Building a Cell-free Metabolic Model using Variational Autoencoders

Nicholas L. Larus-Stone
Queens' College

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
Computer Laboratory
William Gates Building
15 JJ Thomson Avenue
Cambridge CB3 0FD
United Kingdom

Email: nl363@cl.cam.ac.uk

May 30, 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

This is the abstract. Write a summary of the whole thing. Make sure it fits in one page.

Programmable matter (the synthetic biology aspect)

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Despite more than a century of active biology research, running biological experiments is still time consuming and difficult. Therefore, having a good computational model of the biological system of interest is incredibly valuable. What makes a computational model good? Most importantly, a good model needs to accurately describe how the biological system acts in the real world. A model is only useful if it accurately represents the biological system. This allows experiments and hypothesizes to be tested using a model instead of in vivo, speeding up the pace of research. A good model should also be consistent–meaning it should facilitate running the same experiment multiple times and getting the sam results. Though biological experiments can have wide variance due to external conditions, we want our model to vary only due to intrinsic biological sources of variation. Exact reproducibility is very difficult in laboratory environments, but a computer model can and should eliminate the extrinsic sources of variance that one encounters in a laboratory setting. Finally, the best models not only describe the system, but also facilitate deeper insights into the underlying biology. A good model should help researchers improve their understanding of the system that is modeled and provide a starting point for novel insights.

Biological organisms are incredibly complex systems that can be examined at many different levels. For example, one model could use a physical model of the system to describe the shape of an cell as it grows. A different model could use multi-omic data–genomic, transcriptomic, and metabolomic measurements–to model the production of compounds that allow for cellular growth. As all models are inherently limited in some way, the choice of what type of model that is used to describe a system is extremely important.

This dissertation will focus on using metabolic information to build a good computational model for a specific type of biological system called a cell-free system. In particular, we use a Variational Autoencoder (VAE) with a custom loss function on top of a metabolic model called Flux Balance Analysis (FBA). We combine these modeling techniques to build a pipeline that learns which reactions are most important for cell-free systems. The goal is to provide a model for biologists to better understand what is occurring in cell-free systems. This will allow them to anticipate how different starting conditions will affect their experiments before actually running the experiment.

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. Many disciplines in biology involve examining the metabolism of an entity as it is 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 the metabolism, some work has also attempted to engineer the metabolism. Metabolic engineering has the potential to create novel advances in medicine and energy production [1]. Metabolic models have gained popularity over the past few decades as an important tool to aid in the manipulation of cellular metabolism. These models are used to predict how the phenotype of the organism will respond to various environmental or genetic inputs.

Most metabolic models represent the chemical reactions in a cell through a mathematical representations of the reaction coefficients. Individual reaction representations can be connected together and constrained 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 among the metabolic engineering community. In particular, Flux Balance Analysis (FBA), is one of the most popular metabolic modeling techniques that has been used in hundreds of different works on metabolism [3] These applications range from optimizing metabolic pathways [4] to investigating gene networks [5]. 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, thus FBA cannot directly be applied to a cell-free system.

Cell-free Systems Cell-free protein synthesis (CFPS) systems are a reduced version of cells that allows these systems to produce proteins without being alive. The Central Dogma of biology, as enunciated by Francis Crick, is that DNA makes RNA makes protein [6]. While this may be a slight oversimplification, the point remains that the production of protein is the end goal for biological systems. CFPS systems take that to an extreme by removing all endogenous DNA and RNA and membranes while retaining the machinery crucial 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 biologist determines the output of the CFPS system by customizing the DNA that he or she adds to a cell-free system. The combination of this simplicity with the inexpensive nature of cell-free systems have made them increasingly popular in an array of applications [].

CFPS systems have two key aspects that make it 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 executing an experiment does not require growing up cells. This meant that we were able to start from scratch and generate relevant data within a few months instead of the many months or years that a cell-based system could take. CFPS systems should also be simpler than an full cellular system and should therefore be easier to model accurately. By definition, CFPS systems contain a strict subset of the reactions that occur in a cell-based system. However, one issue is that we do not know which reactions are included in that subset.

