TOWARD A WHOLE-CELL MODEL OF *ESCHERICHIA COLI*

A DISSERTATION
SUBMITTED TO THE DEPARTMENT OF BIOENGINEERING
AND THE COMMITTEE ON GRADUATE STUDIES
OF STANFORD UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Derek Nathaniel Macklin
May 2017

This dissertation is online at: http://purl.stanford.edu/mg965qb5459

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Markus Covert, Primary Adviser

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Russ Altman

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Zev Bryant

Approved for the Stanford University Committee on Graduate Studies.

Patricia J. Gumport, Vice Provost for Graduate Education

This signature page was generated electronically upon submission of this dissertation in electronic format. An original signed hard copy of the signature page is on file in University Archives.

Abstract

Whole-cell computational models comprehensively simulate the growth and division of single cells, explicitly accounting for the functions of all known gene products and their interactions. Such models have the potential to revolutionize biology by serving as a platform to interpret complex behaviors, prioritize experiments, and enable design. In 2012, our lab completed the first whole-cell model of the simplest culturable organism, *Mycoplasma genitalium*. Since then we have focused our efforts on modeling *Escherichia coli*, one of the foundational model organisms in biology. In addition to having 10 times more genes and 50 times more molecules than *M. genitalium*, *E. coli* exhibits sophisticated regulation in response to environmental stimuli and perturbations. Currently, we have an *E. coli* model that incorporates the function of over 1200 genes and synthesizes tens of thousands of data points collected from both high- and low-throughput experiments performed over the last six decades. In building this model, we have incorporated many of *E. coli*'s feedback control mechanisms, included hundreds of kinetic constraints in a model of metabolism, decreased simulation runtime more than ten-fold, and demonstrated the ability of our simulated cells to reliably reproduce over multiple generations. Furthermore, we have used the model to explore behaviors that arise from the interactions of multiple biological processes. In doing so, we have uncovered and quantified the prevalence of sub-generational gene expression. As the model continues to expand in size and scope, we hope that it will further our understanding of cell physiology and find practical applications in synthetic biology and medicine.

Acknowledgments

I had no idea what to expect when I started grad school. I had applied simply because I thought it was the most interesting thing I could do. Little did I know how lucky I'd be to join a lab with amazing, world-class colleagues who'd make this a truly memorable experience. Little did I know how lucky I'd be to make some great friends—friends with whom I could both commiserate and celebrate. Little did I know how lucky I'd be to have the continued, unwavering support of my family throughout the entire process.

First I would like to thank my advisor, Markus Covert. After I had rotated in labs studying topics as diverse as medical imaging and synthetic biology, Markus, to my surprise, took a chance on me and let me try my hand at systems biology. Pretty quickly, I was hooked. I thoroughly enjoyed the combination of modeling and experiments. Systems biology was a way that I could use programming, my primary hobby growing up, to study some of the most complex and fascinating systems on the planet: cells. Throughout my time in the lab, Markus always made himself available to meet. During our many discussions, it was never lost on me that I was learning from not only a great scientist, but also a great person. Markus's endless enthusiasm, and his interest in not just the project, but the person (or people) behind the project are singular qualities that I admire and that I continually try to adopt.

I would also like to thank my committee members: Russ Altman, Zev Bryant, and KC Huang. When we weren't discussing basketball, our conversations always pushed me to re-examine my data and re-visit my assumptions. Their perspectives and feedback have helped me to be a better scientist.

None of this work would have been possible without my labmates. Nick Ruggero was crazy enough to join me in thinking that we could build a whole-cell model of *E. coli*, while Jonathan Karr and Jayodita Sanghvi were incredibly generous sharing lessons from their experience modeling *Mycoplasma genitalium*. Elsa Birch helped me with some of the first modeling I did in the lab while I was studying phage with Nate Maynard, and she also helped me and Nick innovate ways to better model metabolism, which John Mason and Morgan Paull then extended. Nate Maynard and later Mialy DeFelice taught me the techniques that I used to experimentally interrogate *E. coli*. Javier Carrera played a crucial role in reigning in the transcriptional regulatory network so that we could model it. Travis Horst and Heejo Choi brought enthusiasm and a critical eye to the *E. coli*

model as we wrapped up this iteration. Other labmates, including Jake Hughey, Miriam Gutschow, Keara Lane, Sergi Regot, Takamasa Kudo, David Van Valen, Katie Bodner, Stevan Jeknic, Silvia Carrasco, Inbal Maayan, and Kimberly Chin provided valuable discussions in lab meetings, lunches, and happy hours. While we were in the Clark Center, I could always count on Shatz Lab members Jaimie Adelson and Richie Sapp for an entertaining conversation whenever I passed by their benches.

When I started at Stanford, I felt woefully under-prepared for the graduate-level coursework. Fortunately, I had an amazing cadre of classmates who helped me get up to speed and whose friendship helped me weather the highs and lows of research over the coming years: Eric Chehab, Livia Yanez (and Javier Yanez, by courtesy appointment), Melina Mathur, Yuan Yao, Patrick Ye, Shrivats Iyer, Wenying Pan, Russ Toll, Haisam Islam, Kate Niehaus, Julie Dethier, Sylvie Liong, and Katelyn Cahill-Rowley. From problem sets to trivia nights to weddings, it's been a lot of fun (maybe minus the problem sets). I also want to thank my housemates Eric Chehab, Ben Mears, Cameron Kruse, and Patricia Penton for a great living environment.

Some of my most intellectually transformative endeavors in grad school came from my DOE Computational Science Graduate Fellowship. Not only did this provide me the impetus to take graduate-level math and computer science courses (in addition to funding me), but also the opportunity to work with Zhong Wang and Rob Egan at the Joint Genome Institute for three months. These experiences, in addition to the annual program review, taught me new perspectives and tools that proved critical to building this iteration of the *E. coli* whole-cell model.

I had my first experience doing research in Jack Judy's lab at UCLA, working with Neschae Fernando and later Jun Isobe. I still remember the feeling of excitement I had walking into the Engineering IV building, looking forward to sinking my teeth into an as-of-yet unsolved problem. I'd also like to thank Joe DiStefano for running the Computational and Systems Biology interdepartmental program. At the time, I didn't fully appreciate how unique the program was in terms of providing interdisciplinary training—that training has paid dividends many times over.

Memorable teachers also deserve a mention: Judy Wolthausen, Brian Safine, James Conn, Debra Troxell, Bob Kucer, Cheryl Unland. In different ways, they all imparted their love of learning and dared me to think deeply and critically.

Throughout all of the ups and downs, staying in touch with long-time friends—Scott Surrette (and, in fact, the whole Surrette family), David Kirakossian, Karla and Luz Taborga, Robbie Paolini—helped me stay grounded.

The Golden State Warriors—from all-stars Steph Curry, Klay Thompson, Draymond Green, and Kevin Durant, to the selfless role-players, to the coaching staff led by Steve Kerr—provided much-needed entertainment and served as an example of a tightly-knit, high-performing team who excel at and genuinely enjoy their work. Doug Macklin made it possible for me to attend their NBA playoffs and finals games while living on a grad student stipend.

Finally, I would like to thank my family. I'm incredibly fortunate to have so many aunts, uncles,

and cousins—too many to list here! In particular I want to thank Robert and Jan Kohne, who graciously opened their home to me when I worked at the Joint Genome Institute. My grandparents David and Gertrude Macklin, and Richard and Gabrielle Kohne, always encouraged my educational endeavors, whether it was to learn math, programming, or a foreign language. Even though they probably don't believe it, my younger brothers Doug and Richard Macklin have been role models for me as we've grown-up and transitioned to adulthood. Most importantly, I want to acknowledge my parents Neal and Renée Macklin. They are my lifelong teachers, role models, and friends, and remain my bedrock of support. I can't thank them enough.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Complexity surrounds us. From our physical environment to the organizations and institutions that we interact with, today complex and interwoven systems present us perhaps our greatest challenges and—if we can understand and harness them—our greatest opportunities. How do we manage complexity? How do we make it tractable?

This dissertation focuses on tackling complexity in a particular domain, namely biology. However, the tools and approaches that we use, mathematical modeling and computational simulation, are widely applicable. We hope this work serves as a case study demonstrating their effectiveness in dealing with complexity. We first used these tools to investigate how the replication of a virus, bacteriophage lambda, is incredibly dependent on a seemingly obscure aspect of the physiology of its host, the bacterium *Escherichia coli*. We then applied these tools to much larger systems, dynamically simulating the life cycles of *Mycoplasma genitalium* and later *Escherichia coli*. These simulations account for every molecule in the cell and work towards incorporating the function of every single gene (our simulations of *Mycoplasma genitalium* do incorporate the function of every gene). By linking diverse aspects of cell physiology we create a powerful platform to assess and increase our understanding of these organisms. Long-term, in close conjunction with experimental work, we hope this platform will one day enable the reliable, rational design and control of biological systems for medical and biotechnology applications.

1.1 Thesis contributions

One of the many great things about working in the Covert Lab has been the opportunity to work on an incredibly rich and diverse set of problems. My contributions to these projects, which constitute the body of this dissertation, are:

- **Understanding the role of host tRNA modification in lambda phage infectivity**.

In a study published in 2010, members of the Covert lab sought to understand which of *E. coli*'s genetic factors inhibited or enhanced lambda phage's ability to infect it [3]. Using a library of nearly 4000 single-gene deletion strains [4], they assessed lambda phage infectivity of these genetic mutants using plaque assays. They followed up on hits by infecting liquid cultures of *E. coli* with lambda phage and measured their growth dynamics on a plate reader. One of the most puzzling results from this study was an iron-sulfur cluster biosynthesis pathway implicated in modulating lambda phage infectivity.

How does iron-sulfur cluster biosynthesis affect lambda phage's ability to infect its *E. coli* host? This was the question that postdoc Nate Maynard was attempting to answer when I started working in the lab. As he trained me on the requisite experimental techniques, we began to uncover this mystery. Concurrently, under Markus's guidance, I developed and parameterized a mathematical model of the iron-sulfur cluster biosynthesis pathway and linked it to a model of phage infection that my labmate Elsa Birch had developed during the previous phage screen study [3]. The modeling process helped me and Nate carefully consider this complex system to refine our hypotheses and prioritize our experiments. Ultimately we found that iron-sulfur cluster biosynthesis affects *E. coli*'s tRNA modification mechanisms, which in turn affects the translational fidelity of critical lambda phage components. This work is presented in Appendix A and was published in 2012 [5].

- **Analyzing the first gene-complete whole-cell model of *Mycoplasma genitalium*.**

  During the lambda phage project I came to appreciate the value in modeling a complex system to better understand it. One of the things that bothered me, though, was the need to make certain modeling assumptions regarding the rest of the system (the rest of the cell)—the need to assume that certain quantities remained constant or weren't affected by the components that we were explicitly modeling. At that time, my labmates Jayodita Sanghvi and Jonathan Karr were completing a computational model of *Mycoplasma genitalium* (the smallest known self-replicating organism) that included the function of every annotated gene and accounted for every molecule in the cell. Intrigued by their approach and the potential applications for synthetic biology (and naively assuming it couldn't be *that* difficult) I started working with them to help analyze the output of their simulations and uncover unexpected behaviors.

  As I started working with Jonathan and Jayodita, three big questions became apparent: (1) How do you effectively organize and store data from thousands of simulations in a manner that is amenable to exploration? The simulation output totaled a few terabytes of data and it was not homogeneous in the sense that it wasn't all just data on, say, positions of particles. The hybrid nature of the model meant that some output would represent reaction fluxes, while other output would represent counts of molecules, while still other output would represent the occupancy of the chromosome at nucleotide-level resolution. (2) How do you visualize

simulation output, both for exploration and communication purposes? The heterogeneous nature of the output data meant that aside from plotting quantities of interest against time, the choice of visual representation for maximizing information content was non-obvious. In the field of data visualization, this would likely be considered a challenge in information visualization rather than scientific visualization (scientific visualization focuses on displaying data with a strong spatial component [6]). (3) How do you benchmark and assess model performance? In addition to demonstrating that the model was well-fit, we also needed to demonstrate that it could make useful predictions and provide non-trivial biological insight. The initial results of our work are presented in Chapter 2 and was published in 2012 [7]. Jonathan took the lead in further attempting to address Question (1) by releasing WholeCellSimDB [8] and Question (2) by releasing WholeCellViz [9], while Jayodita took the lead in further addressing Question (3) by using the model to correctly predict kinetic parameters of enzymes using growth rates of gene deletion strains [10] (and of course Markus oversaw these projects). Despite these efforts, I would argue that none of these three questions have satisfactorily been answered—they are still, and will continue to be, ripe areas of research.

- **Developing the core of an *E. coli* whole-cell model**.

  The biggest long-term challenge that I saw with the *Mycoplasma* whole-cell model was that it was difficult to validate (this challenge and others are discussed in more detail in Chapter 3). *M. genitalium* lacked robust tools for genetic manipulation and it was difficult to experimentally interrogate. A model of *E. coli*, on the other hand, would be considerably easier to parameterize and validate. There are well-established protocols for working with *E. coli*, and much of the early molecular biology work was performed in *E. coli* [11]. The trade-off we had to make, though, was that *E. coli* contained about 10 times as many genes and 50 times as many molecules as *M. genitalium*—it would be much more difficult to work with from a computational standpoint.

  As a result, a lot of the early work on the *E. coli* whole-cell model was an engineering effort. Nick Ruggero, John Mason, and I rewrote the pipeline and data structures for the whole-cell simulations, resulting in an order of magnitude improvement in run-time. *E. coli* simulations take 10-15 minutes to run, compared to the 10-14 hours required for the *Mycoplasma* simulations, despite accounting for 50 times the number of molecules (Chapter 4). With this much-improved run-time, we could focus our efforts on novel modeling.

  In terms of modeling, we have made a number of improvements over the *Mycoplasma* work. Our *E. coli* simulations can now grow and double reliably for multiple generations (Chapter 4). We can simulate *E. coli* in different media conditions at different growth rates (demonstrated in Chapter 7). We also have a much more detailed and quantitative model of transcriptional regulation due to experimental and computational work performed by Javier Carrera, Heejo

Choi, and myself (Chapter 5). Furthermore, due to efforts by Travis Horst, Markus Covert, Morgan Paull, and myself, we have a much more constrained model of metabolism that incorporates hundreds of kinetic parameters. We hope that this core *E. coli* model can be rapidly expanded to a gene-complete model.

