs and already published reduced models. We conclude with a discussion of future work in this area.

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 we use 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/translation machinery remains intact in the cell extract. These systems are also known as Transcription (TX)/Translation (TL) 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 our work 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. We can represent this by writing the following:

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

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 metabo-

lites ($N >> M$)—so there is no unique solution to this equation. This issue is solved by specifying a specific objective function to optimize for as well as adding constraints to our flux vector v. We can specify constraints on different reactions in order to limit the amount of flux proceeding through a given reaction. For instance, we represent a reaction as irreversible by setting the lower bound on that reaction to be 0. This specifies that it can only proceed in one direction and ensures that the model reflects the underlying biology. We use Linear Programming (LP) to solve this constrained, underspecified system.

In order to find an optimal solution to this problem, we first need to specify an objective function. This objective function determines which fluxes are most phenotypically relevant. When choosing an objective function, we can either choose a specific reaction in which to optimize the flux or create a pseudo-reaction that contains a combination of important metabolites. A common choice of objective function is the Biomass Objective Function, which incorporates many essential metabolites necessary for cell growth [19]. While some of our models use the Biomass Objective Function, we also use the production of a specific product as our objective function. In the general case, we can represent our objective function as Z. We specify which reactions are part of the objective function using the vector c. When the objective function is just the optimization of a single reaction (e.g. production of a desired product), c is extremely sparse, having only a single entry. However, c can also have multiple entries when the objective function is a linear combination of desired products.

Now we can reformulate our problem to be a 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. v_0 is an example of a reversible reaction because flux can go either way. v_1 is irreversible, but unconstrained, while v_2 is irreversible but constrained due to some limiting factor. We can then use a LP solver to find the optimal solution to this problem.

The use of FBA involves this key steady state assumption—i.e. the system is stable. CFPS systems are dynamic systems and do not have a period where intake and output are equal until the system is no longer producing protein. However, we can consider the period of maximal production of a CFPS system to be its steady state. Thus, we can use FBA to analyze this pseudo-steady state, which is the interval that we are interested in.

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 we will just refer to as autoencoders, use two neural networks to accomplish this. One network, which we will refer to as an encoder, performs the mapping from the original dataset to the lower dimensional subspace. A second network, called 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

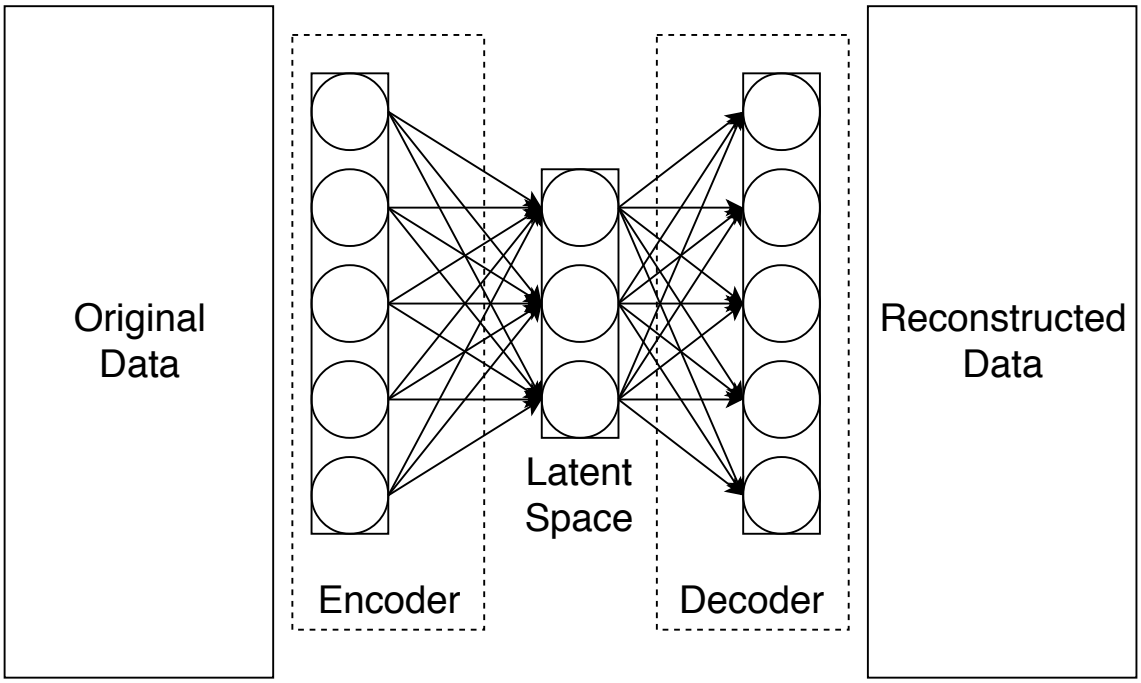

Figure 2.1: Example of a single layer autoencoder. In this case, both the encoder and the decoder are a single layer neural network. These layers can be stacked to create even more complex dimensionality reductions.

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 2.3 demonstrates the structure of a very simple autoencoder. While the main idea of an autoencoder is unchanged, there are a number of different version 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, we can add a sparsity constraint to limit the number of activations among the network layers. This forces the autoencoder to learn a representation with sparse network weights. 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 we want to do more than just learn arbitrary encoding and decoding functions. For instance, we might want to impose constraints on the encoded representation. VAEs are a class of autoencoder that does exactly this. 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. Throughout this work, we will use a normal distribution as our encoded distribution, though other distributions can also be used with VAEs. Thus, the encoder learns to map samples to the mean and standard deviation of the normal distribution. Borrowing terminology from the field of probabilistic models, we call this our latent space. In order to decode a data point, we can sample from the encoded probability distribution to generate a reconstruction in the original dimension.

VAEs measure loss using a combination of two loss terms. Firstly, we care about how well we are able to reconstruct our original data. So, we keep a reconstruction loss term to measure how well the reconstructed data mimics the original data. Our 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 difference between probability distributions. We can compute the the KL divergence between our latent space and a prior distribution, which in most cases is a normal distribution. With the combination of those two loss terms, we can then train the VAE as we would any other neural network—by using gradient descent to minimize the loss function.

Let $x_i \in X$ be our data and z the latent representation learned by the VAE. Our encoder network can be represented by the function p which has parameters θ. Similarly, our 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 our latent representation. Minimizing the second part of the loss requires the latent representation to be similar to our prior distribution. We learn the θ and ϕ terms that minimizes this function.