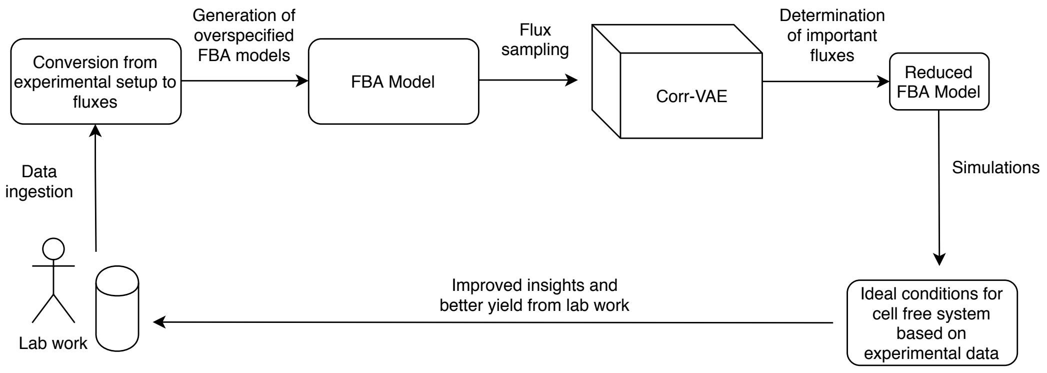
Despite their reduced nature, CFPS systems have not been fully modeled very often. Among these attempts, almost none of them have used metabolic models to examine CFPS systems. We believe that metabolic models are a natural fit to model CFPS systems. Since CFPS are not living organisms, their effectiveness is entirely based on reactions constraints and energy regeneration. Thus, it makes sense to examine these systems from the perspective of metabolism. The few existing metabolic models for CFPS systems are hand-constructed by a specific lab. We use the standard GEMs that already exist for the original organism, allowing us to create a more scalable and generalizable way to generate these models of CFPS systems.

Contributions This dissertation makes a number of contribution in the field of metabolic modeling for CFPS systems.

- This dissertation provides an end-to-end system to generate models for CFPS systems. Given biological data generated in a lab, the computational pipeline will then produce a metabolic model for the given CFPS system.
- Within that framework, we provide a the first automated reduction system that tailors GEMs specifically for CFPS systems. That means our system can be theoretically used for any cell-free system which is derived from an organism that has a fully specified GEM, of which there are many dozens.
- Additionally, it is the first use to our knowledge of applying deep learning or autoencoders to FBA models. As deep learning becomes ever more popular, our system provide an early example of how it may be incorporated into the field of metabolic modeling.

Paper outline The structure of this dissertation is as follows. Chapter 2 provides background information on cell-free systems, FBA, and VAEs. We describe the different types of cell-free systems, how we produce our system, and some uses and advantages for cell-free systems.

Chapter 3 provides related work in the fields of modeling cell-free systems, FBA, applying machine learning to FBA, and VAEs. We focus on describing



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 1.1: A figure showing the overall pipeline for the system to generate cell-free metabolic models.

the current state of the art in metabolic modeling, specifically focusing on FBA and the creation of a GEM for E. coli. Next, we focus on efforts to model cell-free systems, describing the few examples of FBA applied to cell-free systems. Then, we describe the relevant literature in autoencoders, in particular VAEs. Finally, we describe previous applications of machine learning to FBA.

Chapter 4 describes the system we have designed and built to model cell-free systems. The entire pipeline is shown in fig 1. We begin by describing the biological experiments and the type of data that was generated. Next, we describe how to ingest the experimental setup and convert it into various FBA models. Then, we explain how we flux sampled from the FBA models to generate a larger dataset for our deep learning algorithms. That dataset is then ran through a VAE, which generates a latent representation. That latent representation is used to reduce the FBA models. Finally, we describe how we the reduced FBA model to improve future experimental yield.

Chapter 5 describes the results of the experiments. This includes results showing the improvment of my technique over naive solutions as well as novel biological insights that were unearthed due to my system.

Finally, we conclude with a chapter of discussion and directions 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 [7]. These CFPS systems were quite crude and had low protein yields, reducing their utility. More recently, work involving the removal of certain genes allowed for stabilized amino acid synthesis leading to improved yield of CFPS systems [8]. Another crucial development was the ability to ensure cofactor regeneration pathways were maintained in the CFPS system [9]. These recent improvements in CFPS systems have led to a multitude of applications for cell-free technologies [].

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 [10]. 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 aded, 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 other type of CFPS system is an extract-based system created using a crude cell extract. This is the type of CFPS system we will be using throughout this dissertation. Creating an extract-based CFPS system involves growing up cells and then blending them up into extract. This extract is then used as the basis of the CFPS system with the assumption that all the important transcription/translation machinery will be in the cell extract. Growing up 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 way of creating CFPS systems is that the exact composition of these cell extracts is unknown and could potentially vary between batches of CFPS systems. This motivates our work to provide a system that can accurately model these crude CFPS systems.