I have also had the opportunity to contribute to a number of other projects more peripherally. To fulfill the requirements of my Department of Energy Computational Science Graduate Fellowship (CSGF), I worked at the Joint Genome Institute in Walnut Creek, CA for three months. There I collaborated with Zhong Wang and Rob Egan to develop the Feature Array Search Tool for Estimating Resemblance of proteins (FASTERp).

During my coursework, I was able to complete final projects that contributed to other efforts in the lab. With Harendra Guturu, I made an interactive figure to visualize NF-κB dynamics, which was included as a supplemental figure to [12]. I also helped make a machine learning classifier to identify mis-labeled cells in the lab's image analysis pipeline [13].

As I became more senior in the lab, I was able to contribute computational expertise to other projects. I helped David Van Valen scale, package, and distribute his DeepCell image segmentation framework [14] so that other members of our lab and KC Huang's lab could run it on the Stanford Sherlock cluster. I also assembled a Docker image to share the work with the broader scientific community [15]. Additionally, based on my experience performing RNA-seq in *E. coli*, I provided guidance on constructing the single-cell RNA-seq analysis pipeline used by Keara Lane and David Van Valen in their work linking NF-κB translocation dynamics to gene expression [16].

It's been incredible having the opportunity to work on so many projects with such a great group of people. I can only hope that my future endeavors will be as stimulating and exciting!

1.2 Structure of this dissertation

The remainder of this dissertation is structured as follows:

- Chapter 2 is a re-print of the *Cell* paper describing the first gene-complete whole-cell model of *Mycoplasma genitalium*—work I performed with Jonathan Karr, Jayodita Sanghvi, and Markus Covert.
- Chapter 3 is a re-print of a perspective paper from *Current Opinion in Biotechnology* that I worked on with Nick Ruggero and Markus Covert. This summarizes our early experiences in whole-cell modeling, enumerating the many challenges we face in the field going forward. Building a whole-cell model of *E. coli* has been my attempt to address some of these challenges.
- Chapter 4 discusses how we addressed the engineering challenges we faced while building the *E. coli* model. This work was done with Nick Ruggero, John Mason, and Kalli Kappel.

- Chapter 5 describes experimental and modeling efforts to construct a quantitative genome-scale model of transcription within the context of an *E. coli* whole-cell model. This work was done with Javier Carrera, Heejo Choi, and Markus Covert.
- Chapter 6 describes the efforts to construct a kinetically-constrained genome-scale model of metabolism in a whole-cell model of *E. coli*. This work was done with with Travis Horst, Morgan Paull, John Mason, and Markus Covert.
- Chapter 7 is a manuscript submitted to Nature that uses the model to assess the consistency of the existing data sets that describe *E. coli* physiology by revisiting Francis Crick's goal of obtaining a "complete solution" of *E. coli*. It also describes emergent phenomena that we observe from simulation output. This work was done with Nick Ruggero, Javier Carrera, Heejo Choi, Travis Horst, Mialy DeFelice, Sam Bray, and Markus Covert.
- Chapter 8 concludes the body of the dissertation with perspectives on the field going forward. I revisit and expand upon some of the issues discussed in Chapter 3.
- Appendix A is a re-print of the *Molecular Systems Biology* paper describing my phage work performed with Nate Maynard and Markus Covert.
- Appendix B is the supplemental material for the paper re-printed in Appendix A.
- Appendix C is the supplemental material for the manuscript re-printed in Chapter 7.

Chapter 2

A Whole-Cell Computational Model Predicts Phenotype from Genotype

Abstract

Understanding how complex phenotypes arise from individual molecules and their interactions is a primary challenge in biology that computational approaches are poised to tackle. We report a whole-cell computational model of the life cycle of the human pathogen *Mycoplasma genitalium* that includes all of its molecular components and their interactions. An integrative approach to modeling that combines diverse mathematics enabled the simultaneous inclusion of fundamentally different cellular processes and experimental measurements. Our whole-cell model accounts for all annotated gene functions and was validated against a broad range of data. The model provides insights into many previously unobserved cellular behaviors, including in vivo rates of protein-DNA association and an inverse relationship between the durations of DNA replication initiation and replication. In addition, experimental analysis directed by model predictions identified previously undetected kinetic parameters and biological functions. We conclude that comprehensive whole-cell models can be used to facilitate biological discovery.

Chapter reproduced from: JR Karr*, JC Sanghvi*, DN Macklin, MV Gutschow, JM Jacobs, B Bolival, N Assad-Garcia, JI Glass, MW Covert. "A Whole-Cell Computational Model Predicts Phenotype from Genotype" *Cell*. 2012. 150(2): 389-401

2.1 Introduction

Computer models that can account for the integrated function of every gene in a cell have the potential to revolutionize biology and medicine, as they increasingly contribute to how we understand, discover, and design biological systems [17]. Models of biological processes have been increasing in complexity and scope [18–20], but with efforts at increased inclusiveness of genes, parameters, and molecular functions come a number of challenges.

Two critical factors in particular have hindered the construction of comprehensive, "whole-cell" computational models. First, until recently, not enough has been known about the individual molecules and their interactions to completely model any one organism. The advent of genomics and other high-throughput measurement techniques has accelerated the characterization of some organisms to the extent that comprehensive modeling is now possible. For example, the mycoplasmas, a genus of bacteria with relatively small genomes that includes several pathogens, have recently been the subject of an exhaustive experimental effort by a European consortium to determine the transcriptome [21], proteome [22], and metabolome [23] of these organisms.

The second limiting factor has been that no single computational method is sufficient to explain complex phenotypes in terms of molecular components and their interactions. The first approaches to modeling cellular physiology, based on ordinary differential equations (ODEs) [24–29], were limited by the difficulty in obtaining the necessary model parameters. Subsequently, alternative approaches were developed that require fewer parameters, including Boolean network modeling [30] and constraint-based modeling [20, 31]. However, the underlying assumptions of these methods do not apply to all cellular processes and conditions, and building a whole-cell model entirely based on either method is therefore impractical.

Here, we present a "whole-cell" model of the bacterium *Mycoplasma genitalium*, a human urogenital parasite whose genome contains 525 genes [32]. Our model attempts to: (1) describe the life cycle of a single cell from the level of individual molecules and their interactions; (2) account for the specific function of every annotated gene product; and (3) accurately predict a wide range of observable cellular behaviors.

2.2 Results

2.2.1 Whole-Cell Model Construction and Integration

Our approach to developing an integrative whole-cell model was to divide the total functionality of the cell into modules, model each independently of the others, and integrate these submodels together. We defined 28 modules (Figure 2.1A) and independently built, parameterized, and tested a submodel of each. Some biological processes have previously been studied quantitatively and in depth, whereas other processes are less well characterized or are hardly understood. Consequently,

each module was modeled using the most appropriate mathematical representation. For example, metabolism was modeled using flux-balance analysis [33], whereas RNA and protein degradation were modeled as Poisson processes.

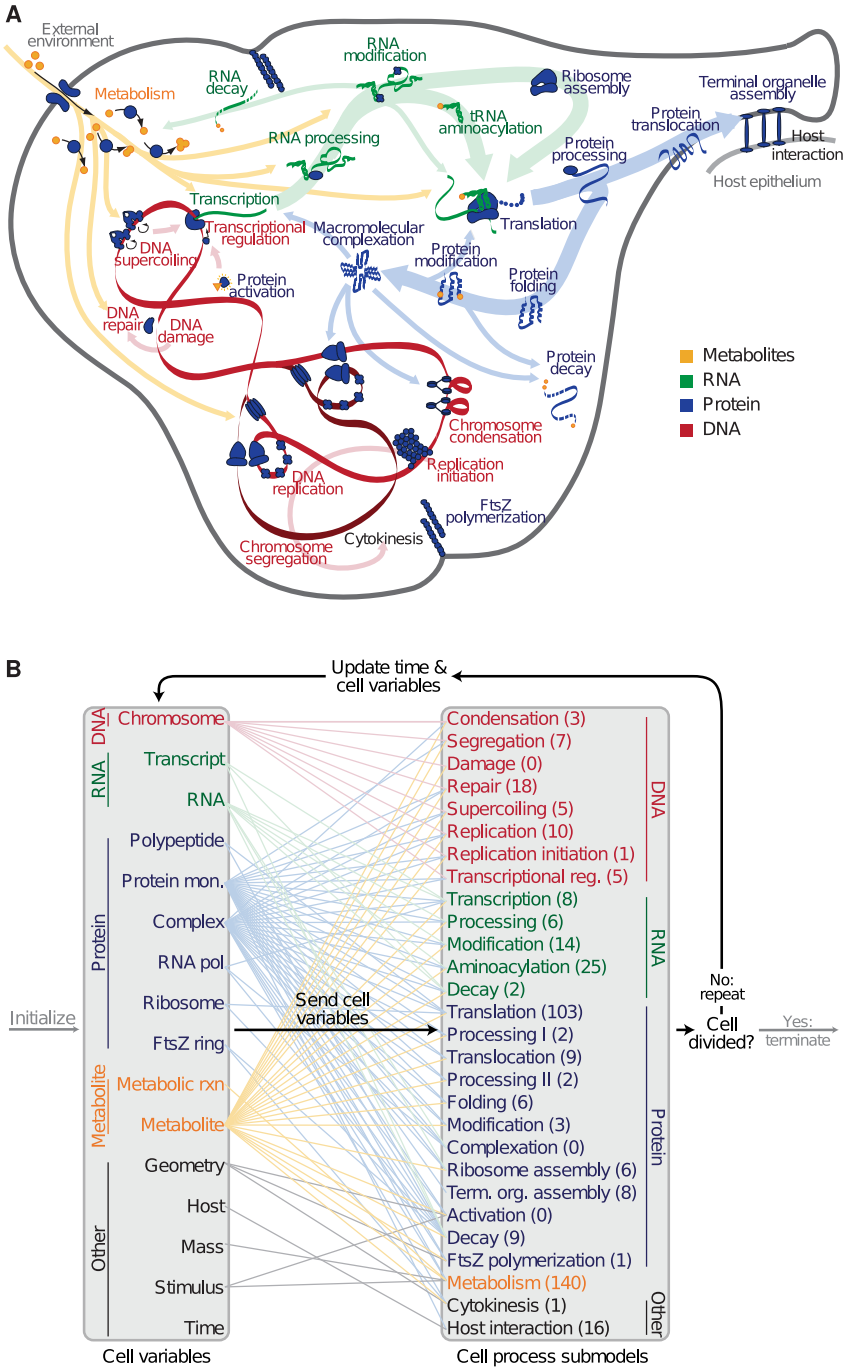


Figure 2.1: *M. genitalium* Whole-Cell Model Integrates 28 Submodels of Diverse Cellular Processes. (A) Diagram schematically depicts the 28 submodels as colored words—grouped by category as metabolic (orange), RNA (green), protein (blue), and DNA (red)—in the context of a single *M. genitalium* cell with its characteristic flask-like shape. Submodels are connected through common metabolites, RNA, protein, and the chromosome, which are depicted as orange, green, blue, and red arrows, respectively. (B) The model integrates cellular function submodels through 16 cell variables. First, simulations are randomly initialized to the beginning of the cell cycle (left gray arrow). Next, for each 1 s time step (dark black arrows), the submodels retrieve the current values of the cellular variables, calculate their contributions to the temporal evolution of the cell variables, and update the values of the cellular variables. This is repeated thousands of times during the course of each simulation. For clarity, cell functions and variables are grouped into five physiologic categories: DNA (red), RNA (green), protein (blue), metabolite (orange), and other (black). Colored lines between the variables and submodels indicate the cell variables predicted by each submodel. The number of genes associated with each submodel is indicated in parentheses. Finally, simulations are terminated upon cell division when the septum diameter equals zero (right gray arrow).

A key challenge of the project was to integrate the 28 submodels into a unified model. Although we and others had previously developed methods to integrate ODEs with Boolean, probabilistic, and constraint-based submodels [18, 34–36], the current effort involved so many different cellular functions and mathematical representations that a more general approach was needed. We began with the assumption that the submodels are approximately independent on short timescales (less than 1 s). Simulations are then performed by running through a loop in which the submodels are run independently at each time step but depend on the values of variables determined by the other submodels at the previous time step. Figure 2.1B summarizes the simulation algorithm and the relationships between the submodels and the cell variables. Data S1 (available online) provides a detailed description of the complete modeling process, including reconstruction and computational implementation.

2.2.2 Model Training and Parameter Reconciliation

Our model is based on a synthesis of over 900 publications and includes more than 1,900 experimentally observed parameters. Most of these parameters were implemented as originally reported. However, several other parameters were carefully reconciled; for example, the experimentally measured DNA content per cell [37, 38] represents less than one-third of the calculated mass of the mycoplasma chromosome. Data S1 details how we resolved this and several similar discrepancies among the experimentally observed parameters.

Once the model was implemented and all parameters were reconciled, we verified that the model recapitulates key features of our training data. We simulated 128 wild-type cells in a typical *Mycoplasma* culture environment, with each simulation predicting not only cellular properties such as the cell mass and growth rate but also molecular properties including the count, localization, and activity of each molecule (Movie S1 illustrates the life cycle of one in silico cell). We found that

the model calculations were consistent with the observed doubling time (Figures 2.2A and 2.2B), cellular chemical composition (Figure 2.2C), replication of major cell mass fractions (Figure 2.2D), and gene expression ($R^2 = 0.68$).