Chapter 3

Related Work

This chapter introduces related work in the relevant fields of computer science and biological modeling. In particular, we examine work in the field of building metabolic models using flux balance analysis. Next, we present a collection of automatic reduction systems for metabolic models. However, none of these reduction systems have been applied to modeling cell-free systems. We then examine what types of models have been used to model cell-free systems. Finally, we investigate recent work on variational autoencoders.

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 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 will use 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 reduced 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.

Our methods differ from these earlier reduction techniques in a few ways. Everyone so far has constrained themselves to looking at linear techniques of dimensionality reduction. Not all relationships are linear and there are limits to how well linear models can perform. We approach the problem differently by using nonlinear techniques and deep learning to reduce metabolic models. In addition, these techniques all deal with the stoichiometric matrix of the system without having actual experimental data to learn from. Finally, none of the techniques above have specifically targeted 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 re-

actions 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]. Autoencoders (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]. Our Corr-VAE also introduces a correlation loss term, but our correlation is based on external experimental data.

Chapter 4

Design and Implementation

The primary deliverable of this dissertation is a system (shown in Figure 1.2) that is able to generate metabolic models of cell-free systems. Our system has four major parts. First, we ingest experimental data and convert it into a standardized format. Next, we use the data to construct a group of cellular metabolic models and sample fluxes from those models to create a dataset. Then, we perform dimensionality reduction to elucidate underlying trends in the experimental data. Finally, we use 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 reflcets biological reality, we 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. We 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.

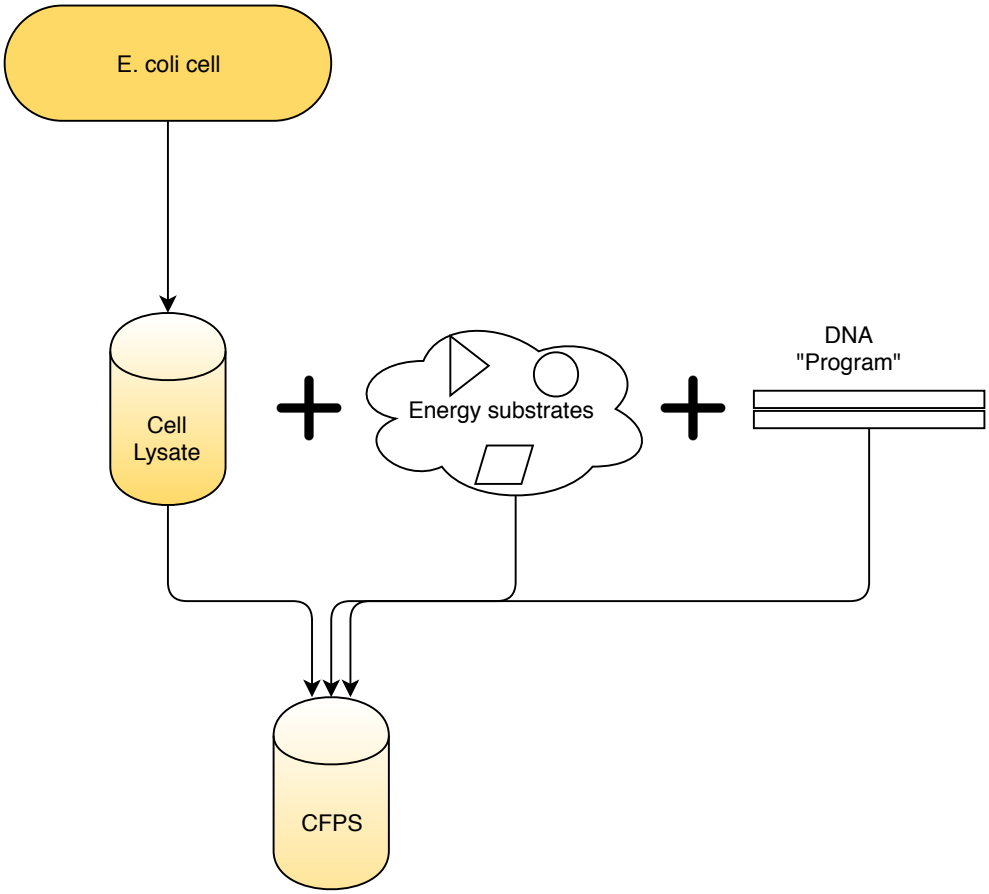

Figure 4.1: Process for creating a CFPS system. The *E. coli* cells can be grown up in bulk, allowing future preparation of CFPS reactions to be far quicker than typical *in vivo* methods.

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. Our protocol is based off of the open protocol out of the Federici lab [52]. We begin by growing up 1L of BL21 *E. coli* cells in an overnight culture of Lysogeny Broth (LB) until they have reached Optical Density (OD) of 1.6. Then, we spin down the cells and remove the supernatant, storing the pellet in the refrigerator overnight if necessary. Next, we perform 3 more sets of spins and washes with S30 buffers

Reactant	**Amount** (μL)	**Purpose**
MDX	5	Energy substrates
AAs	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	50	Protein production

Table 4.1: List of reactants for a 50 μL CFPS reaction. Note that MDX is a mixture of substrates (see the complete list in **??**. AAs includes all 20 of the amino acids.

to remove any extracellular debris. Finally, we use a bead beater to lyse the cells and spin one last time to remove the cellular debris and beads. At the end of this process we have cell extract which can be stored in a -80° freezer.

Once we have this cell extract, we have the core cellular machinery that is necessary for transcription and translation. However, we need to add the cofactors and reactants that are important for the transcription and translation reactions. So, we add energy substrates to the cell extract to create a cell-free protein expression system. These substrates include energy substrates such as Cyclic Adenosine Monophosphate (cAMP), maltodextrin, and Nicotinamide Adenine Dinucleotide (NAD). They also include vital parts of transcription and translation such as Nucleoside Triphosphates (NTPs), AAs, and Transfer RNAs (tRNAs). See Table 4.1 for a complete listing of the reactants and their purpose in the system. Table **??** shows the final concentration for each substrate.

Given this cell-free expression system, we have essentially assembled a biological computer. Whatever DNA "program" is added, the system will work to produce the appropriate protein result. This platform is incredibly powerful because it allows us 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 coerced to uptake the DNA and

incorporate it into their production process.