2.2 FBA

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 can be represented using a matrix S, where each row represents a separate reaction and each column is a metabolite. Entries in this matrix consist of the stoichiometric coefficient for each metabolite in the reaction, where metabolites that are consumed 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. FBA then makes the assumption 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:

$$S * v = 0 \tag{2.1}$$

where S is the known stoichiometric matrix and v is the unknown flux vector.

This system is underspecified–we typically have more reactions than metabolites–and therefore there is no unique solution to this equation. We deal with this issue by specifying reaction constraints and choosing an objective function. We can specify constraints on different reactions in order to limit the amount of flux proceeding through a given reaction. For instance, if we want to represent a certain reaction as irreversible, we could set the lower bound on that reaction to be 0, specifying that it can only proceed in one direction. When choosing an objective function, we can either choose a specific reaction to optimize the flux through or create an artificial reaction that contains metabolites we care about and optimize for that. A common choice to optimize for is the Biomass Objective Function, which incorporates many essential metabolites necessary for cell growth [11]. Our choice of objective functions will be discussed further in Section 4.2.2

With the constraints and objective function in mind, we can reformulate our problem to be linear programming problem of the form:

$$maxZ = cvs.t.S * v = 0and - 1000 < v_0 < 1000and0 < v_1 < 1000... \quad (2.2)$$

We can then use a linear programming solver to find a feasible solution to this problem. It may be useful to imagine the potential space of all FBA solutions that can satisfy the constraints as a hypercube. Then, the linear programming solver finds an optimum at one corner of the hypercube based on the objective function.

The use of FBA involves this key steady state assumption–i.e. the system is stable. CFPS systems don't have the same type of steady state as a typical cell where intake and output are equal. However, we can consider the steady state of a CFPS system to be the period of maximal production. Thus, we can use FBA to analyze this pseudo-steady state and optimize that time of production, which will lead to greater protein production overall.

A typical FBA model includes thousands of reactions–this is extremely high dimensional data. In order to extract insights from the data, we need to use dimensionality reduction techniques.

2.3 VAEs

Autoencoders are a dimensionality reduction technique with a very simple idea: given a dataset, try to reconstruct that dataset by passing it through a lower dimensional subspace. The original idea behind autoencoders was that this lower dimensional representation of the data functioned as a compressed version of the data. Instead of hand designing compression/decompression algorithms, these functions can be learned to be application specific. These autoencoders are often worse than hand-designed compression algorithms and have a lot of difficulty competing with classic algorithms on problems such as image compression [12]. However, the lower dimensional compressed representation can be used as a dimensionality reduction technique to uncover non-obvious relationships within the original data. This is similar to the idea behind Principal Component Analysis (PCA), though PCA projects the data into a subspace in a linear manner, while autoencoders are able to learn non-linear transformation.

The implementation of autoencoders usually uses neural network layers to learn the encoding and decoding functions. Figure 2.3 demonstrates an example structure of a very simple autoencoder. There are a number of different types 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 amount 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 back to a clean representation of the data. This can either be trained on noisy data or it can be given the original representation, corrupt it, and then try to learn the original representation from the corrupted 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, also known as the latent space. Variational au-