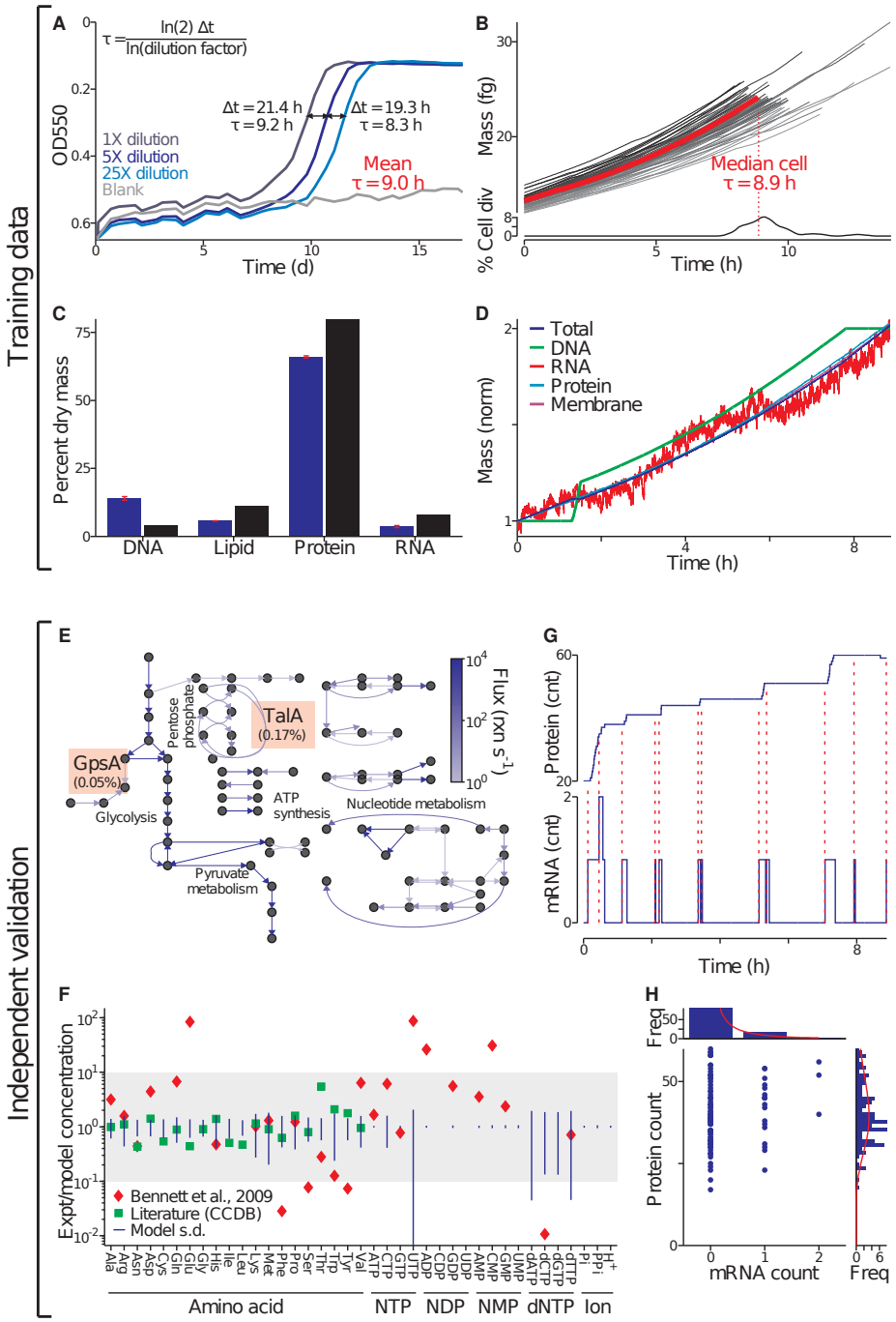


Figure 2.2: The Model Was Trained with Heterogeneous Data and Reproduces Independent Experimental Data across Multiple Cellular Functions and Scales.
(A) Growth of three cultures (dilutions indicated by shade of blue) and a blank control measured by OD550 of the pH indicator phenol red. The doubling time, τ, was calculated using the equation at the top left from the additional time required by more dilute cultures to reach the same OD550 (black lines).
(B) Predicted growth dynamics of one life cycle of a population of 64 in silico cells (randomly chosen from the total simulation set). Median cell is highlighted in red. Distribution of cell-cycle lengths is shown at bottom.
(C) Comparison of the predicted and experimentally observed [37] cellular chemical compositions. Red bars indicate model SD; Morowitz et al. (1962) did not report SD.
(D) Temporal dynamics of the total cell mass and four cell mass fractions of a representative in silico cell. Mass fractions are normalized to their initial values.
(E) Average predicted metabolic fluxes. Arrow brightness indicates flux magnitude. The ratios of the GpsA and TalA fluxes to the Glk flux are indicated in orange boxes and are comparable to experimental data [23].
(F) Ratios of observed [39, 40] and average predicted concentrations of 39 metabolites. Blue bars indicate model SD.
(G) Temporal dynamics of cytadherence high-molecular-weight protein 2 (HMW2, MG218) mRNA and protein expression of one in silico cell. Red dashed lines indicate the direct link between mRNA synthesis and subsequent bursts in protein synthesis.
(H) HMW2 mRNA and protein copy number distribution of an unsynchronized population of 128 in silico cells. Histograms indicate the marginal distributions of the copy numbers of mRNA (top) and protein (right). Red lines indicate log-normal regressions of these marginal distributions. The absence of correlation between the copy numbers of mRNA and protein and the shapes of the marginal distributions is consistent with recent single-cell measurements by [41].

2.2.3 Model Validation against Independent Experimental Data

Next, we validated the model against a broad range of independent data sets that were not used to construct the model and which encompass multiple biological functions—metabolomics, transcriptomics, and proteomics—and scales from single cells to populations. In agreement with earlier reports [23], the model predicts that the flux through glycolysis is >100-fold more than that through the pentose phosphate and lipid biosynthesis pathways (Figure 2.2E). Furthermore, the predicted metabolite concentrations are within an order of magnitude of concentrations measured in Escherichia coli for 100% of the metabolites in one compilation of data [39] and for 70% in a more recent high-throughput study [40] (Figure 2.2F). Our model also predicts "burst-like" protein synthesis due to the local effect of intermittent messenger RNA (mRNA) expression and the global effect of stochastic protein degradation on the availability of free amino acids for translation, which is comparable to recent reports by [42] and [43] (Figure 2.2G). The mRNA and protein level distributions predicted by our model are also consistent with recently reported single-cell measurements (Figure 2.2H; compare to [41]). Taking all of these specific tests of the model's predictions together, we concluded that our model recapitulates experimental data across multiple biological functions and scales.

2.2.4 Prediction of DNA-Binding Protein Interactions

Models are often used to predict molecular interactions that are difficult or prohibitive to investigate experimentally, and our model offers the opportunity to make such predictions in the context of the entire cell. Whereas previous studies have either focused on the genomic distribution of DNA-binding proteins [44] or on the detailed diffusion dynamics of specific DNA-binding proteins [45], the whole-cell model can predict both the instantaneous protein chromosomal occupancy as well as the temporal dynamics and interactions of every DNA-binding protein at the genomic scale at single-cell resolution. Figure 2.3A illustrates the average predicted chromosomal protein occupancy as well as the predicted chromosomal occupancies for DNA and RNA polymerase and the replication initiator DnaA, which are three of the 30 DNA-binding proteins represented by our model. Consistent with a recent experimental study by [44], the predicted high-occupancy RNA polymerase regions correspond to highly transcribed ribosomal RNAs (rRNAs) and transfer RNAs (tRNAs). In contrast, the predicted DNA polymerase chromosomal occupancy is significantly lower and biased toward the terC (see below for further discussion).

The model further predicts that the chromosome is explored very rapidly, with 50% of the chromosome having been bound by at least one protein within the first 6 min of the cell cycle and 90% within the first 20 min (Figure 2.3B). RNA polymerase contributes the most to chromosomal exploration, binding 90% of the chromosome within the first 49 min of the cell cycle. On average, this results in expression of 90% of genes within the first 143 min (Figure 2.3C), with transcription lagging RNA polymerase exploration due to the significant contribution of nonspecific RNA polymerase-DNA interactions to RNA polymerase diffusion [46].

The model also predicts protein-protein collisions on the chromosome. Previous researchers have studied the collisions of pairs of specific proteins [47], but experimentally determining the collisions among all pairs of DNA-binding proteins at the genomic scale at single-cell resolution is currently infeasible. Our model predicts that over 30,000 collisions occur on average per cell cycle, leading to the displacement of 0.93 proteins per second. Figure 2.3D illustrates the binding dynamics of the same proteins depicted in Figure 2.3A over the course of the cell cycle for one representative simulation and highlights several protein-protein collisions. Further categorization of the predicted collisions by chromosomal location indicates that the frequency of protein-protein collisions correlates strongly with DNA-bound protein density across the genome (Figure 2.3F) and that the majority of collisions are caused by RNA polymerase (84%) and DNA polymerase (8%), most commonly resulting in the displacement of structural maintenance of chromosome (SMC) proteins (70%) or single-stranded binding proteins (6%) (Figure 2.3E and Table S2F).

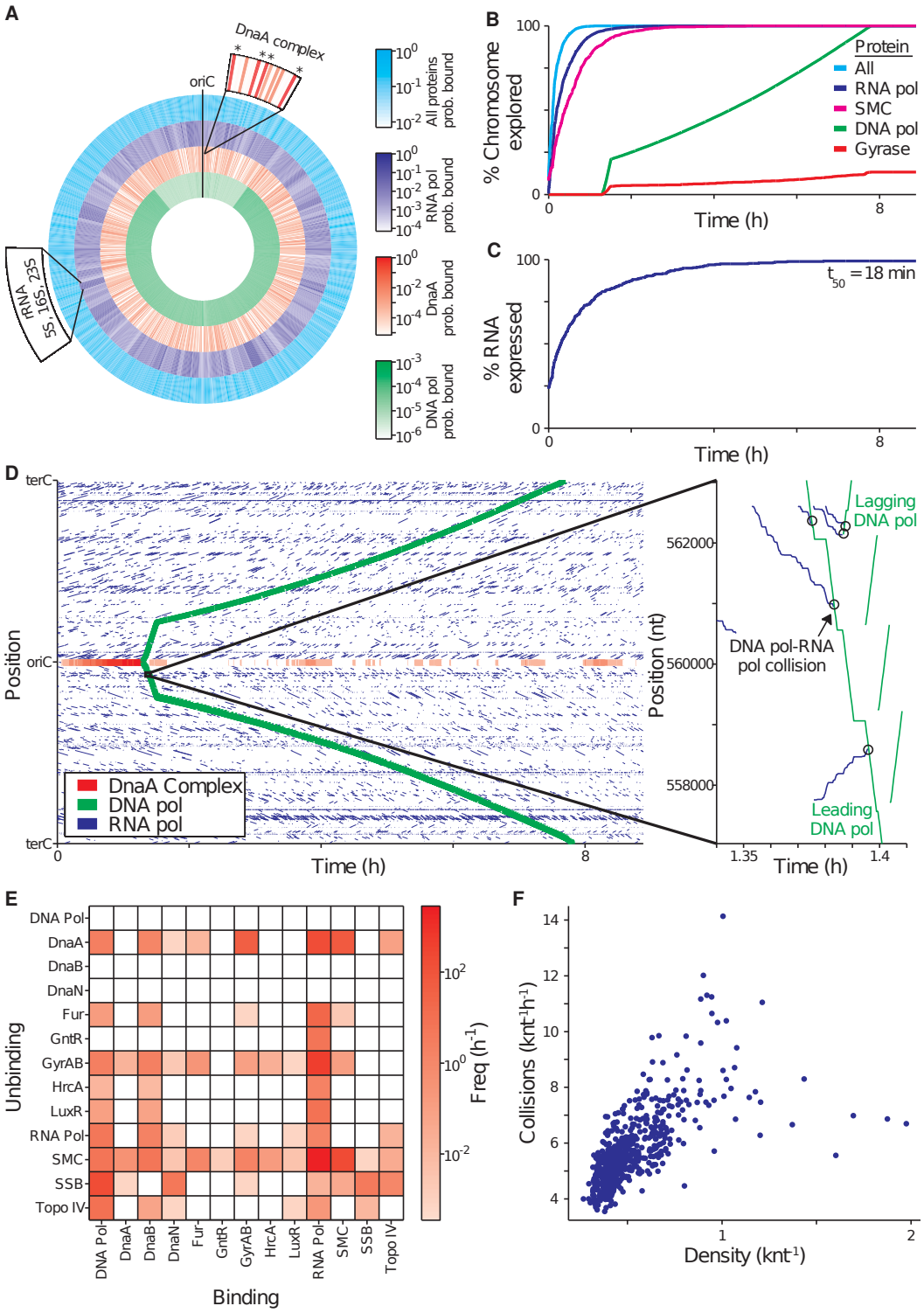

A
DnaA complex
oriC
rRNA 5S, 16S, 23S
All proteins prob. bound
RNA pol prob. bound
DnaA prob. bound
DNA pol prob. bound
B
% Chromosome explored
Protein
All
RNA pol
SMC
DNA pol
Gyrase
Time (h)
C
% RNA expressed
t50 = 18 min
Time (h)
D
terC
oriC
terC
Position
DnaA Complex
DNA pol
RNA pol
Time (h)
Position (nt)
Lagging DNA pol
DNA pol-RNA pol collision
Leading DNA pol
Time (h)
E
DNA Pol
DnaA
DnaB
DnaN
Fur
GntR
GyrAB
HrcA
LuxR
RNA Pol
SMC
SSB
Topo IV
Unbinding
Binding
Freq (h-1)
F
Collisions (knt-1h-1)
Density (knt-1)

Figure 2.3: The Model Highlights the Central Physiological Role of DNA-Protein Interactions.
(A) Average density of all DNA-bound proteins and of the replication initiation protein DnaA and DNA and RNA polymerase of a population of 128 in silico cells. Top magnification indicates the average density of DnaA at several sites near the oriC; DnaA forms a large multimeric complex at the sites indicated with asterisks, recruiting DNA polymerase to the oriC to initiate replication. Bottom left indicates the location of the highly expressed rRNA genes.
(B and C) Percentage of the chromosome that is predicted to have been bound (B) and the number of genes that are predicted to have been expressed (C) as functions of time. SMC is an abbreviation for the name of the chromosome partition protein (MG298).
(D) DNA-binding and dissociation dynamics of the oriC DnaA complex (red) and of RNA (blue) and DNA (green) polymerases for one in silico cell. The oriC DnaA complex recruits DNA polymerase to the oriC to initiate replication, which in turn dissolves the oriC DnaA complex. RNA polymerase traces (blue line segments) indicate individual transcription events. The height, length, and slope of each trace represent the transcript length, transcription duration, and transcript elongation rate, respectively. The inset highlights several predicted collisions between DNA and RNA polymerases that lead to the displacement of RNA polymerases and incomplete transcripts.
(E) Predicted collision and displacement frequencies for pairs of DNA-binding proteins.
(F) Correlation between DNA-binding protein density and frequency of collisions across the chromosome. Both (E) and (F) are based on 128 cell-cycle simulations.

2.2.5 Identification of Metabolism as an Emergent Cell-Cycle Regulator

The model can also highlight interesting aspects of cell behavior. In reviewing our model simulations, we noticed variability in the cell-cycle duration (Figure 2.2B) and wanted to determine the source of that variability. The model representation of the *M. genitalium* cell cycle consists of three stages: replication initiation, replication itself, and cytokinesis. We found that there was relatively more cell-to-cell variation in the durations of the replication initiation (64.3%) and replication (38.5%) stages than in cytokinesis (4.4%) or the overall cell cycle (9.4%; Figure 2.4A). This data raised two questions: (1) what is the source of duration variability in the initiation and replication phases; and (2) why is the overall cell-cycle duration less varied than either of these phases?