4.1.2 Datasets

We used the protocol above to create two datasets for training. The first dataset was created by hand and followed standard lab procedures. We pipetted each of the reactants in the ratios specified by Table 4.1 to create a 200 μL mastermix. From that mastermix, we then used 5 μL aliquots combined with each of our different reaction conditions. In order to test different reaction conditions, we added an additional 1 μL of various reactants. For this dataset, we experimented by adding 16 differing ratios of sugar, phosphate, potassium, and nucleotides. The DNA circuit in the CFPS system simply produces Red Fluorescent Protein (RFP), so we were able to judge the amount of protein production by measuring fluorescence readout with a plate reader.

However, there are multiple downsides to doing this by hand, so our second dataset was created using a Labcyte Echo acoustic liquid handler. Creating large datasets by hand takes a long time and limits the amount of data one can generate. Additionally, pipetting by hand has an intrinsic error, so we hoped that automating the creation of the reactions would reduce human error. The Echo is an acoustic liquid handler that can combine different amounts of liquid to create each of the CFPS reactions. We were able to perform the experiments at $2\mu L$ of CFPS system and 500 Nanoliter (nL) of additional reactants. The use of automation allowed us to test 48 different reaction conditions and shows that this could be expanded to an even larger scale.

Finally, we received a third dataset from the recent Karim and Jewett paper that uses CFPS systems to perform metabolic engineering [53]. Our protocol is very similar to the one used in the Jewett lab, though some of the reactant concentrations differ. This dataset is primarily useful because it was created in a different lab and they investigated a metabolic pathway whose output is not fluorescence. Since it was created in different lab conditions, we can check

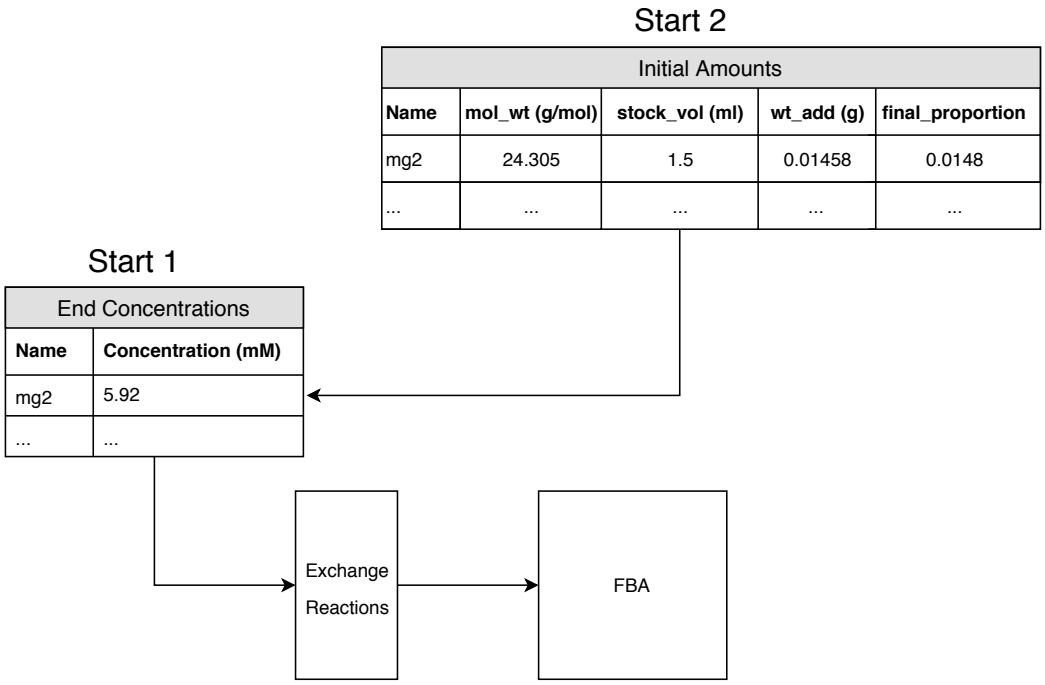

Figure 4.2: Two possible ways to incorporate the experimental setup in our pipeline. In Start 1, the user already knows the final concentrations of each reactant and can upload that as a CSV. Otherwise, the user can upload information about the reactants that are added and our tools will automatically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

that our model is learning something about cell-free systems in general, not just overfitting our data. The pathway that they investigated is an inherently metabolic pathway, so we can also see if our metabolic model is better able to describe these types of gene circuits.

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, we wanted to make it end-to-end and as easy to use as possible. With that in mind, we created tools to automatically incorporate the experimental setup into the models. Figure 4.2 shows the two main ways that the experimental setup could show up. The first is in the form of end concentrations of the different reactants in the cell-free system. This is the easiest because in that case we can simply convert concentrations into fluxes. Fluxes have the unit mol/min/gDW, so we can convert metabolite concentrations into flues 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. We set t to be 24, the time scale of our reactions. Using this equation our tool is able to ingest a CSV mapping the name of the reactant to the final concentration and turn them into flux constraints.

However, we also support users who might only know the relative amounts of each reactant that they are adding. 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 concentrations in their mixture. To support these users, we allow them to upload a CSV (or combination of CSVs) that contains information about each reactant used to make the system. For each reactant, we require name of each reactant, the molecular weight, the volume used to create the stock, the weight used to create the stock, and the final volume added to the cell-free reaction. We use this information to calculate the final concentration of each reactant in the CFPS system. Once we have manipulated the data into the same form as the first type of upload, we can proceed with the rest of our computational pipeline.

4.2.2 Data Incorporation

After we ingest the experimental setup, we need to incorporate it into our FBA model. We use the COBRApy python package [55] to handle our FBA models. The base model that we start with is the iJO1366 model for *E. coli* which consists of 1805 metabolites and 2583 reactions [31]. This model needs to be adapted to cell-free systems because they no longer have the same physical structure as *E. coli* cells. First, we need to make the cell-free reaction reflect the fact that there are no longer any membranes or cellular compartments. We do this by iterating over every reaction that contains a periplasmic or extracellular component. For each reaction, we then replace every metabolite with either the corresponding cytosolic metabolite or, if no corresponding metabolite exists, we just change the metabolite to be in the cytosol. At this point, we have all of the reactions occurring in the same location, which reflects how a CFPS system works.