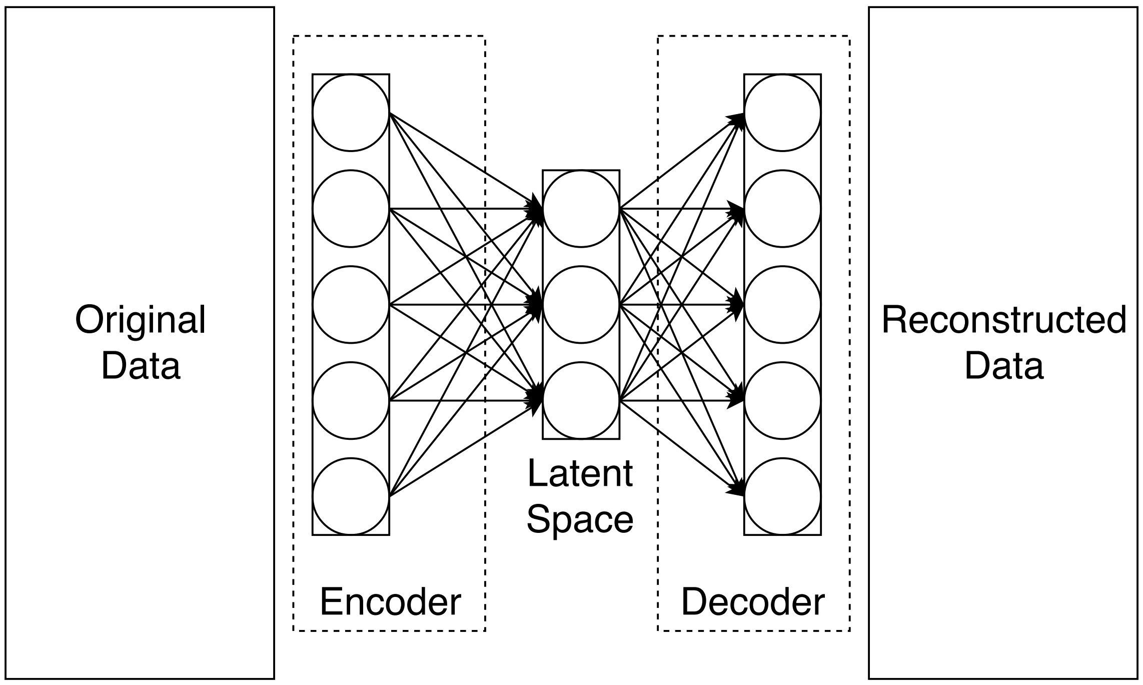


Figure 2.1: Example of a single layer autoencoder. In this case, both the encoder and the decoder are a single layer neural network. These layers can be stacked to create even more complex dimensionality reductions.

toencoders (VAEs) are a class of autoencoder that does exactly that. VAEs constrain the encoded representation to be a probability distribution. So, instead of learning functions that encode/decode the data, the VAE actually learns the parameters of the encoded probability distribution. Specifically, the encoder learns to map samples to a mean and standard deviation, which is our latent space. Then, when decoding the sample, we can sample randomly using the encoded mean and standard deviation as parameters for our probability distribution.

We measure loss in this scheme through a combination of two loss terms. First, we still care about how well we are able to reconstruct our original data. So, we can have a reconstruction loss term which measures how well the reconstructed data maps to the original data. Our second loss term constrains the latent space to ensure that it is well formed and provides a regularization term against overfitting. We use KL divergence, a common way of measuring difference between probability distributions. We can calculate the the KL divergence between our latent space and our 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.

Chapter 3

Related Work

This chapter introduces related work in the relevant fields of computer science and biological modeling. The key insight is that very few techniques have focused on modeling cell-free systems and no one has yet applied deep learning techniques to metabolic models.

3.1 FBA and GEMs

The earliest work on metabolic modeling comes from the 1980s with the use of stoichiometric equations to predict the yield of specific fermentation products [13]. This work was soon expanded to other systems and the use of linear programming (LP) to solve for the optimal result was proposed [14]. Constraint based analysis was soon applied to the description of the metabolic system in E. coli [15]. Improvements on this initial model have been stepwise in the years following. For instance, a large step involved the addition of catabolic pathways to the model [16].

With the growth in genome sequencing, however, came the ability to describe metabolic systems on the basis of their underlying genome. FBA using Ajo260 IJO1366 as GEM i use

Extensions of FBA Elemental modes ME-FBA

3.2 Applying ML to FBA

One problem with using the GEMs that other people have realized is that they can be overspecified. Since they contain every known reaction that is going on Reduction to core models LumpGem redGEM NetworkReducer MinNW

PCA + FBA Bayesian FBA Network models on FBA

3.3 Modeling Cell-Free systems

Modeling cell-free systems has typically taken the form of building kinetic models of transcription and translation. This involves the authors choosing which reactions they believe are most important and then writing out those reactions. In 2013, the Murray lab at CalTech applied this to a commercial cell-free system and was able to accurately describe the dynamics of their gene circuit of interest [17]. This was followed up by further work that expanded the [18] More recent work on this Recent work includes Bayesian parameter inference [19]

2012 cell free model remove by hand [20] Varner ssFBA build from scratch. sequence specific. [21]

Hybrid agent-based model for quantitative in-silico cell-free protein synthesis. 2016 https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5288458/

3.4 VAEs

We use a VAE in conjunction with FBA to model a cell-free system. Kingma 2013 and Welling 2014 first introduce the idea

Chapter 4

Design and Implementation

The primary deliverable of this dissertation is a system (shown in Figure 1) that is able to generate metabolic models of cell-free systems. Our system has four major stages. First, we ingest experimental data and incorporate it into a standardized format. Next, we convert the data into 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 reflected biological reality, we first had to generate the data we wanted to use for training. Gathering high quality biological data is difficult and was a crucial key 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, while the full protocol can be found in the Appendix.