With respect to the first question, replication initiation occurs as DnaA protein monomers bind or unbind stochastically and cooperatively to form a multimeric complex at the replication origin (Figure 2.4B, top) [25]. When the complex is complete, DNA polymerase gains access to the origin, and the complex is displaced. We found a correlation ($R^2 = 0.49$) between the predicted duration of replication initiation and the initial number of free DnaA monomers (Figure 2.4C); however, the low correlation indicated that the duration depends on more than the initial conditions. In particular, we observed that the stochastic aspect of the transcription and translation submodels creates variability in the number of new DnaA monomers produced over time, as well as the DnaA-binding and -unbinding events themselves. This indicates that the variability in replication initiation duration depends not only on variability in initial conditions but also in the simulation itself.

As to the second question, because the replication submodel is substantially more deterministic than the initiation submodel, we expected to find a straightforward relationship between the progress

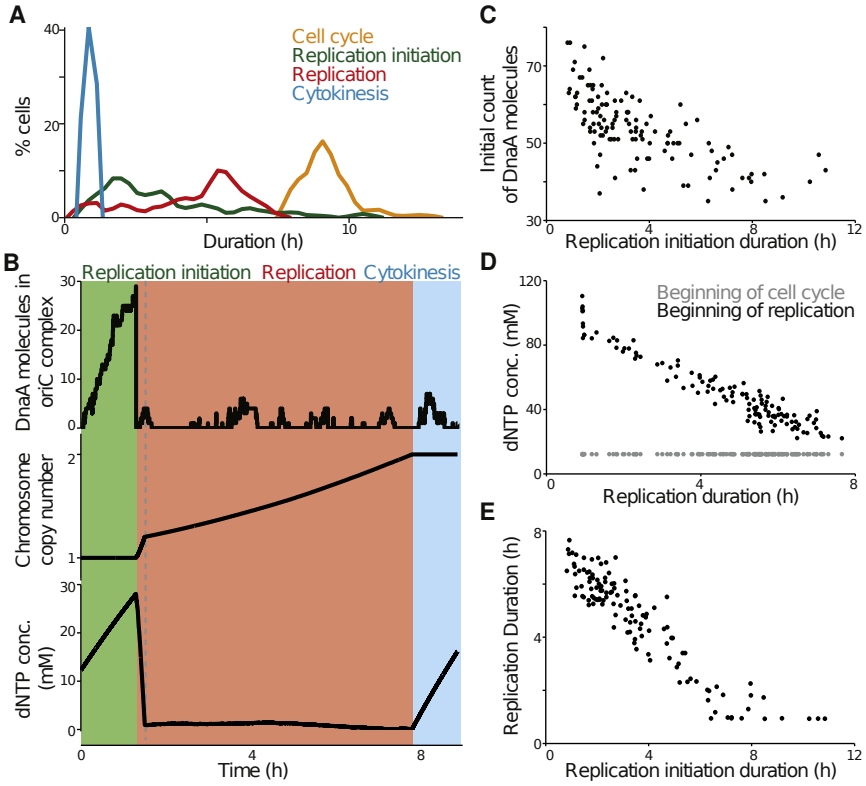


Figure 2.4: The Model Predictions Regarding Regulation of the Cell-Cycle Duration.
(A) Distributions of the duration of three cell-cycle phases, as well as that of the total cell-cycle length, across 128 simulations.
(B) Dynamics of macromolecule abundance in a selected cell simulation. Top, the size of the DnaA complex assembling at the oriC (in monomers of DnaA); middle, the copy number of the chromosome; and bottom, the cytosolic dNTP concentration. The quantities of these macromolecules correlate strongly with the timing of key cell-cycle stages.
(C) Correlation between the initial cellular DnaA content and the duration of the replication initiation cell-cycle stage across the same 128 in silico cells depicted in (A).
(D) Correlation between the dNTP concentrations (both at the beginning of the cell cycle and at the beginning of replication) and the duration of replication across the same 128 in silico cells depicted in (A).
(E) Correlation between the duration of replication initiation and replication across the same 128 in silico cells depicted in (A).

of replication and the cell cycle. Instead, the model predicts that DNA replication proceeds at two distinct rates during the cell cycle. This is reflected in the motion and DNA-binding density of DNA polymerase (Figures 2.3A and 2.3D) and in the dynamics of DNA synthesis as compared to the synthesis of other macromolecules (Figure 2.4B, middle). Initially, replication proceeds quickly due to the free deoxyribonucleotide triphosphate (dNTP) content in the cell (Figure 2.4B, bottom).

When DNA polymerase initially binds to the replication origin, dNTPs are abundant, and replication proceeds unimpeded. When the dNTP pool is exhausted, however, the rate of replication slows to the rate of dNTP synthesis. Accordingly, the duration of the replication phase in individual cells is more closely related to the free dNTP content at the start of replication than to the dNTP content at the start of the cell cycle (Figure 2.4D).

This change in the availability of dNTPs imposes a control on the cell-cycle duration. Specifically, the duration of the initiation and replication phases is inversely related to each other in single cells (Figure 2.4E), such that longer initiation times led to shorter replication times. This occurs because cells that require extra time to initiate replication also build up a large dNTP surplus, leading to faster replication. This interplay buffers against the high variability in the duration of replication initiation, giving rise to substantially less variability in the length of the cell cycle. The whole-cell model therefore presents a hypothesis of an emergent control of cell-cycle duration that is independent of genetic regulation.

2.2.6 Global Distribution of Energy

The model also provided an opportunity to develop a quantitative assessment of cellular energetics, which represents one of the most connected aspects of our model. To begin, we investigated the synthesis dynamics of the high-energy intermediates ATP, GTP, FAD(H_2), NAD(H), and NADP(H) and found that ATP and guanosine triphosphate (GTP) are synthesized at rates greater than 1,000-fold higher than the others (Figure 2.5A). Notably, the overall usage of ATP and GTP did not vary considerably in all but the very slowest of our simulations (Figure 2.5B), underscoring the role of metabolism in controlling the cell-cycle length. We then considered the processes that use ATP and GTP and found that usage is dominated by production of mRNA and protein (Figure 2.5C). We also found a large (44%) discrepancy between total energy usage and production (Figure 2.5D). Others have noted an uncoupling between catabolism and anabolism, attributing the difference to factors such as varying maintenance costs or energy spilling via futile cycles [48], and the model's prediction estimates the total energy cost of such uncoupling.

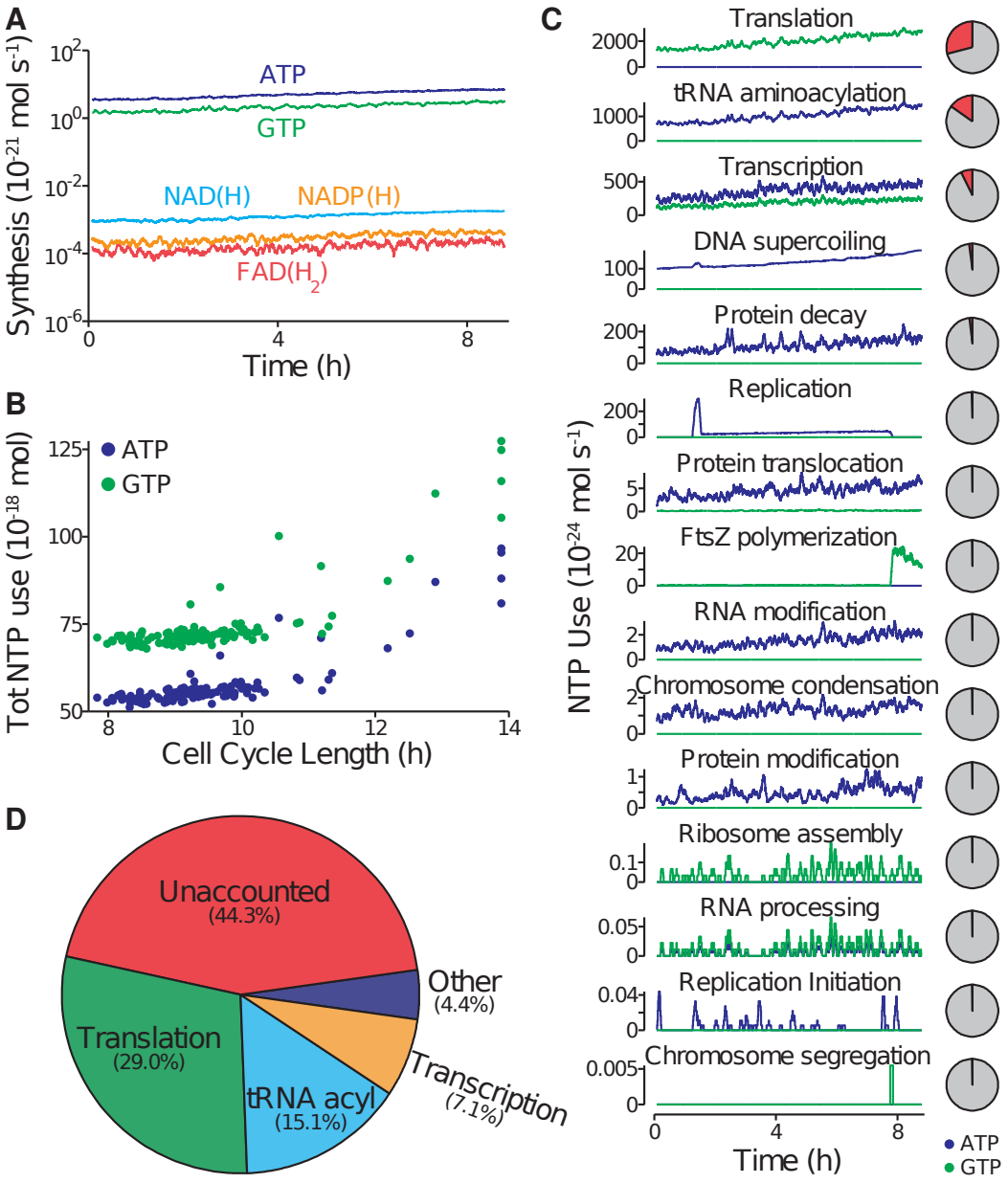

A
Synthesis (10^{-21} mol s^{-1})
ATP
GTP
NAD(H)
NADP(H)
$FAD(H_2)$
Time (h)
B
Tot NTP use (10^{-18} mol)
ATP
GTP
Cell Cycle Length (h)
C
NTP Use (10^{-24} mol s^{-1})
Translation
tRNA aminoacylation
Transcription
DNA supercoiling
Protein decay
Replication
Protein translocation
FtsZ polymerization
RNA modification
Chromosome condensation
Protein modification
Ribosome assembly
RNA processing
Replication Initiation
Chromosome segregation
Time (h)
ATP
GTP
D
Unaccounted (44.3%)
Other (4.4%)
Translation (29.0%)
tRNA acyl (15.1%)
Transcription (7.1%)

Figure 2.5: Model Provides a Global Analysis of the Use and Allocation of Energy.
(A) Intracellular concentrations of the energy carriers ATP, GTP, FAD(H_2), NAD(H), and NADP(H) of one in silico cell.
(B) Comparison of the cell-cycle length and total ATP and GTP usage of 128 in silico cells.
(C) ATP (blue) and GTP (green) usage of 15 cellular processes throughout the life cycle of one in silico cell. The pie charts at right denote the percentage of ATP and GTP usage (red) as a fraction of total usage.
(D) Average distribution of ATP and GTP usage among all modeled cellular processes in a population of 128 in silico cells. In total, the modeled processes account for only 44.3% of the amount of energy that has been experimentally observed to be produced during cellular growth.

2.2.7 Determining the Molecular Pathologies of Single-Gene Disruption Phenotypes

Having considered these above-described model predictions for the wild-type *M. genitalium* strain, we next performed in silico genome perturbations to gain insight into the genetic requirements of cellular life. We performed multiple simulations of each of the 525 possible single-gene disruption strains (over 3,000 total simulations) and found that 284 genes are essential to sustain *M. genitalium* growth and division and that 117 are nonessential. The model accounts for previously observed gene essentiality with 79% accuracy ($p < 10^{-7}$; [49]; Figure 2.6A).

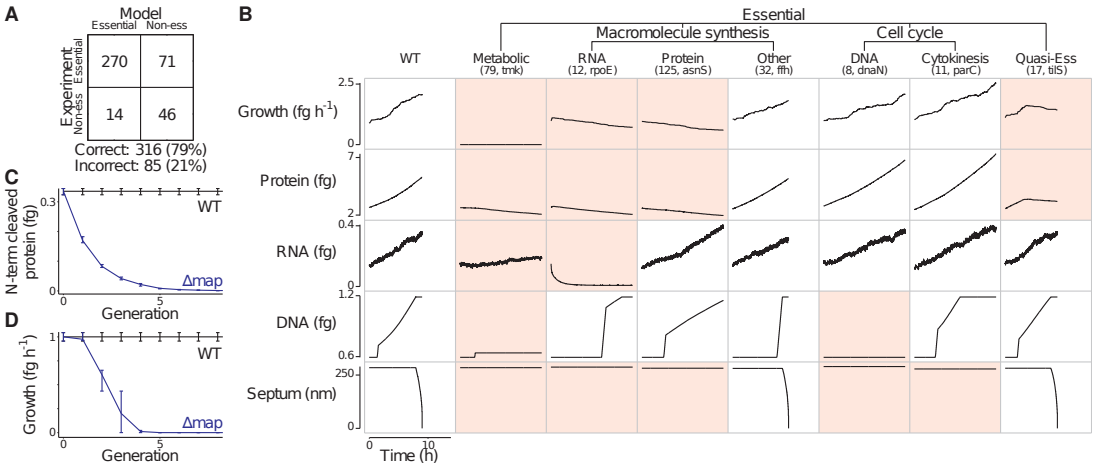


Figure 2.6: Model Identifies Common Molecular Pathologies Underlying Single-Gene Disruption Phenotypes.
(A) Comparison of predicted and observed [49] gene essentiality. Model predictions are based on at least five simulations of each single-gene disruption strain; see Data S1 for details.
(B) Single-gene disruption strains were grouped into phenotypic classes (columns) according to their capacity to grow, synthesize protein, RNA, and DNA, and divide (indicated by septum length). Each column depicts the temporal dynamics of one representative in silico cell of each essential disruption strain class. Disruption strains of nonessential genes are not shown. Dynamics significantly different from wild-type are highlighted in red. The identity of the representative cell and the number of disruption strains in each category are indicated in parenthesis.
(C and D) Degradation and dilution of N-terminal protein content (C) of methionine aminopeptidase (map, MG172) disrupted cells causes reduced growth (D). Blue and black lines indicate the map disruption and wild-type strains, respectively. Bars indicate SD.