Our next task is to handle how exchange reactions are dealt with in the model. Exchange reactions are pseudo-reactions that allow us to constraint the amount of a reactant that is allowed to be in the model. For each of the reactants in our CFPS system, we replace the bounds on the exchange reactions based on the fluxes we calculated from the setup data earlier. At the end of this process, we have converted the full-scale *E. coli* model to better approximate a CFPS system. While still overspecified, we have now created a new base cell-free model.

We also extend this model to explicitly incorporate the transcription and translation reactions that are so crucial to CFPS systems. Most FBA models do not explicitly consider 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. We built auxiliary tools such that, given the sequence of the gene of interest, we can automatically generate a version of the Varner model. Next, we convert that model to Python and extract the relevant transcription and translation re-

actions. We incorporate those reactions into our model and then can set the production of the protein of interest as our FBA objective. This model is called our TXTL cell-free model.

4.3 Dataset generation

4.3.1 Model generation

Once we have created these base models, we can use them in conjunction with the different experimental conditions to create a different model for each condition. To do this, we take in a CSV consisting of the experimental conditions and a quantitative output. Each row represents a different experiment. The columns contain either the different concentrations for starting reactants or the different amounts added. In the latter case, we can re-calculate the final concentrations of each of the reactants for each experimental starting condition. In both cases, once we have the new final concentrations for each reactant, we create a new FBA model.

Were we simply to solve the different models we have created, many would have the same flux results. For these naive models, Section 5.3 demonstrates that they do not describe our biological results very well. Clearly, FBA alone cannot describe the full richness of a real system. We also have the problem that the model is overspecified. These FBA models contain every reaction that is occurring in a steady state bacterium. While we do harvest the bacteria at steady state, some enzymes will degrade and others will not be important in cell-free systems. These types of changes cannot be encapsulated solely through a change of objective function and coalescence to a single-compartment reaction.

One can imagine FBA as a coarse bijective function, so points that vary only slightly in the input space will get projected into the same output point. We want to do better by first passing the fluxes through a dimensionality reduction technique such as a VAE. However, in order to do this, we first

need a large dataset of fluxes to train on. Since we only have a small number (dozens, not hundreds) of models, we cannot simply use the fluxes from the optimal solution to the model. We supplement our dataset by flux sampling from each model.

4.3.2 Flux sampling

Flux sampling is the process of sampling possible fluxes through each reaction. This gives us a distribution of fluxes for each reaction, for each model. We use optGpSampler to sample 2000 fluxes from each of our models [56].

Each model has different experimental conditions and therefore has slightly different flux distributions for each reaction. So, after we generate flux samples for each model, we can combine the sampled fluxes to generate two new datasets. One dataset is flat—meaning we simply concatenate all of the sampled fluxes into a dataset of size $(2000E) \times R$ where E is the number of experiments and R is the number of reactions. This dataset is useful to investigate the fluxes and see if we can perform dimensionality reduction to gain insights on the fluxes in our models in general. The other dataset we can make involves sampling with replacement from the fluxes of each model and then stacking the fluxes to create a new dataset. This dataset has size $N \times E \times R$ where N is the number of samples with replacement we drew. Using this dataset allows us to incorporate the correlation term in our VAE.

4.4 VAE

The key part of our system was the usage of a Variational Autoencoder in order to reduce the dimensionality of our model. Our VAE was implemented using the Keras framework [57] and the implementation was designed to be as modular as possible. We wanted to make the VAE as modular as possible so that end-users would be able to easily change the parameters of the VAE without having to have any prior training in deep learning. Thus,

our implementation is able to take in 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.

The structure of the VAEs we used are as follows. We 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, 512, and 256. We also tried using latent dimensions of size 2 and 10. For activations between layers of the encoder and the decoder, we used Rectified Linear Unit (ReLU) activations [58]. The activation of our final layer of the decoder is one of sigmoidal, tanh, or linear, depending on how the dataset was scaled. When using the custom correlation loss function, we required a stacked dataset, otherwise a flat dataset could be used.

4.4.1 Corr-VAE

The key insight for the correlation loss is that we did not just want to reduce our dimensionality in an unbiased way—we care about how well it describes real-world data. Thus, we can use our additional information to perturb the latent space to better reflect biological reality. We know what the relative rankings of the different models should be, so we can add a correlation loss term between the experimental data and the latent representation. Recall Equation 2.3 that described the VAE loss function. We now train with a modified loss function:

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

where x_i' is the reconstructed data and d is our scaled biological data. We call a VAE with this loss function a Corr-VAE.

4.5 FBA model reduction

Now that we have run our models through our correlational VAE we have a lower dimensional representation of the model. We can use this lower dimensional representation to create a better cell-free model. One way we could use our system to do this is by running the whole system on each set of new data. We first generate the FBA model using the tool chain described above. Then, using the trained autoencoder, we can transform the optimal fluxes and get the reconstructed fluxes that better describe a cell-free system. This could be very useful for batch variation or comparisons between different labs.

However, one of our goals is to build a new cell-free model. So we can use the dimensionality reduction from the VAE to create a more accurate reduced model. We do this by thresholding the latent space and removing reactions that have fluxes close to 0. This simple thresholding allows us to identify reactions that are not important in cell-free systems because they do not vary between different runs of our cell-free reactions. Section 5.3 shows that these reduced models with the differential conditions give us better explanations of experimental data than full GEMs.

Chapter 5

Results

Our system is able to produce biologically relevant dimensionality reductions that allow us to make better reduce models for CFPS systems. We show the following key points:

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

- The separation between the different models improves as we add more latent dimensions. This is replicated across different CFPS systems.

- As noise is added to our model, Corr-VAE performance falls off, showing that we are learning real trends in the data.

- We can use the latent representations of our Corr-VAE to reduce GEMs and create CFPS models that correlate better with experimental data.

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

We want to use our VAE to learn how changes in the experimental conditions affect the underlying CFPS system. If the VAE is going to be useful for us to extract information about the underlying cell-free system, it will need to be able to distinguish between different starting experimental conditions. In order to assess how well our VAE is learning the differences between experimental conditions, we can examine the latent space of the VAE. If our VAE is performing well, we should see a clear separation between fluxes generated from different models. For this set of experiments, we 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 project the same test dataset into 2 dimensions for visualization purposes. The VAE and the Corr-VAE have identical networks of 3 layers of size 1024 for their encoder and decoder networks.