4.1.1 Cell-free systems

As shown in Fig 4.1.1, 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 [22]. We begin by growing up 1L of BL21 E. coli cells in an overnight culture of LB until they have reached 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 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. See Appendix A for the full detailed protocol.

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 cAMP, maltodextrin, and NAD. They also include vital parts of transcription and translation such as NTPs, AAs, and tRNAs. See Table 4.1 for a complete listing of the reactants and the concentrations we use.

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.

Table 4.1: List of reactants for a 50 μL CFPS reaction. Note that MDX is a mixture of substrates and that AAs include all 20 of the amino acids.

Reactant	Amount (μL)	Purpose
MDX	5	Energy substrates
AAs	10	Building blocks for translation
PEG	2.5	Molecular crowding
HMP	1	Phosphate source
Mg	0.74	Important cofactor
K	0.8	Charge homeostasis
Cell Extract	25	TX/TL machinery
CFPS	50	Protein production

Table 4.2: My caption

Compound	Concentration (mM)
pi	0.981
mg	5.92
k	48.0
nad	0.454
coa	0.353
camp	1.01
folinic acid	0.092
spermidine	0.181
glucose	5.42
ATP/GTP	2.09
CTP/UTP	1.26
amino acids	35.6
trna	0.010
dna	0.071

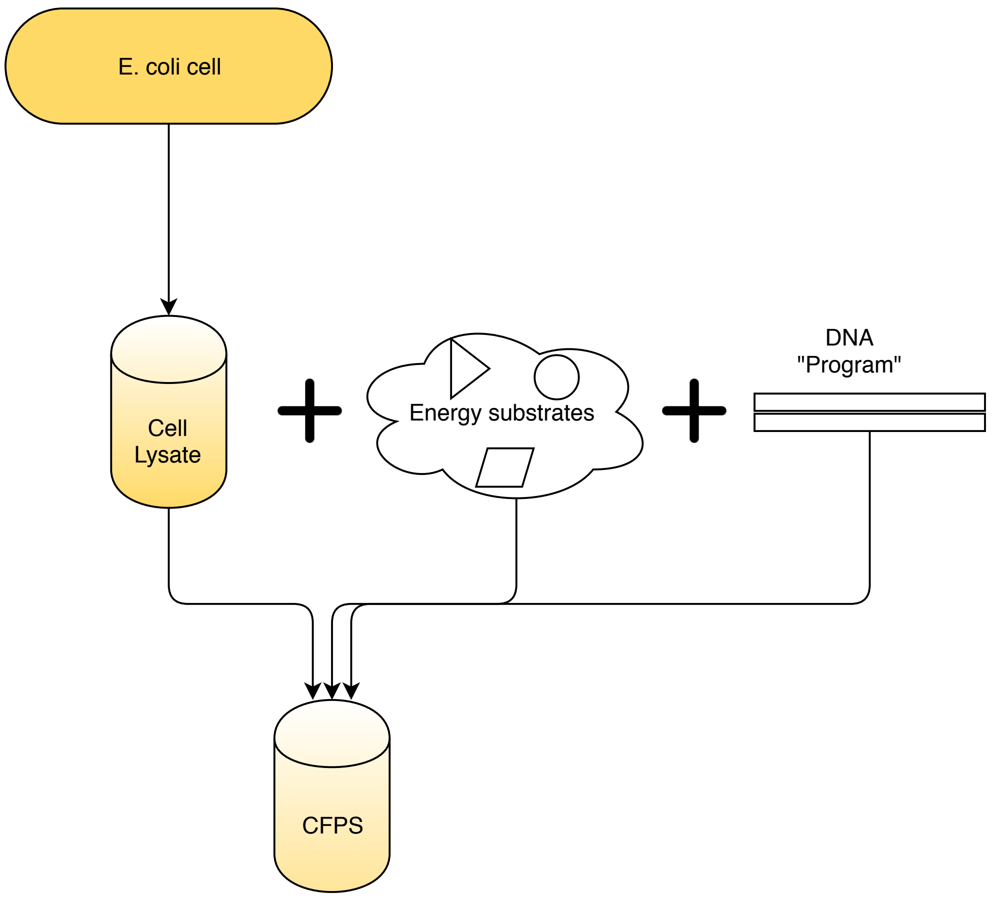


Figure 4.1: Overall process for creating a CFPS

4.1.2 Datasets

Once that protocol was established and standardized as a reliable means of protein production, we were able to create two datasets. Our first dataset followed the typical procedure a biologist in the lab would use to create a CFPS. We combined each of the ingredients by hand into a 50 uL master-mix and then used 5 uL aliquots of that for our test conditions. In order to test different reaction conditions, we added an additional 1 uL of sugar, phosphate, or nucleotides in differing ratios. These different ratios of additional reactants are our independent variables. Since we added just a simple DNA circuit that produces RFP, the fluorescence readout is our dependent variable.