In cases in which the model prediction agrees with the experimental outcome with respect to gene essentiality, we found that a deeper examination of the simulation can generate insight into why the gene product is required by the system. We examined the capacities of the 525 simulated gene disruption strains to produce major biomass components (RNA, DNA, protein, and lipid) and to divide. As shown in Figure 2.6B, the nonviable strains were unable to adequately perform one or more of these major functions. The most debilitating disruptions involved metabolic genes and resulted in the inability to produce any of the major cell mass components. The next most debilitating gene disruptions impacted the synthesis of a specific cell mass component, such as RNA or protein. Interestingly, in these cases, the model predicted an initial phase of near-normal growth followed by decreasing growth due to diminishing protein content. In some cases (Figure 2.6B, fifth column), the time required for the levels of specific proteins to fall to lethal levels was greater than one generation (Figures 2.6C and 2.6D). A third class of lethal gene disruptions impaired cell-cycle processes. For these, the model predicted normal growth rates and metabolism, but it also predicted incapacity to complete the cell cycle. The remaining lethal gene disruption strains grew so slowly compared to wild-type that they were considered nonviable (Figure 2.6B). We conclude that the model can be used to classify cellular phenotypes by their underlying molecular interactions.

2.2.8 Model-Driven Biological Discovery

Using computational modeling as a complement to an experimental program has previously been shown to facilitate biological discovery [17]. This is often accomplished by reconciling model predictions that are initially inconsistent with observations [18]. To test the utility of the whole-cell model in this context, we experimentally measured the growth rates of 12 single-gene disruption strains—ten of which were correctly predicted to be viable and two of which were incorrectly predicted to be nonviable—for comparison to our model's predictions (Figure 7A). We found that two-thirds of the predictions were consistent with the measured growth rates.

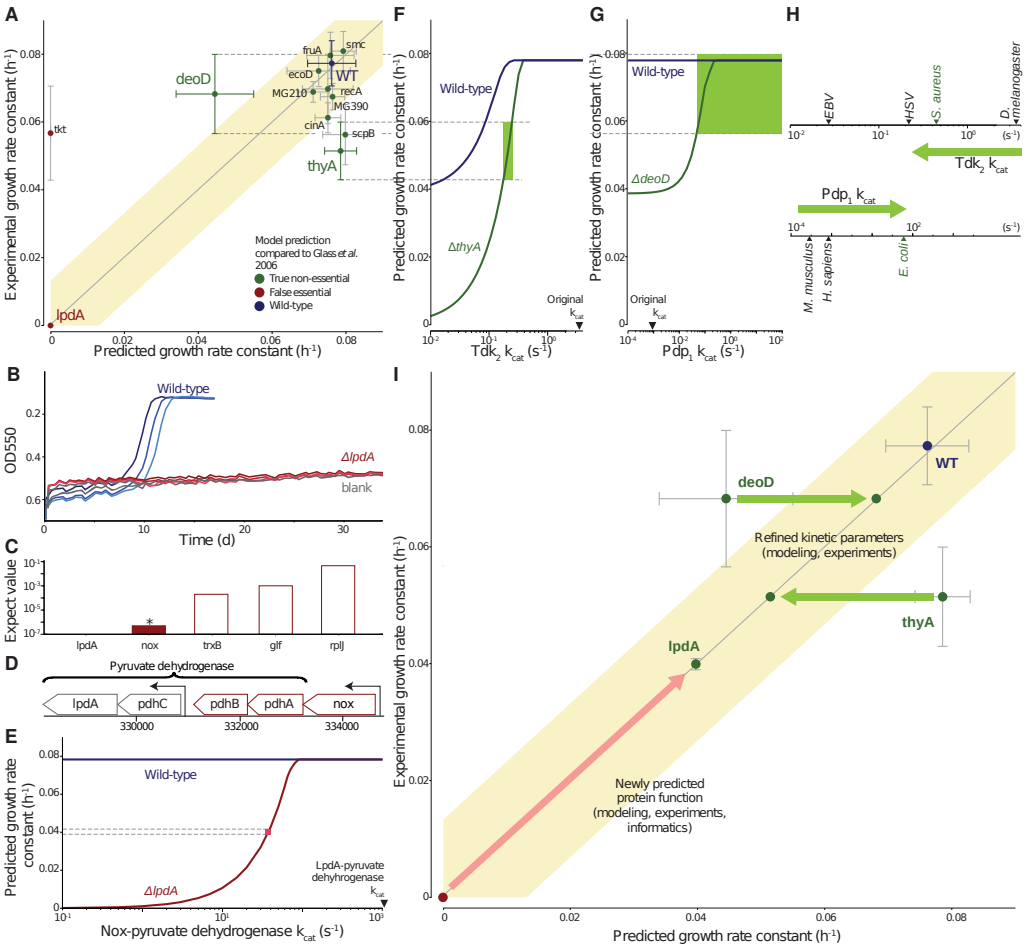

A
Experimental growth rate constant (h-1)
Predicted growth rate constant (h-1)
deoD
WT
thyA
lpdA
tkt
fruA
smc
ecoD
MG210
recA
MG390
cinA
scpB
Model prediction compared to Glass et al. 2006
True non-essential
False essential
Wild-type
B
OD550
Time (d)
Wild-type
ΔlpdA
blank
C
Expect value
lpdA
nox
trxB
glf
rplJ
D
Pyruvate dehydrogenase
lpdA
pdhC
pdhB
pdhA
nox
330000
332000
334000
E
Predicted growth rate constant (h-1)
Wild-type
ΔlpdA
LpdA-pyruvate dehyhrogenase kcat
Nox-pyruvate dehydrogenase kcat (s-1)
F
Predicted growth rate constant (h-1)
Wild-type
ΔthyA
Original kcat
Tdk2 kcat (s-1)
G
Predicted growth rate constant (h-1)
Wild-type
ΔdeoD
Original kcat
Pdp1 kcat (s-1)
H
EBV
HSV
S. aureus
D. melanogaster
Tdk2 kcat
Pdp1 kcat
M. musculus
H. sapiens
E. coli
I
Experimental growth rate constant (h-1)
Predicted growth rate constant (h-1)
deoD
WT
Refined kinetic parameters (modeling, experiments)
thyA
lpdA
Newly predicted protein function (modeling, experiments, informatics)

Figure 2.7: Quantitative Characterization of Selected Gene Disruption Strains Leads to Identification of Novel Gene Functions and Kinetic Parameters.
(A) Comparison of measured and predicted growth rates for wild-type and 12 single-gene disrupted strains. Model predictions that fall within the shaded region were considered consistent with experimental observations; the region has a width of four times the SD of the wild-type strain growth measurement. Horizontal and vertical bars indicate predicted and observed SD.
(B) Growth curves for the wild-type and *lpdA* gene disruption strains and blank, similar to Figure 2.2A.
(C) Expectation values determined by performing a pBLAST search of the *M. genitalium* genome with the LpdA sequence as a query. The asterisk and colored bar indicate a significant match ($E < 10^{-6}$).
(D) Detail of the *M. genitalium* genome. The pyruvate dehydrogenase complex genes are indicated by the top bracket, and transcription units identified in *M. pneumoniae* [21] are indicated by arrows. The transcription unit including *nox* is highlighted in color.
(E) Allowing Nox to partially replace LpdA in pyruvate dehydrogenase reconciles model predictions and experimental observations. The blue and red lines represent the predicted wild-type and Δ*lpdA* strain growth rates as a function of the Nox-pyruvate dehydrogenase k_{cat}. The pink box indicates the k_{cat} at which the model predictions are consistent with both the wild-type and Δ*lpdA* strain experimentally measured growth rates.
(F and G) Diagnosing the discrepancy between predictions and experiment for the *thyA* (F) and *deoD* (G) gene disruption strains. Some of the functionalities of ThyA and DeoD can be replaced by the enzymes Tdk and Pdp, respectively. The predicted growth rates of the wild-type and gene disruption strains depend on the k_{cat} of these enzymes. The green region highlights the range of k_{cat} values that are consistent with the measured growth rates of both the wild-type and gene disruption strain.
(H) Newly predicted k_{cat} values are similar to values that were measured in closely related organisms. Measured values of k_{cat} for Tdk (top) and Pdp (bottom) are shown; green arrow indicates the initial and revised k_{cat} values. The nearest *M. genitalium* relative is highlighted in green.
(I) Model-based biological discovery. Comparison of model predictions to experimental measurements identified gene disruption strains of particular interest, including the *lpdA*, *deoD*, and *thyA* disruption strains. Further investigation—using a combination of experiments, modeling, and/or informatics—led to new and more consistent measurements and predictions. Most importantly, the higher consistency reflected novel insights into *M. genitalium* biology. The arrows (red for *lpdA*, green for *deoD* and *thyA*) indicate the shift from lower to higher consistency between model and experiment, and each arrow is annotated with the new biological insight and the supporting evidence in parentheses. The overall graph format is the same as in Figure 2.7A. Horizontal and vertical bars indicate predicted and observed SD.

The most interesting of these comparisons concerned the *lpdA* disruption strain. The *lpdA* gene was originally determined to be nonessential [49]. Consequently, we initially classified the model's prediction as false (Figure 2.6A). However, we did not detect growth using our colorimetric assay (Figure 2.7B), which was a discrepancy that warranted further investigation. An alternative method to determine the doubling time yielded a value that was 40% lower than the wild-type (Table S1). Taken together, the data suggested that disrupting the *lpdA* gene had a severe but noncritical impact on cell growth.

In an effort to resolve the discrepancy between our model and the experimental measurements,

we determined the molecular pathology of the *lpdA* disruption strain. The *lpdA* gene product is part of the pyruvate dehydrogenase complex, which catalyzes the transfer of electrons to nicotinamide adenine dinucleotide (NAD) as a subset of the overall pyruvate dehydrogenase chemical reaction [50]. The viability of the *lpdA* disruption strain suggests that this reaction could be catalyzed by another enzyme with a lower catalytic efficiency.

Because previous studies have shown that many *M. genitalium* genes are multifunctional [51,52], we searched the genome for candidates encoding an alternative NAD electron transfer pathway. We found that the Nox sequence was far more similar to the LpdA sequence than any other gene product in the genome, with 61% coverage, 25% identity, and an expectation value of less than 10^6 (Figure 2.7C). Furthermore, the *nox* gene product, NADH oxidase, has been shown to oxidize NAD [53]. Moreover, the *nox* locus falls in a suboperon that contains two other pyruvate dehydrogenase genes and has been shown to be coexpressed with *pdhA* [21] (Figure 2.7D), strongly suggesting a functional relationship between the products of these two genes. Our model suggests that, to reproduce the observed growth rate in the absence of *lpdA*, the hypothetical Nox-dependent reaction would require a k_{cat} of ~50 s^{-1} (Figure 2.7E), which represents only ~5% of the maximum throughput of this enzyme. We therefore concluded that substrate promiscuity of Nox is likely to enable the *lpdA* disruption strain to survive.

Four gene disruption strains exhibited growth rates that were quantitatively different than those predicted by the model (Figure 2.7A); of these, we used the complete simulations for the *thyA* and *deoD* strains to determine the underlying pathology of the respective gene disruptions. The *thyA* gene product catalyzes thymidine monophosphate (dTMP) production and can be complemented by the *tdk* gene product. We therefore hypothesized that, by reducing the k_{cat} value for Tdk in the model, we would see a reduction in the growth rate of the *tdk* disruption strain. Reducing the Tdk k_{cat} in the model did indeed reduce the predicted growth rate of the *thyA* strain, but it also affected the wild-type growth rate (Figure 2.7F). Only a small range of the k_{cat} values both reduced the *thyA* strain growth rate to the experimentally observed levels and was also consistent with the wild-type growth rate.

In a similar case, purine nucleoside phosphorylase (DeoD) catalyzes the conversion of deoxyadenosine to adenine and D-ribose-1phosphate; these products can also be produced by the *pdp* gene product from deoxyuridine. We identified a Pdp k_{cat} range for which the wild-type and *deoD* gene disruption strains produce the same growth rate (Figure 2.7G).

Significantly, these newly predicted k_{cat} values are consistent with previously reported values. In the original model reconstruction, to least constrain the metabolic model, we conservatively set each of these k_{cat}s to the least restrictive value found during the reconstruction process. For Tdk and Pdp, these values corresponded to distantly related organisms; however, the newly predicted k_{cat} values are consistent with reports from more closely related species (Figure 2.7H).

In each of these three cases (*lpdA*, *deoD*, and *thyA*), identifying a discrepancy between model

predictions and experimental measurements led to further analysis, which resolved the discrepancy and also provided insight into *M. genitalium* biology (Figure 2.7I). These results support the assertion that large-scale modeling can be used to guide biological discovery [54, 55].

2.3 Discussion

We have developed a comprehensive whole-cell model that accounts for all of the annotated gene functions identified in *M. genitalium* and explains a variety of emergent behaviors in terms of molecular interactions. Our model accurately recapitulates a broad set of experimental data, provides insight into several biological processes for which experimental assessment is not readily feasible, and enables the rapid identification of gene functions as well as specific cellular parameters.

In contemplating these results, we make two observations based on comparing this work in whole-cell modeling with earlier work in whole-genome sequencing. First, similar to the first reports of the human genome sequence, the model presented here is a "first draft," and extensive effort is required before the model can be considered complete. Of course, much of this effort will be experimental (for example, further characterization of gene products), but the technical and modeling aspects of this study will also have to be expanded, updated, and improved as new knowledge comes to light.

Second, in whole-genome sequencing as well as in whole-cell modeling, *M. genitialium* was a focus of initial studies, primarily because of its small genome size. The goal of our modeling efforts, as well as that of early sequencing projects, was to develop the technology in a reduced system before proceeding to more complex organisms. However, *M. genitalium* presents many challenges with regard to experimental tractability. Resistance to most antibiotics, the lack of a chemically defined medium, and a cell size that requires advanced microscopy techniques for visualization all greatly limit the range of experimental techniques available to study this organism. As a result, much of the data used to build and validate the model were obtained from other organisms. Therefore, although the results we report suggest several experiments that could yield important insight with respect to *M. genitalium* function, comprehensive validation of our approach will require modeling more experimentally tractable organisms such as *E. coli*.