Figure 5.1 visualizes 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, when we use our custom loss function, we are able to recover clear distinctions between the different 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 demonstrate that Corr-VAE may be a more effective way to understand data.

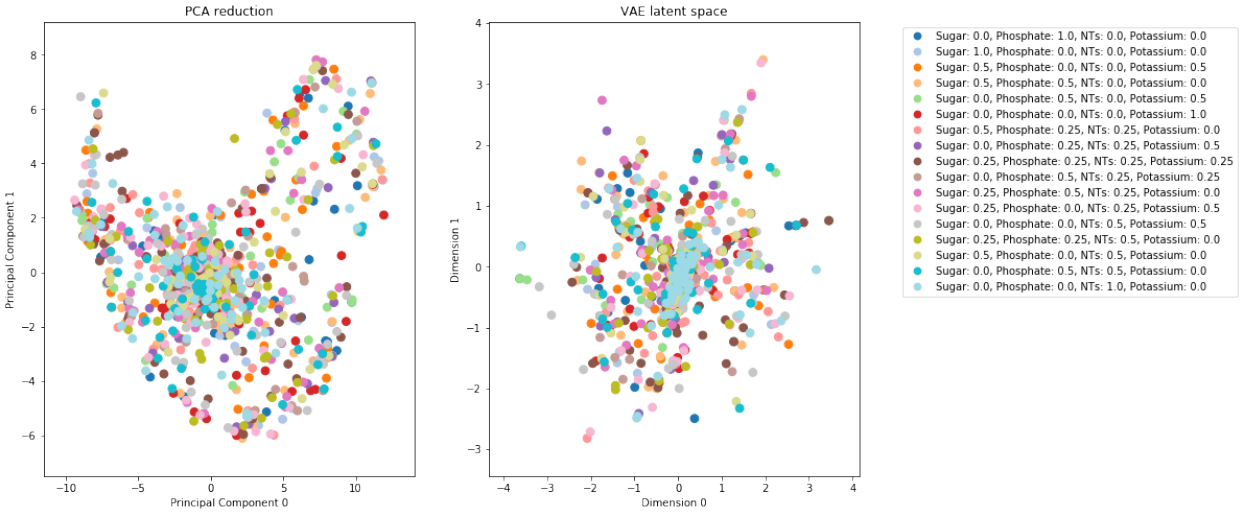

Figure 5.1: Left: first two principal components of a PCA on our test dataset. Right: the two latent dimensions of a trained VAE with the classic loss function. Different colors represent different experimental conditions. Neither PCA nor a basic VAE can learn a clear representation of the different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

Our original dataset had on the order of 2600 features, so using a latent space of only 2 dimensions is a very severe dimensionality reduction. We also trained a Corr-VAE with a latent space that had 10 dimensions to see if more dimensions allowed for better reconstruction. In order to visualize the latent space of this Corr-VAE, we projected our 10-dimensional latent space 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 able to clearly separate different starting experimental conditions. These results suggest that more biological applications should consider using a VAE instead of PCA for dimensionality reduction purposes. We also show that this technique can generalize across different CFPS systems. Figure 5.4 is generated using the same structure for our Corr-VAE, but on the Karim dataset.

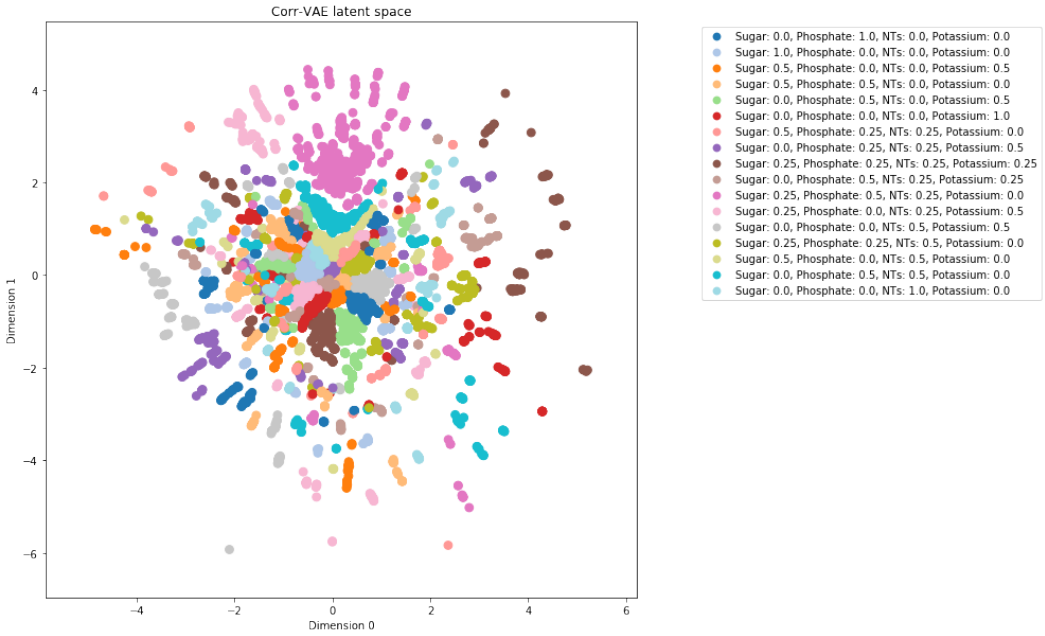

Figure 5.2: Different colors represent different experimental conditons. This plot shows the two latent dimensions of a trained Corr-VAE. Colors cluster together and show that a Corr-VAE can recover the original models.

5.2 Noise experiments

We also validated that our VAE was performing as intended. We can examine a few aspects of our reconstructed fluxes to ensure that the results we are seeing are not simply due to chance. Figure 5.5 plots the standard deviations of each of the reactions for our test dataset as well as our reconstructed dataset. We can see that the reconstructed dataset maintains a very similar shape to our original dataset, while also reducing the standard deviation of fluxes with each reaction. Although we did not investigate this further, this implies that the VAE could also be used to de-noise our dataset.