However, there are multiple downsides to doing this by hand. First of all, it takes a long time and so there is a limit to the amount of data one can generate. Additionally, pipetting by hand has an intrinsic error. To get around these issues, we also generated a larger dataset using the Labcyte Echo robot. This is an acoustic liquid handler that allowed us to generate a dataset in the same spirit as before, though we were able to vary a much larger array of reactions conditions and perform the experiments at the 2 uL volume.

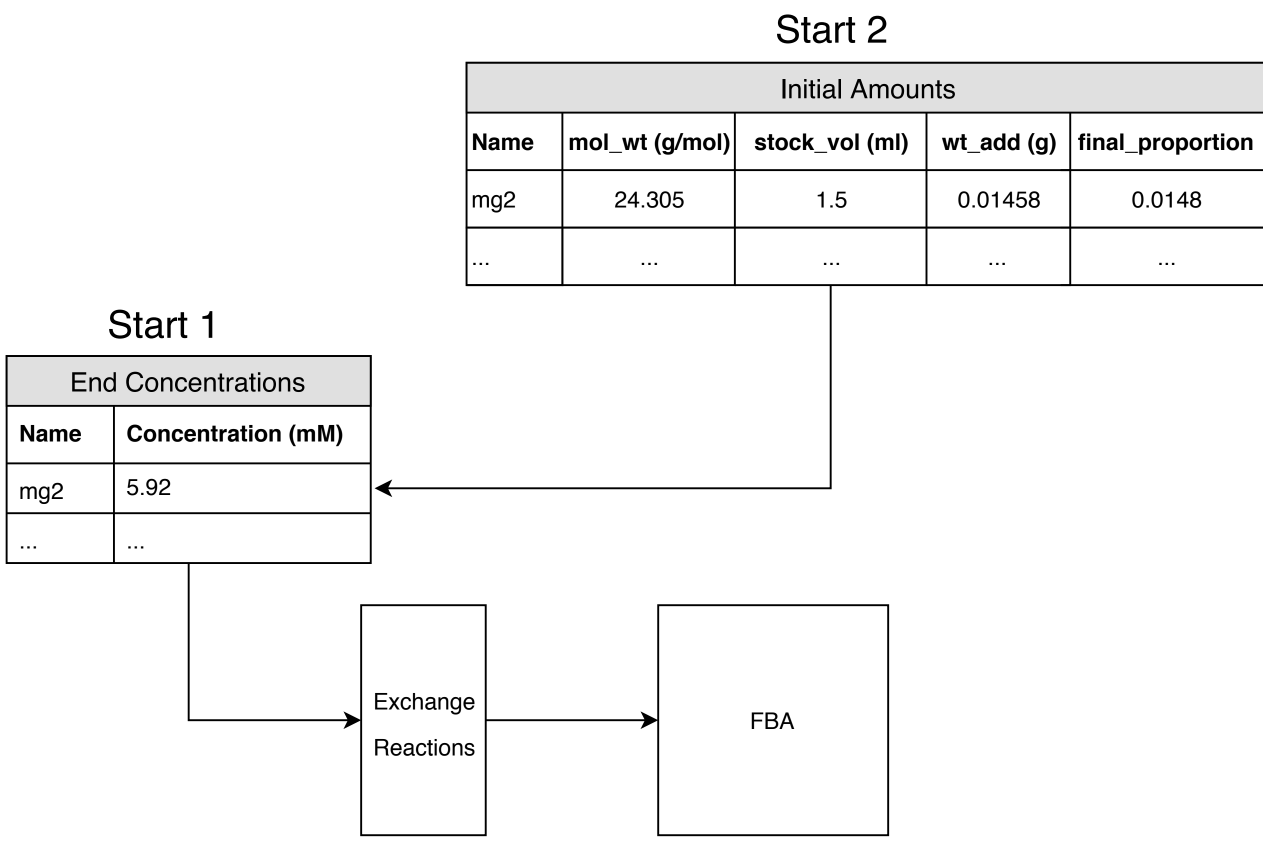
Finally, we received a third dataset from the recent Karim and Jewett paper [23]. Our protocol is based off of the CFPS from the Jewett lab, so it is a very similar setup. However, it was created in different lab conditions, so we can check to see if our model generalizes across lab conditions. Additionally, the pathway that they investigated is an inherently metabolic pathway, so we were able to easily incorporate it into our metabolic model.