We are optimistic that whole-cell models will accelerate biological discovery and bioengineering by facilitating experimental design and interpretation. Moreover, these findings, in combination with the recent de novo synthesis of the *M. genitalium* chromosome and successful genome transplantation of *Mycoplasma* genomes to produce a synthetic cell [56–59], raise the exciting possibility of using whole-cell models to enable computer-aided rational design of novel microorganisms. Finally, we anticipate that the construction of whole-cell models and the iterative testing of them against experimental information will enable the scientific community to assess how well we understand integrated cellular systems.

2.4 Experimental Procedures

2.4.1 Reconstruction

The whole-cell model was based on a detailed reconstruction of *M. genitalium* that was developed from over 900 primary sources, reviews, books, and databases. First, we reconstructed the organization of the chromosome, including the locations of each gene, transcription unit, promoter, and protein-binding site. Second, we functionally annotated each gene, beginning with the Comprehensive Microbial Resource (CMR) annotation. Functional annotation was primarily based on homologs identified by bidirectional best BLAST. To fill gaps in the reconstructed organism and to maximize the scope of the model, we expanded and refined each gene's annotation using primary research articles and reviews (see Data S1 and Table S3). Third, we curated the structure of each gene product, including the posttranscriptional and posttranslational processing and modification of each RNA and protein and the subunit composition of each protein and ribonucleoprotein complex. After annotating each gene, we categorized the genes into 28 cellular processes. We curated the chemical reactions of each cellular process. The reconstruction was stored in an MySQL relational database. See Data S1 and Table S3 for further discussion of the reconstruction.

2.4.2 Cellular Process Submodels

Because biological systems are modular, cells can be modeled by the following: (1) dividing cells into functional processes; (2) independently modeling each process on a short timescale; and (3) integrating process submodels at longer timescales. We divided *M. genitalium* into the 28 functional processes illustrated in Figure 2.1 and modeled each process independently on a 1 s timescale using different mathematics and different experimental data. The submodels spanned six areas of cell biology: (1) transport and metabolism; (2) DNA replication and maintenance; (3) RNA synthesis and maturation; (4) protein synthesis and maturation; (5) cytokinesis; and (6) host interaction. Submodels were implemented as separate classes. See Data S1 for further discussion of each submodel.

2.4.3 Submodel Integration

We integrated the submodels in three steps. First, we structurally integrated the process submodels by linking their common inputs and outputs through 16 cell variables (shown in Figure 2.1), which together represent the complete configuration of the modeled cell: (1) metabolite, RNA, and protein copy numbers; (2) metabolic reaction fluxes; (3) nascent DNA, RNA, and protein polymers; (4) molecular machines; (5) cell mass, volume, and shape; (6) the external environment, including the host urogenital epithelium; and (7) time. Second, the common inputs to the submodels were computationally allocated at the beginning of each time step. Third, we refined the values of the submodel parameters to make the submodels mutually consistent. See Data S1 for further discussion.

2.4.4 Simulation Algorithm

The whole-cell model is simulated using an algorithm comparable to those used to numerically integrate ODEs. First, the cell variables are initialized. Second, the temporal evolution of the cell state is calculated on a 1 s timescale by repeatedly allocating the cell variables among the processes, executing each of the cellular process submodels, and updating the values of the cell variables. Finally, the simulation terminates when either the cell divides or the time reaches a predefined maximum value. See Data S1 for further discussion.

2.4.5 Single-Gene Disruptions

Single-gene disruptions were modeled by (1) initializing the cell variables, (2) deleting the in silico gene, and (3) calculating the temporal evolution of the cell state for the first generation postdisruption. We also calculated the mean growth rate of each single-gene disruption strain at successive generations postdisruption. See Data S1 for further discussion of the implementation of disruption strains and their computational analysis.

2.4.6 Computational Simulation and Analysis

We used the whole-cell model to simulate 192 wild-type cells and 3,011 single-gene deletants. All simulations were performed with MATLAB R2010b on a 128 core Linux cluster. The predicted dynamics of each cell were logged at each time point and subsequently analyzed using MATLAB. See Data S1 for further discussion.

2.4.7 Bacterial Culture

M. genitalium wild-type and mutant strains with single-gene disruptions by transposon insertion [49] were grown in *Spiroplasma* SP-4 culture media at 37°C and 5% CO_2. Growth was detected using the phenol red pH indicator. Cells were harvested for quantitative growth measurement at pH 6.3-6.7. See Data S1 for more information about media and culture conditions.

2.4.8 Colorimetric Assay to Measure Cell Growth

To measure the growth rates of the wild-type and mutant strains, cells were collected from 10 cm plate cultures at pH 6.3-6.7, resuspended in 3 ml of fetal bovine serum (FBS), and serial filtered through 1.2, 0.8, 0.45, and 0.2 μm polyethersulfone filters to sterilize and separate individual cells. Cells were then plated at 5-, 25-, and 125-fold serial dilutions in triplicate on a 96-well plate and incubated at 37°C and 5% CO2. Six wells per plate were filled with blank SP-4 phenol red media as a negative control. Optical density readings were taken twice a day at 550 nm to measure the decrease in phenol red color as pH decreased. Growth rate constants were calculated from the additional time

required for consecutive dilutions to reach the same OD550 value and were averaged over two to three independent sets of three replicates. See Data S1 for further description of these calculations. We used a heteroscedastic two-sample two-tailed t test to determine whether the doubling time of each single-gene disruption strain differed significantly from that of the wild-type. The growth rates of several slow-growing strains were also measured by DNA quantification using a modified version of the procedure described in [49]. See Data S1 for further discussion.

2.4.9 Source Code

The model source code, training data, and results are freely available at SimTK (`https://simtk.org/home/wholecell`).

2.5 Acknowledgments

We thank R. Altman, S. Brenner, Z. Bryant, J. Ferrell, K. Huang, B. Palsson, S. Quake, L. Serrano, J. Swartz, E. Yus, and the Covert Lab for numerous enlightening discussions on bacterial physiology and computational modeling; T. Vora for critical reading of the manuscript; and M. O'Reilly and J. Maynard for graphical design assistance. This work was supported by an NIH Director's Pioneer Award (1DP1OD006413) and a Hellman Faculty Scholarship to M.W.C.; NSF and Bio-X Graduate Student Fellowships to J.C.S.; NDSEG, NSF, and Stanford Graduate Student Fellowships to J.R.K.; a Benchmark Stanford Graduate Fellowship to D.N.M.; and a U.S. Department of Energy Cooperative Agreement (DE-FC02-02ER63453) to the J. Craig Venter Institute.

Chapter 3

Challenges in whole-cell modeling

Abstract

Integrated whole-cell modeling is poised to make a dramatic impact on molecular and systems biology, bioengineering, and medicine — once certain obstacles are overcome. From our group's experience building a whole-cell model of *Mycoplasma genitalium*, we identified several significant challenges to building models of more complex cells. Here we review and discuss these challenges in seven areas: (1) experimental interrogation, (2) data curation, (3) model building and integration, (4) accelerated computation, (5) analysis and visualization, (6) model validation, and (7) collaboration and community development. Surmounting these challenges will require the cooperation of an interdisciplinary group of researchers to create increasingly sophisticated whole-cell models and make data, models, and simulations more accessible to the wider community.

3.1 Introduction

Predictive and comprehensive models of cellular physiology are critical to understanding and engineering biological systems. Such whole-cell models have the potential to guide experiments in molecular biology, enable computer-aided design and simulation in synthetic biology, and inform personalized treatment in medicine. Constructing and validating models with sufficient scope, detail, and predictive power, for a variety of cells, will be a massive undertaking.

Beginning in the late 1970s [60], researchers began modeling cell physiology, primarily using ordinary differential equation (ODE) approaches, creating increasingly detailed models over the next three decades [28, 29, 61]. Later, other groups introduced frameworks that generally require fewer

Chapter reproduced from: DN Macklin*, NA Ruggero*, MW Covert. "The future of whole-cell modeling" *Current Opinion in Biotechnology.* 2014. 28: 111-115

parameters than ODE systems including constraint-based [62, 63] and Boolean methods [30]. Combining these approaches for their respective benefits, our group developed a hybrid methodology: we modeled individual biological processes, each with its own mathematical representation, and merged their outputs to compute the overall state of the cell [35]. Using this approach, we simulated the life cycle of individual *Mycoplasma genitalium* cells, accounting for every molecule and representing the function of every annotated gene [7].

Several unforeseen obstacles arose during the modeling process, which should inform any future whole-cell modeling efforts. Specifically, modeling larger cells and more complex physiology presents challenges in (1) experimental interrogation, (2) data curation, (3) model building and integration, (4) accelerated computation, (5) analysis and visualization, (6) model validation, and (7) collaboration and community development, shown in Figure 3.1. No single research group can simultaneously innovate in all these areas. Rather, a broader community will need to coalesce to tackle these problems. We address this article to that community, discussing the challenges and highlighting notable progress in each area.

3.2 Experimental interrogation

Parameterizing and validating the *M. genitalium* whole-cell model was particularly challenging due to a lack of organism-specific data. Many values were estimated from measurements made in other species. Future efforts will ideally simulate well-characterized organisms, for example *Mycoplasma pneumoniae* [21–23, 64], *Escherichia coli* [65], and *Saccharomyces cerevisiae* [66, 67]. Because whole-cell models simulate the life-cycle of an individual cell, one would ideally use spatially-resolved, genome-scale, dynamic, single-cell measurements to parameterize and validate the models. However, many published measurements are static ensemble averages representing a population mean at a single time point [68–72]. This lack of data ultimately presents the modeler with a dilemma: either infer missing data, or create a less detailed model of a particular phenomenon. To create the *M. genitalium* model, we necessarily inferred some degree of dynamical behavior. Faced with a similar problem, others have found ways to incorporate static spatial data in their efforts to create dynamic 3D cell-scale simulations [73]. Promising work in advancing single-cell measurement techniques and technologies [41, 74–76] will ultimately drive more detailed and accurate modeling. To make these efforts even more impactful and useful, the experimental community could work to establish standardized conditions and place a higher value on consistent, reproducible measurements.

3.3 Data curation

No single technology exists which can chronically measure and record the entire state of a single cell. As a result, heterogeneous data sets must be combined and unified for model parameterization

and validation. While efforts such as the BioCyc databases have sought to unify genomic and metabolic pathway information [77], separate databases contain functional parameters such as kinetic rates [78, 79] and expression levels [80]. To compile the data required to build the *M. genitalium* model, which we share via WholeCellKB [81], we had to download and synthesize parameters from these and other databases as well as the primary literature. For larger and more complex organisms, the sheer magnitude of data to collect, and the number of discrepancies to resolve, will present significant hurdles to parameterizing a model.

Since parameterization data increases with organism complexity and known physiology, a part-time manual curation effort will not be tenable. Researchers will need to exploit advances in natural language processing to extract information from the primary literature en masse [82], or outsource part of the effort. Formally interacting with domain experts, as has been done in the flux-balance analysis community [83], will be critical to assembling consensus data sets. Ultimately, a combination of computer-automated and human-augmented approaches will be necessary to gather and assemble the data for larger whole-cell models.

A collection of centralized, organism-specific databases similar to WholeCellKB will be required for subsequent whole-cell modeling efforts. In the best case, researchers would go beyond including raw data for each figure in a paper [84] and would deposit their results to the appropriate database in a machine-readable format. Dedicated curators would update the database schemas to incorporate new types of information as needed. In addition, the databases would alert the community to significant discrepancies between parameters and flag them as critical issues to resolve. By providing these capabilities, the databases would link experimental evidence to whole-cell models.

3.4 Model building and integration

Comprehensively representing cell physiology in a single computational model requires integrating diverse phenomena over multiple length and time scales, handling the different levels of understanding associated with each phenomenon, and representing the state of the cell in sufficient detail. Our lab's approach to meeting these requirements relies on the notion of biological modularity [85], allowing us to divide the cell into independent state variables (e.g., representing metabolite counts or the functional state of macromolecules) and cellular processes (e.g., transcription, metabolism) [7]. We create sub-models of each cellular process using a mathematical representation informed by available data and current understanding. We assume that, over a small time step, each sub-model can independently execute and update a subset of the cell state variables. To meaningfully combine sub-models in this fashion, we must (1) establish and link common variables, and (2) ensure that the combined behavior is consistent with physical laws and biological phenotypes.

To avoid duplicating work, it is desirable to incorporate published models of particular biological processes into a whole-cell modeling framework. This often requires that the published models be

modified to use the common whole-cell state variables, which may, for example, involve changing the published model's quantities from concentrations to counts, or linking its variables to the appropriate cell compartment in the whole-cell framework. Establishing mathematical methods for properly converting a spatially-resolved variable, used in a detailed sub-model, to a bulk quantity, or even to a Boolean value, used in a less-detailed sub-model, would ease the data interconversion between sub-models. Numerical analysis of these methods could be performed to examine factors which affect stability and accuracy of the simulations, and to quantify numerical uncertainty in model predictions.

With a collection of sub-models that properly interface with cell state variables, it must further be enforced that their aggregate behavior does not violate physical laws. For example, the aggregate action of multiple sub-models should not result in the consumption of more resources than are present. To avoid this situation, we developed a method to allocate cell state variables to biological processes proportional to each process's need. In the future, this top-down approach could be replaced with one more grounded in physical laws.

Furthermore, the aggregate behavior of a collection of sub-models should be consistent with biological phenotypes. For instance, the small molecule, RNA, protein, and DNA mass fractions, must approximately double over the exponentially-growing cell's life cycle. This requirement constrains certain sub-model parameters so that metabolism, for example, produces nucleotides and amino acids in the proportions needed by replication, transcription, and translation. The *M. genitalium* model performed this adjustment prior to simulation; however, new methods must be developed to update these loosely-coupled parameters during simulation. Importantly, this will enable proper incorporation of regulatory sub-models [86,87] which modify the nucleotide and amino acid demands as the RNA and protein expression profiles change in response to perturbations.

3.5 Accelerated computation

Computational simulation is a powerful scientific and engineering tool because it enables rapid and inexpensive exploration of alternative scenarios and hypotheses, as well as design optimization. Such investigations, however, hinge on efficient computation in order to explore a sufficiently large portion of parameter space. The whole-cell simulations of *M. genitalium*, which each took approximately ten hours to run, do not meet this criteria. We can extrapolate that, without innovation in this area, simulations of more complex organisms will take considerably longer to execute. High-performance parallelized computing technologies, such as the Compute Unified Device Architecture (CUDA) [88] or Message Passing Interface (MPI) [89], or even custom hardware platforms [90], in the spirit of Anton [91] or Neurogrid [92], should be adapted and investigated for their abilities to speed-up the execution of whole-cell simulations.