Another key point we investigated was whether or not the correlations we produce are spurious. We wanted to ensure that our VAE is not creating correlations out of thin air just to reduce its loss function. Figure 5.6 shows what happens when we add noise to our experimental data. First, we generate a dataset with fluxes that are highly correlated to our experimental

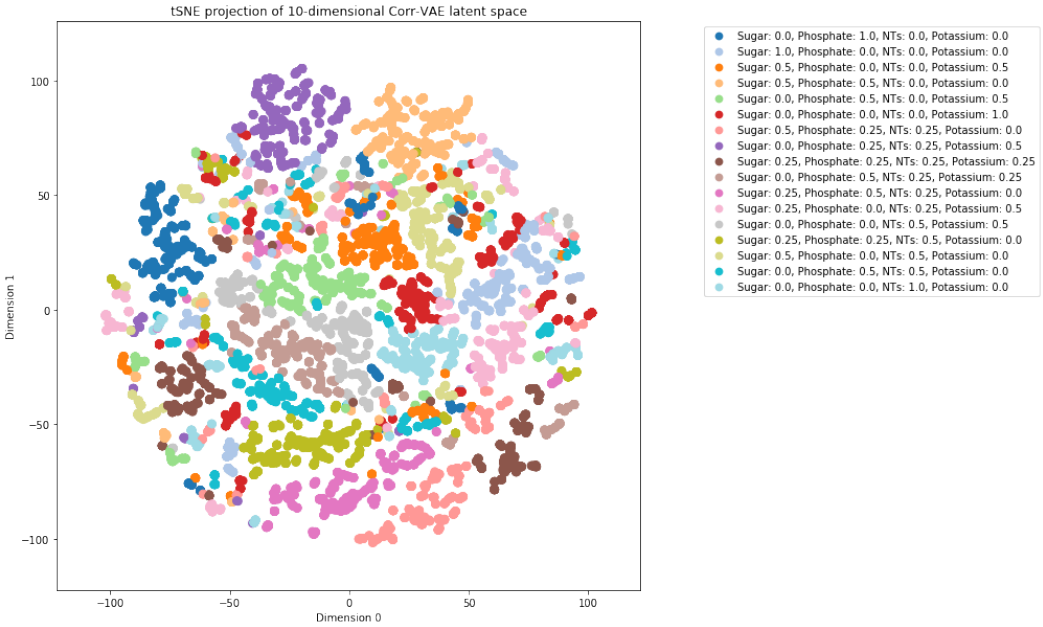

Figure 5.3: Different colors represent different experimental conditons. This plot shows a tSNE projection of the 10-dimensional latent space of a Corr-VAE trained on the Manual dataset. Adding more latent dimensions allows better reconstruction of the invidual models.

data. This is represented by the green line. We can then run those fluxes through our VAE and check the correlation. The blue line shows that this improves the correlation of our fluxes because of our loss function. Next, we can add noise to the data by randomly drawing from a normal distribution and adding it to the flux for each reaction. The dots show that as we add increasing amounts of noise to the data, the correlation goes to about 0. This means that the VAE is not simply artificially creating a correlation but is learning something about the underlying data.

5.3 Comparison of reduced models

A key goal for our system was to produce reduced models that could accurately explain our biological data. We tested this by solving for the optimal flux distribution for each model of our reaction conditions. We then found

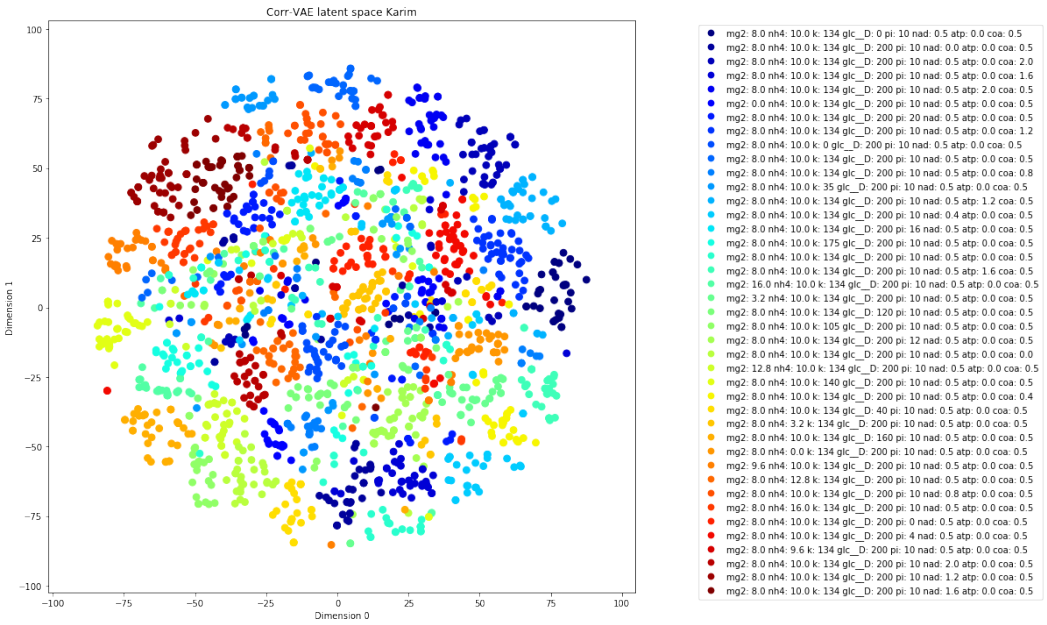

Figure 5.4: Different colors represent different experimental conditons. This plot shows a tSNE projection of the 10-dimensional latent space of a Corr-VAE trained on the Karim dataset. Our Corr-VAE is able to reconstruct the individual models for multiple types of CFPS systems.

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 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