4.2 Data ingestion and incorporation

2 csvs 1 for experimental setup 2 columns, one with canonical abbreviation of chemical, other with end concentration This data then gets converted into fluxes and put in as a constraining exchange reaction

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. There are 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 we are able to convert concentrations into fluxes. Thus, our tool is able to ingest that in the form



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:

of a CSV that has a column with the name of the reactant and the final concentration.

However, we also support biologists who just know the relative amounts of each reactant they add to the mixture without knowing the final concentration. Thus, we support a format whereby a user can upload a CSV that contains the 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. From that, we are able to calculate the final concentrations and manipulate the data into the same form as the first type of upload. Once we have that, we can unify the pipeline.

4.2.2 Data Incorporation

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 move the metabolite wholesale. Thus, at the end, we have all of the reactions occurring in the same location, which reflects biological reality.

Our next task is to remove exchanges and replace the medium with our data. We do this by using the concentrations we calculated from the setup data earlier. For each of the reactants that we have, we remove the previous exchange reactions. Then, we convert the final concentration to a flux using this equation: TODO. Finally, we add back in the exchange reactions with that flux as an upper bound. This represents the fact that the metabolite could be entirely used up, but it doesn't have to be used at that level. At the end of this, we have a base cell-free model.

We can use this model as a basis, but we can also extend this model to explicitly incorporate the transcription and translation reactions we care about. Given the sequence of the gene of interest, we automatically generate the hand-crafted cell-free FBA model from the Varner lab in Julia [21]. We can then convert that model to python and extract the transcription and translation reactions. We incorporate those reactions into our model and then can set the production of the protein of interest as our FBA objective.

Once we've done this, we need to create differential models based on the different experimental data setups. To do this, we take in a CSV consisting of the experimental conditions and a quantitative output. Each row represents a different experiment, while the columns represent different starting conditions. We can then re-calculate the final concentrations of each of the reactants for each experimental starting condition and create a modified FBA model for each starting conditions.

4.3 FBA and flux sampling

4.3.1 FBA

Were we simply to solve the different models we've created, many would have the same flux results. When we do solve these models without performing dimensionality reduction, we get a correlation of TODO (nearly 0) with our experimental results. Clearly, FBA alone cannot describe the full richness of a real system. 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 can do better by first passing the fluxes through a dimensionality reduction technique such as a VAE.

We also have the problem that the model is overspecified. These FBA models contain every reaction that's occurring in a steady state bacterium While we do harvest the bacteria at steady state, some enzymes will degrade and others will not be as important in cell-free systems. These types of changes can't be encapsulated solely through a change of objective function.

If we just run the naive FBA models on the different reaction conditions, we find that we have a correlation of almost 0. This means that the typical FBA model just doesn't describe CFPS So, we want to use a VAE to improve the way we describe CFPS. However, we first need a large dataset to train on. The way we accomplish this is by flux sampling from each model.

4.3.2 Flux sampling

In order to determine possible fluxes through each reaction, need to sample from it We use OptGP sampler to sample 2000 fluxes from each of our models. Can see that increasing the number of samples leads to a more normal distribution Figure **??** shows what happens as we increase the number of samples we draw from each model.

We sample from our different reactions to generate a new dataset. Each experimental condition has slightly different flux distributions for each reaction. This is important because it's these minor perturbation that we hope to use to create a specified model. We then combine these sampled fluxes into a dataset of size N x E x R where N is the number of samples, E is the number of experiments, and R is the number of reactions.

4.4 VAE

We next implemented a VAE in order to examine the latent space of the fluxes. Our original implementation simply took in the flux dataset as a nxD matrix X with a vector $nx1$ y that specified the "class" of each flux (i.e. which model it came from). We then ran it through a 2 layer autoencoder with layer sizes TODO and TODO. We experimented with latent dimensions of size 2 and 10. This created a latent space of size $nx2$.

We were able to explore this latent space. However, this only reflects information about the fluxes. We wanted to incorporate the experimental data we'd gathered–we know which models should be producing more fluxes than the others. We used this insight to create a new loss function for our autoencoder.