3.6 Data analysis and visualization

Raw simulation data, like raw experimental data, typically requires extensive analysis to be adequately understood and communicated. Techniques from machine learning and dynamical systems analysis could be used to explore and interrogate simulated single-cell phenotypes. These analyses could suggest novel hypotheses about the dynamics of single cells that wouldn't emerge from static, population-averaged data.

To complement analysis technologies, advances are needed in large-data visualization. While our group released WholeCellViz to expose a portion of the *M. genitalium* data set [9], going forward more sophisticated tools must be developed, particularly for exploration, rather than just communication, of large data sets. This requires the development of not only new visual motifs for biological data, but also improvements in data processing and retrieval to enable interactive interfaces for manipulating entire data sets. Existing tools [93] offer these interactive exploratory interfaces, but generally operate on smaller data sets [94]. Fortunately, these problems are recognized as pressing issues by the visualization community [95]. Preliminary work has begun to explore new visual motifs for biological data [96], [97], [98], and the high-performance computing community is supporting new techniques to improve data retrieval [99].

3.7 Model validation

Model predictions and experimental validation are linked by an iterative process in which each provides feedback on the other [54]. For the initial validation of the *M. genitalium* whole-cell model, we simply compared model predictions to as many heterogeneous data sets as possible that were withheld from model reconstruction. We have also used the model to predict the outcome of experiments which are performed subsequently [10]. Nevertheless, the validation process for the *M. genitalium* model has been guided more by intuition than by a systematic methodology. Ideally, a quantitative metric would exist to specify how much of a model has been validated and would point to data sets needed to improve the coverage of validation. More subtly, methods should be developed which can differentiate novel predictions (e.g., gene essentiality in the *M. genitalium* model) from outputs arising directly from parameter fitting (e.g., biomass composition in the *M. genitalium* model). These innovations would support more widespread model adoption by building trust in the predictions.

3.8 Collaboration and community development

Whole-cell models of more complex microbes and cell types will likely become community endeavors, particularly as the models grow in scope and detail. To facilitate interaction with the broader community, we released the entire code base for the *M. genitalium* whole-cell model under the

MIT license [100], permitting open development and re-use. Going forward, we must engage the broader community in contributing to whole-cell model development. The interface between cell state variables and process sub-models must be explicitly documented in detail to lower the barrier to contribution. Furthermore, a formal plug-in system must be developed to simplify the incorporation of alternate sub-models for a particular process. At the project-management level, metrics to quantify contribution and guidelines for authorship need to be proposed and ratified. At the community level, workshops, conferences, and competitions [101] specifically focusing on whole-cell modeling need to be organized to engage the breadth of contributing researchers.

3.9 Conclusion

The need to address the aforementioned challenges provides a wealth of opportunities for interdisciplinary contribution by experimentalists, modelers, computer scientists, statisticians, bioinformaticians, and software engineers. We hope a community will form where scientists and engineers from diverse backgrounds can collaborate and innovate together to overcome these obstacles.

Whole-cell modeling can help researchers prioritize experiments by identifying knowledge gaps and by highlighting measurement discrepancies [10]. Additionally, the comprehensive scope of a whole-cell model enables predictions of the pleiotropic effects of perturbation [102], critical to the future of synthetic biology and personalized medicine. Addressing the issues discussed here will enable whole-cell modeling to realize its potential, and in the process make an impact on model-guided science, synthetic biology, and medicine.

3.10 Acknowledgments

We thank Elsa Birch, Ellen Casavant, Shrivats Iyer, and Jonathan Karr for their critical feedback of this manuscript, as well as members of the Covert Lab for enlightening discussions on the topic. This work was supported by an NIH Director's Pioneer Award (5DP1LM011510-05), an Allen Distinguished Investigator Award, and an award from the Stanford Bio-X Corporate Forum and Agilent to MWC, a Benchmark Stanford Graduate Fellowship and DOE CSGF Fellowship (DE-FG02-97ER25308) to DNM, and an NSF Graduate Research Fellowship to NAR.

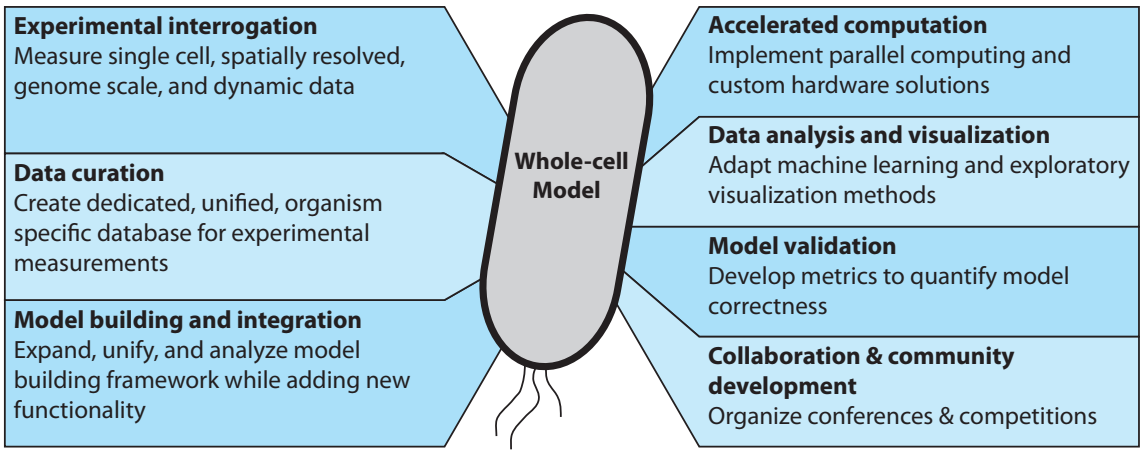


Figure 3.1: The interdisciplinary challenges faced by future whole-cell modeling efforts. A community of scientists and engineers will need to innovate together to surmount these challenges.

Chapter 4

Engineering improvements in *E. coli* whole-cell simulations

4.1 Background

The gene-complete whole-cell simulations of *M. genitalium* are a truly amazing feat of science and engineering. The journal Cell recognized this work as one of its top 40 landmark publications of the last 4 decades [103]. Nonetheless, the simulations, and the code backing them, have serious shortcomings which needed to be addressed in order to scale to an *E. coli* whole-cell model:

1. **Run-time**. Simulations of a single life cycle of *M. genitalium* take 10-14 hours to run. While *E. coli*'s life cycle is shorter by approximately a factor of 10, it has 50 times more molecules than *M. genitalium*. If we conservatively assume that this leads to only 5 times longer run-time ($1/10 \cdot 50 = 5$), it would take the better part of a week to run *E. coli* simulations. Not only does the test-debug cycle begin to become unsustainable at this point, but this practically precludes the ability to run multiple generations—something that would be critical when dynamically simulating environmental shifts in *E. coli*.

2. **Inability to simulate multiple generations**. Jonathan Karr attempted to simulate multiple generations of an *M. genitalium* life cycle, but was unable to reliably achieve growth and division of daughter cells. As mentioned above, this would be critical when dynamically simulating environmental shifts.

3. **Long onboarding process**. I worked with Jonathan Karr for over 6 months when analyzing the results of the *M. genitalium* simulations. Even with that experience, if I wanted to make a significant modification to the model, I needed to consult Jonathan to ensure that my change would have the intended effect. John Mason had a similar experience when trying to

simulate the effects of antibiotics with the model. The code was terse and lacked comments and documentation. To be fair, I think it is unreasonable to have expected Jonathan and Jayodita, neither of whom are software engineers by training, to write extendable code while also pioneering the field of whole-cell modeling. We recognized, though, that a gene-complete *E. coli* model would span multiple generations of graduate students, and so a better code base and onboarding process would be essential.

The remainder of this chapter discusses how we addressed each of these issues.

4.2 Simulation run-time

There are three main factors that we used to shorten simulation run time for *E. coli* simulations: (1) improving file input/output (I/O), (2) writing "inner loops" in C so that they would be compiled to a binary executable, and (3) caching data across time steps. We discuss each of these below

4.2.1 Improving File I/O

Jonathan Karr noticed that if he ran *M. genitalium* simulations without saving the results to disk, they ran approximately 30% faster. Thus, when building *E. coli* simulations, we knew file I/O would have to be improved. We achieve faster file I/O by taking a three-pronged approach:

1. We are more selective in what we save to disk—if we're reasonably confident that a piece of data will not be used in downstream analysis, we don't save it. If it later turns out that we do want a piece of data in downstream analysis, we can re-run simulations while saving that data. Because we have achieved considerably faster run-time than the *M. genitalium* simulations (spoiler alert!), we deemed this a reasonable approach.

2. *M. genitalium* simulations compressed data while saving it to disk, even before it was analyzed. We save our raw data matrices straight to disk and only compress the data after our downstream analysis is complete.

3. We run our *E. coli* simulations on Stanford's Sherlock cluster which has a high-performance parallel file system. We ran our *M. genitalium* simulations on our lab cluster which lacked this capability—the Sherlock cluster did not exist at that time.

4.2.2 Inner loops

Inner loops are any piece of code which gets called many (e.g., hundreds, thousands, or more) times per time step. They are the workhorses of the simulation. For example, we have routines that elongate nascent polypeptides one amino acid at a time as they are individually produced by one of

E. coli's 10,000+ ribosomes (the actual number of elongating ribosomes varies depending on environmental condition). These routines, for example, have `for` loops that iterate over each elongating ribosome at each of the approximately 16 amino acids they can polymerize each time step. By profiling our code, we have observed that these inner loops are some of the most expensive operations in the simulations. Additionally, it is well-known that `for` loops are inefficient in interpreted languages such as Python. As a result, we have written these routines in Cython, which gets translated to C and compiled to a binary.

4.2.3 Caching

Aside from the routines that elongate nascent polypeptides, one of the most expensive routines in the simulations is solving a linear program for the metabolism sub-model. In the *M. genitalium* simulations (and early versions of *E. coli* simulations), we instantiated an entirely new optimization problem at every time step. Now we cache the optimization problem from the previous time step and update our constraints before passing this information to the numerical solver. This enables the solver to "warm start"—to begin its search for the new optimal solution from the previous optimal solution, which, since the simulation doesn't generally change significantly from time step to time step, shouldn't be too far away.

4.2.4 Results

By implementing the three aforementioned factors, **our *E. coli* simulations**, which we expected could take the better part of a week to complete, **run in 10-15 minutes**. We could further improve the run-time by porting the entire code base to C/C++ and parallelizing the execution of the sub-models (they are assumed to be independent over short time scales, so there's no reason this can't be done). We could also try implementing the simulations on GPUs. All of these options require a very significant investment of effort, and they would also make the simulations more difficult to debug (parallel programs have whole classes of bugs that do not exist in serial programs). Given these trade-offs, we opted to focus our attention on other issues.

4.3 Multiple Generations

In order to see the effect of an environmental shift on *E. coli*, we need to be able to simulate multiple generations. As mentioned earlier, *M. genitalium* simulations were incapable of growing reliably over multiple generations. My labmate Nick Ruggero was able to solve this problem with assistance from postdoc Yu Tanouchi. They realized that the problem with the *M. genitalium* simulations was in the division criteria—cells would divide when their mass had doubled, regardless of how large or small the cell was initially. This would lead to a divergence in the mass of the population and individual

cells would lack critical resources for growth. Nick changed the division criteria so that cells would divide when they added the mass of the average cell to their initial mass. Thus, over time, even in the face of noise, cell mass would regress toward the mean.

4.4 Onboarding process

A large part of the difficulty in modifying the *M. genitalium* simulations is that the code was not written in an extendable manner. Jonathan Karr did write a simplified simulation framework, which I ported to Python and used as the basis for early *E. coli* simulations. This was still cumbersome to work with, particularly indexing into the massive array that represents the state of the cell. Ultimately, my labmate John Mason re-factored the code and developed an API using object-oriented principles that made writing sub-model code much more manageable. After this was completed, working with one of our rotation students Kalli Kappel, we developed a written tutorial describing how to run simulations and make modifications to sub-models. Multiple rotation students have been able to make meaningful contributions to the project within a few weeks of starting, even if they have limited programming experience.

4.5 Conclusion

In summary, we have addressed 3 major limitations present in *M. genitalium* simulations: (1) we have shortened simulation run time from 10-14 hours to 10-15 minutes, (2) we now have the ability to simulate multiple generations, and (3) we have shortened the onboarding time from many months to a few weeks.

Chapter 5

A quantitative genome-scale model of transcription in *E. coli*

5.1 Background

To express genes encoded in DNA, RNA Polymerase produces strands of RNA, often referred to as transcripts, complementary in sequence to their respective DNA templates. These transcripts are either themselves functional (e.g., rRNA, tRNA), or serve as an intermediary messenger in the production of protein (e.g., mRNA). While the precise number varies depending on the environmental condition, a good rule of thumb is that approximately 20% of the dry mass of a bacterium is RNA—the vast majority of that mass being rRNA, a critical component of the venerable ribosome. Table **x** shows the breakdown of our reconstructed RNA mass fraction at 3 different growth rates, as well as data reported by Neidhardt [104] for comparison. In whole-cell simulations, we must ensure that the dry mass fraction of RNA remains roughly constant over multiple generations and we must ensure that each gene is expressed at levels concordant with experimental measurements.

To accomplish these two tasks, in the *M. genitalium* model (presented in Chapter 2), the time series dynamics of counts of an RNA molecule r are initially considered in the form of an ordinary differential equation model:

$$\frac{dr}{dt} = k_p - \frac{\ln 2}{h} r \tag{5.1}$$

where k_p is a production term and h is the measured half-life for a first-order decay term. More generally, in an N-gene system, the expression of the j'th gene can be written as:

$$\frac{dr_j}{dt} = k_{p,j} - \frac{\ln 2}{h_j} r_j \tag{5.2}$$

When a cell is growing exponentially with doubling time τ_d, we additionally know that:

$$\frac{dr_j}{dt} = \frac{\ln 2}{\tau_d} r_j \tag{5.3}$$

Combining these two equations for dr_j/dt, we can solve for the production term $k_{p,j}$:

$$k_{p,j} = \left(\frac{\ln 2}{h_j} + \frac{\ln 2}{\tau_d} \right) r_j \tag{5.4}$$

Note that because we have h_j from half-life data, τ_d from the measured growth rate, and r_j from gene expression data, we can compute an explicit value for $k_{p,j}$. We then create a vector of gene synthesis probabilities v_{synth} by normalizing the $k_{p,j}$ values:

$$v_{\text{synth}} = \frac{1}{\sum_{i=1}^{N} k_{p,i}} \begin{bmatrix} k_{p,1} \\ \vdots \\ k_{p,N} \end{bmatrix} \tag{5.5}$$

In the transcription process in the *M. genitalium* whole-cell simulations, each initiating RNA Polymerase samples from a categorical distribution parameterized by v_{synth} to select a promoter site. Then the RNA Polymerases elongate maximally at a rate of 50 nt/s or until nucleotide pools are depleted. RNA molecules are each individually degraded by sampling from a Poisson distribution parameterized by half-life data and the existing counts of an RNA species.