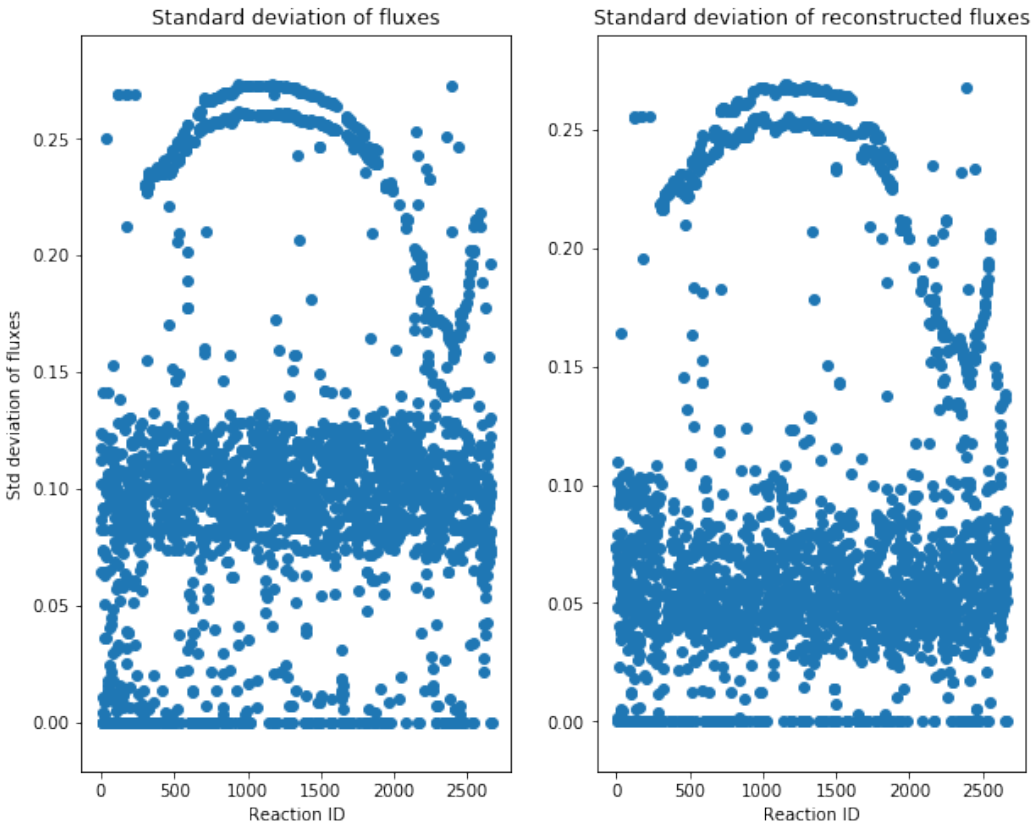

Figure 5.5: Reconstructed fluxes reduce the amount of noise while also maintaining the shape

describe the biological system, the reduction is far better than a full-scale model.

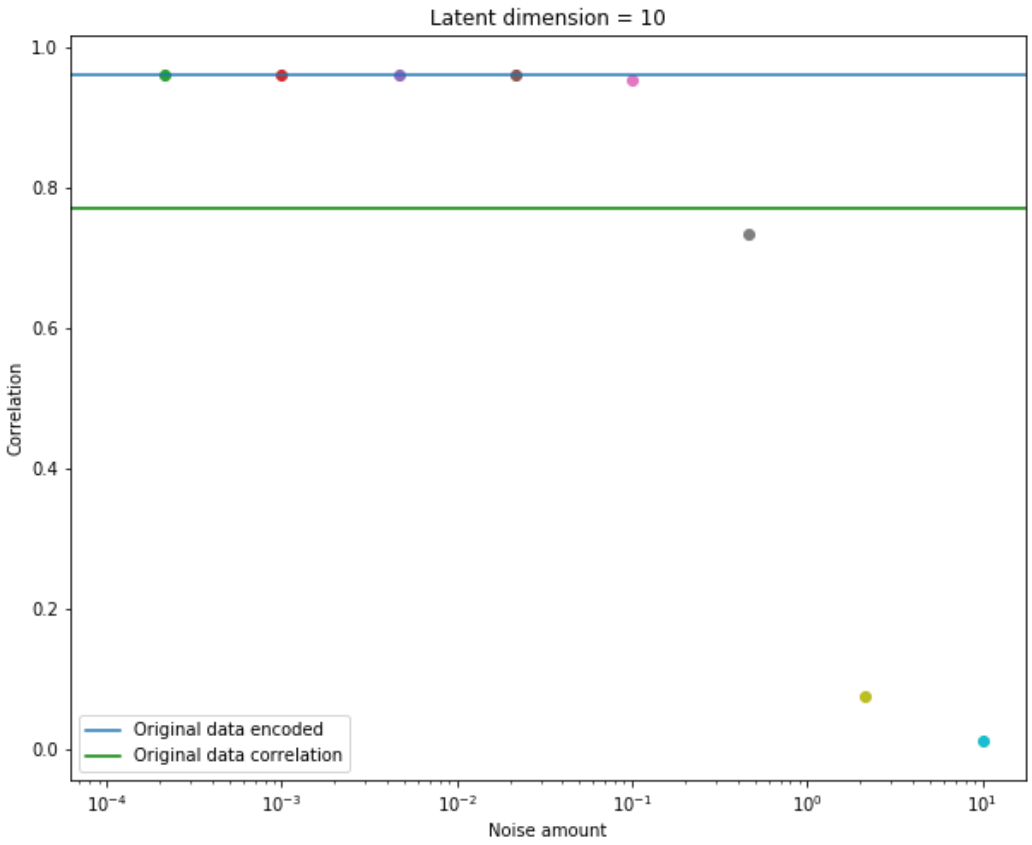

Figure 5.6: Adding noise to the data reduces the correlation from our VAE

Dataset	TXTL	Reduced	Correlation
echo	Yes	No	0.22
echo	No	No	0.14
manual	Yes	No	0.23
manual	No	No	0.24
karim	No	No	0.17
coli-pruned	No	Yes	0.34
manual-reduced	Yes	Yes	0.50
karim-reduced	Yes	Yes	0.43

Table 5.1: Comparison of models. Models that have been produced by our pipeline end in 'reduced'. We compare the full GEMs to our reduced models and show that the using our system improves the correlation with real data

Chapter 6

Future Work

6.1 Future datasets

The datasets that we used in this dissertation are relatively small—on the order of 10s of different parameter combinations. The limited scope of these datasets means that we are only able to coarsely cover the entire parameter space. We hypothesize that we would get better reduced models if we were able to train on larger models. Generating large biological datasets by hand is difficult and error prone. However, we demonstrated the potential for further lab automation by generating a dataset using the Labcyte Echo liquid handler. We encourage future work to 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