4.5 Correlated VAEs

We have additional information that we can use to perturb our latent space. We know what the relative rankings of the different models should be. So, instead of just having the typical loss function of an autoencoder: reconstruction loss and KL divergence, we add another term that we call correlation loss. We are forced to use pearson correlation loss because we can write that in a differentiable manner.

4.6 FBA model reduction

Finally, now that we have run our models through the Corr-VAE and achieved a latent representation, we need to use the insights we've generated to create better models. We can do this in two ways. Firstly, we could do this by any time we want to generate 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 shifted optimum and see if that still is better. This could be very useful for batch variation.

However, it would also be nice to not have to run it through an autoencoder each time. Instead, we could use the insights we've gained to actually create a more accurate reduced model. We do this by thresholding the latent space and removing reactions that have fluxes similar to 0. This allows us to create more useful reduced models. Now, running these reduced models with the differential conditions gives us a correlation of up to 0.5.

[24]

Chapter 5

Results

5.1 Comparison of models

5.2 Emphasis on where FBA-VAE does particularly well

5.3 Noise addition

5.4 Transfer learning

5.5 Biological insights

Chapter 6

Discussion and Future Work

6.1 Improved data gathering

As we've shown, increasing the amount of data is useful for training a more generalizable model. Generating lots of biological data is difficult, but there is a movement towards greater automation of lab tasks. In particular, we showed that a lab robot such as the Labcyte Echo can be used to generate larger scale datasets. We encourage future work to take this even further and generate datasets with hundreds or even thousands of starting conditions.

6.2 Different cell types

We have only explored the usage of this technique on a E. coli cell-free model. However, there is a lot of work currently ongoing to use non-E. coli cell-free models. This system was written with generalizability in mind and should transfer to other organisms as well. In particular, this was the intent with creating a reduction system instead of hand-building a model from scratch. There are dozens of FBA models of other organisms that this could be applied to.

In particular, the primary author of this work is a part of an Open Plant grant which involves the exploration of using wheat-germ cell-free systems. The data is currently being generated through a collaboration with the Earlham Institute. After the data generation is done, we want to apply this to that data to see its effectiveness. Additionally, this system will get a web interface to encourage other biologists to use it for their own data.

6.3 RL system for reduction

One thing that we noticed while thinking about the best way to reduce the model is that we could frame the problem in terms of a reinforcement learning problem. We originally tried to determine which reactions to remove by hand. However, the way we went about this was removing a reaction and re-running the model to see the output. Then, we compared how well we match the experimental data. So, we built a reinforcement learning framework to do this. Rewards are based on how well our collection of models reflects the experimental data.

Simply doing a naive search with this RL framework would take far too long. Starting with 2600 reactions, performing a random search over this space could take up to TODO. Thus, we could use the VAE that we wrote in conjunction with this framework to perform a better directed search. This could allow us to reduce the model even further and get even more specificity in the reduction.

6.4 Improved starting model

One potential issue that is limiting the usefulness of our pipeline is how well described the initial models are. They're only as good as they are allowed to be by the reactions that exist. FBA models undergo improvements periodically and we're using the most prominent model iJO1366. However, as this model is updated, our model can adapt and use more up-to-date reactions.

6.5 Improved VAE

6.6 Batch variation

Batch variation is an important problem facing the cell-free community. Some studies have shown up to 20% variation between batches for the same protocols and DNA constructs. This is an incredibly important issue because the typical workflow for a researcher may involve the characterization of a circuit through a series of different conditions or slightly different DNA programs. However, most people need to compare the performance of these variations. In order to deal with variation, researchers will perform all of their experiments with a single batch. When that batch runs out, though, they will be required to re-characterize and redo all of the experiments they have already done.

We offer two potential solutions with our system. First, a researcher could run a calibration set of experiments and then use that to generate a VAE model based on that data. Then, any future experiments that are performed (either in the same batch or a different one) could be projected into the same latent space in order to allow comparison between the experiments. A second solution would involve running a calibration for each batch and then determine the relative vector differences. This would allow the translation of all of the data for each batch

Appendix A

Experimental Setup

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] 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.

[4] 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.

[5] 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.

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

[7] 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.

[8] 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.

[9] 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.

[10] 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.

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

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

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

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

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

[16] 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.

[17] 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.

[18] 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.

[19] 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 analysis of cell-free transcription–translation reactions from nonmodel bac-

teria. *Proceedings of the National Academy of Sciences*, 115(19):E4340–E4349, 2018.

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

[21] 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.

[22] Preparation of cell-free rnapt7 reactions.

[23] 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.

[24] 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.