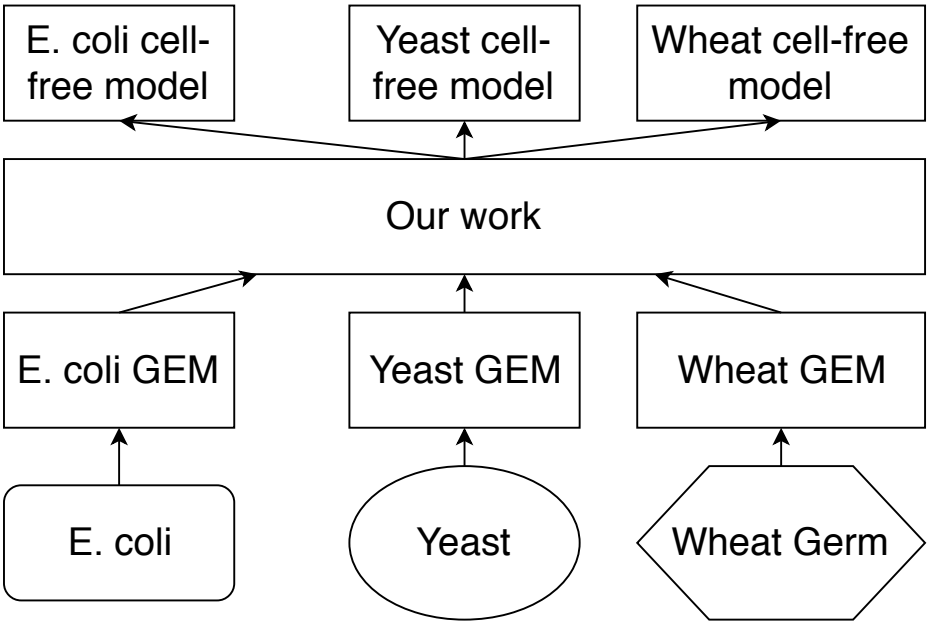

Figure 6.1: Our system was designed to convert any organism's GEMs into an appropriate cell-free model.

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

One issue is that this pipeline is only as good as the initial models. The initial FBA model is an imperfect description of a full *E. coli* cell, so any errors in that overall model will be replicated in our reduced models. FBA models are periodically updated with new data. Our system is built to be flexible and can easily produce new models based on any updates.

6.3 RL system for reduction

Our 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. We removed reactions based on a simple thresholding criterion. Instead of deciding which reactions to remove using a threshold, we 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, more sophisticated search methods could likely produce better reduced models. To that end, we also provide a framework built on top of OpenAI's Gym environment for performing reinforcement learning on FBA models. Given a starting model with almost 2600 reactions, the space of possible reduced models is enormous. Thus, we believe a combination of our Corr-VAE and reinforcement learning could yield a better reduced model.

6.4 Batch variation

Finally, we envision an important use for our system as a way of dealing 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 shows how our system 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 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

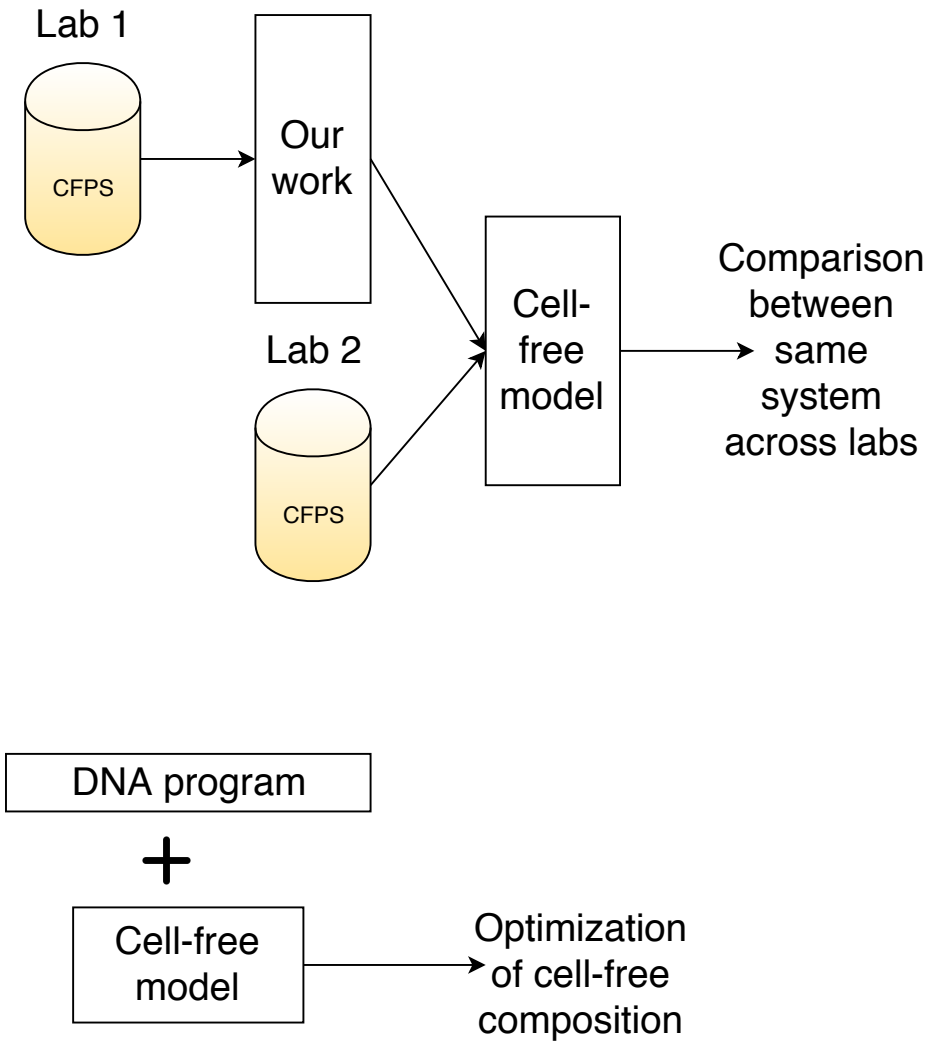

Figure 6.2: Two future applications of our system. Above, multiple experiments can be projected in the same subspace to better compare experimental results. Below, the system can help optimize the energetic composition of a cell-free system

dimensionality of the data. This transformed data should allow for better comparisons across batches.

Chapter 7

Conclusion

In conclusion, we have shown that deep learning techniques can be successfully applied to biology. In particular, computational biologists should start using AEs (especially VAEs) for dimensionality reduction instead of defaulting to using PCA. Among our contributions are the new loss function that we developed for our VAE, which is general enough to be used with any biological data and the set of new *E. coli* CFPS metabolic models. We hope this work encourages other computational biologists to continue work integrating deep learning with biology in general and metabolic models specifically.

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.