While this served as a good starting point for an *E. coli* model, there is a glaring difference between *M. genitalium* physiology and *E. coli* physiology with respect to transcription—in *E. coli*, transcription is heavily regulated by DNA-binding proteins known as transcription factors (TFs). In fact, the expression of more than 400 of *E. coli*'s genes are regulated by transcription factor-mediated interactions. Thus, we deemed it critical to include this aspect of physiology and to undertake the accompanying experimental and modeling efforts.

We decided that the first step in modeling *E. coli* transcription and transcriptional regulation would be to obtain high-quality expression data (measured using RNA-seq) of the K-12 MG1655 strain in M9 minimal media and to treat this as a baseline condition. Then, to leverage more than a decade's worth of published microarray experiments (as well as a handful of RNA-seq experiments), we would compute fold-changes in TF-target gene expression between this baseline condition and perturbed conditions that would modulate TF activity. Combining this fold-change data with the baseline data would yield a complete expression profile for each condition. From these expression profiles, as well as data on transcription factor activity, we planned to infer RNA Polymerase recruitment strengths for each transcription factor. Figure 5.1 illustrates this workflow which we describe in detail in the following sections. The net result of this effort is an integrated, quantitative model of transcriptional regulation that modulates the expression of 355 genes via the activity of 22 transcription factors.

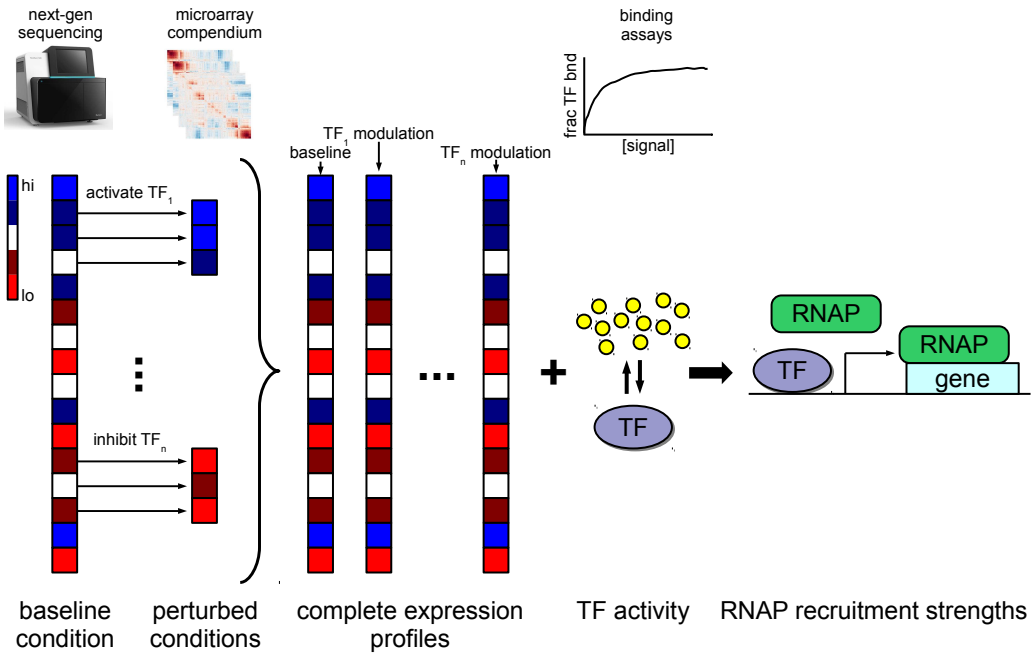


Figure 5.1: Workflow for modeling transcriptional regulation. Beginning with expression data in a baseline condition measured using RNA-seq, compute expression profiles for TF-perturbed conditions using fold-changes obtained from a microarray compendium. Then, using affinity data from binding assays, calculate transcription factor activity. Finally, infer the RNA polymerase recruitment strength of the transcription factor when it is DNA-bound.

5.2 Experimental measurements

To establish our baseline gene expression profile, we wanted to utilize the highest fidelity measurement at our disposal that wasn't cost prohibitive. Thus, we investigated trade-offs between published microarray data and RNA sequencing (RNA-seq) data. While exploring raw microarray data sets, I uncovered a couple of troubling phenomena: (1) reported expression values could be negative (presumably due to background subtraction algorithms used in image analysis), and (2) when comparing expression values from a wild-type condition and a gene deletion condition, the reported expression value in the gene deletion condition could be higher than in the wild-type condition. While we expect there to be some noise in any high-throughput measurement technology, in light of these findings and reports that RNA-seq has higher dynamic range [69]—in addition to the fact that you cannot have a negative number of reads for a gene—we opted to utilize RNA-seq

data. There was one problem: at that time (the summer of 2014), on the Sequence Read Archive, there was hardly any data for *E. coli*, and the data that did exist did not include wild-type K-12 MG1655 in a minimal medium. Thus, we decided to perform the measurements described here and outlined in Figure 5.2.

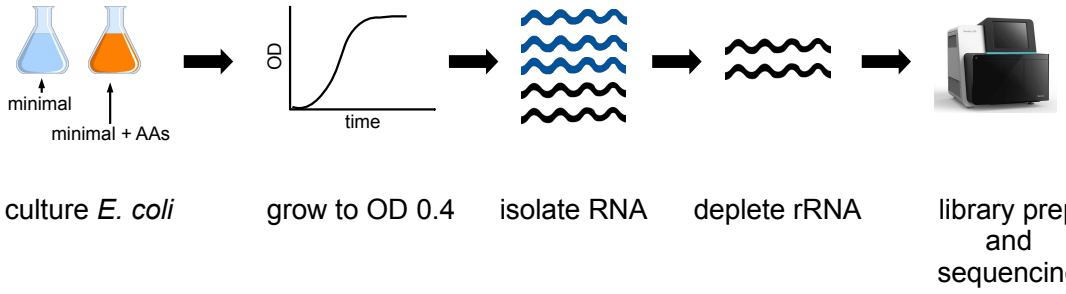


Figure 5.2: Experimental workflow to obtain gene expression in two conditions using next-generation sequencing.

We took overnight cultures of *E. coli* K-12 MG1655 and back-diluted them 1:100 into M9 minimal media + 0.4% glucose and M9 minimal media + 0.4% glucose supplemented with amino acids (designated "+AA"). Table 5.1 lists the amino acid concentrations in the media supplement (they are the same concentrations used in EZ-RDM [105]). The 10 mL cultures were placed in 125 mL flasks and placed on a shaker in the 37°C warm room. When cultures reached an optical density (OD) of 0.4, we isolated RNA using a Qiagen RNeasy Protect Bacteria Mini Kit and later depleted rRNA using an Epicentre Ribo-Zero rRNA Removal Kit. We collected 3 biological replicates, obtained on 3 separate days, of *E. coli* RNA from each condition. RNA samples were submitted to the Stanford Functional Genomics Facility for library preparation and sequencing.

Sequencing was performed on an Illumina NextSeq 500. We obtained approximately 20 million 75 bp paired-end reads per sample. We used bbmap [106] for preprocessing to trim reads and remove reads that map to common contaminants. We also removed reads that mapped to non-coding RNA (since rRNA should have been depleted). We tried a number of packages to process the remaining reads: RSEM [107], EDGE-Pro [108], Seal [106], and Cufflinks [109]. Their results are all highly correlated (with a Pearson r of at least 0.9 for each pairwise comparison). We ultimately opted to use the results from RSEM because the provided transcript-per-million (TPM) normalization was recommended by an expert in the field [110] and was easily adaptable to our modeling efforts. Because the different packages returned such similar results, we wouldn't expect to see significant differences in our simulations had we chosen differently.

Note: While I performed the aforementioned experiments and data analysis, the experiments would not have been possible without the guidance provided by Mialy DeFelice.

Component	Concentration
L-Alanine	0.8 mM
L-Arginine	5.2 mM
L-Asparigine	0.4 mM
L-Aspartic Acid	0.4 mM
L-Glutamic Acid	0.66 mM
L-Glutamine	0.6 mM
L-Glycine	0.8 mM
L-Histidine	0.2 mM
L-Isoleucine	0.4 mM
L-Proline	0.4 mM
L-Serine	10 mM
L-Threonine	0.4 mM
L-Tryptophan	0.1 mM
L-Valine	0.6 mM
L-Leucine	0.8 mM
L-Lysine	0.4 mM
L-Methionine	0.2 mM
L-Phenylalinine	0.4 mM
L-Cysteine	0.1 mM
L-Tyrosine	0.2 mM

Table 5.1: Concentrations of amino acids used in media supplement.

5.3 Fold change data

Note: This work was performed primarily by Javier Carrera, PhD.

We obtained our fold-change data from the EcoMAC microarray compendium [111] which contains over 2200 microarrays, each of which measures the expression of approximately 4500 genes. **Figure x** outlines the steps we took to process and filter the data into usable fold-changes. The first filter we applied to the data set was to restrict it to the MG1655, BW25113, and W3110 strains of *E. coli*. The next filter we applied was to remove studies that lacked a targeted stimulus (e.g., one study examined the effects of a microgravity environment on *E. coli*'s gene expression, another examined the effects of electromagnetic radiation on *E. coli*'s gene expression–both of those were removed from further consideration). The final filter we applied was to make sure that the direction of fold-changes were consistent with the network topology reported in RegulonDB [112].

5.4 Model implementation

There are two aspects to modeling transcriptional regulation:

1. Modeling the activation or inhibition of a transcription factor (e.g., by a ligand)

2. Given an active transcription factor, modeling its effect on RNA Polymerase recruitment to a

promoter site.

We address these topics sequentially below.

5.4.1 Modeling transcription factor activation

We consider three classes of transcription factors based on their mechanism of activation:

1. **One-component systems**: transcription factors that are directly activated or inhibited by a small molecule ligand. Examples of this class include the repressor TrpR which binds tryptophan, and the inducer AraC which binds arabinose.

2. **Two-component systems**: transcription factors that are paired with a separate sensing protein that responds to an environmental stimulus (these are simple analogs to the vast, complicated signaling networks that exist in eukaryotic cells). The sensing protein phosphorylates the cognate transcription factor in a condition-dependent fashion. Examples include ArcA which is phosphorylated by its cognate ArcB in anaerobic conditions, and NarL which responds to the presence of nitrate when phosphorylated by its cognate sensor NarX.

3. **Zero-component systems**: transcription factors that are considered to be active whenever they are expressed. Examples include the Fis and Hns proteins. These two proteins, for instance, are important in maintaining higher-order DNA structure and likely have complex feedback loops modulating their activity. Because this complexity is not yet fully understood, we make the simplifying assumption that these proteins are always active unless they are knocked out.

One-component systems

For a transcription factor with concentration T whose activity is directly modulated by a ligand with concentration L, we assume that the two species achieve equilibrium on a short time scale and that the affinity of the two molecules can be described by a dissociation constant K_d:

$$L + T \rightleftharpoons T^* \tag{5.6}$$

where T^* represents the concentration of the ligand-bound transcription factor.
With the dissociation constant K_d defined as:

$$K_d = \frac{L \cdot T}{T^*} \tag{5.7}$$

and conservation equations:

$$\begin{aligned} L + T^* &= L_T \\ T + T^* &= T_T \end{aligned} \tag{5.8}$$

we have:

$$\begin{aligned} T^* &= \frac{L \cdot T}{K_d} \\ T^* &= \frac{L}{K_d}\left(T_T - T^*\right) \\ T^*\left(1 + \frac{L}{K_d}\right) &= \frac{L}{K_d} T_T \\ \frac{T^*}{T_T} &= \frac{L/K_d}{1 + L/K_d} \\ \frac{T^*}{T_T} &= \frac{L}{L + K_d} \end{aligned} \tag{5.9}$$

As we can see, this result indicates that the fraction of bound transcription factor is a function of ligand concentration and the dissociation constant. Importantly, as we will discuss later, if the ligand concentration is (approximately) constant over time, the fraction of bound transcription factor is (approximately) constant over time.

We note that in cases where the stoichiometry of ligand-TF binding is n-to-1:

$$nL + T \rightleftharpoons T^* \tag{5.10}$$

and the K_d is defined as:

$$K_d = \frac{L^n \cdot T}{T^*} \tag{5.11}$$

the generalized result falls out:

$$\frac{T^*}{T_T} = \frac{L^n}{L^n + K_d} \tag{5.12}$$

To parameterize this model we need ligand concentration data and dissociation constants. We obtain internal ligand concentrations from Rabinowitz lab data [40] if available (e.g., for amino acids when *E. coli* is growing in minimal media), otherwise we assume an internal concentration of 100 μM. We obtain dissociation constants from the literature when possible. If a dissociation constant isn't available, we compute one assuming that half of the transcription factors are active at a ligand concentration of 1 μM (later we describe how these values can be refined in the context of the whole system).

To computationally simulate this model we start with total counts of free transcription factor and ligand, completely dissociated from one another. We then form one molecule of the ligand-TF

complex at a time and evaluate how close the ratio of $L^n \cdot T/T^*$ is to the actual K_d. We select the values of L, T and T^* that minimize the absolute difference between K_d and $L^n \cdot T/T^*$.

Two-component systems

Note: This work was performed primarily by Heejo Choi under the supervision of myself and Markus Covert.

For a transcription factor with concentration T; a cognate sensing protein with concentration S; a ligand with concentration L; subscripts f denoting a free (unbound) form of a molecule, b denoting a ligand-bound form of a molecule, and p denoting a phosphorylated form of a molecule; and ATP, ADP, H^+, and H_2O denoting concentrations of these molecules, we propose a system with the following:

Free (unbound) cognate sensing protein at equilibrium with ligand-bound cognate sensing protein, described by dissociation constant K_d:

$$L + S_f \rightleftharpoons S_b \tag{5.13}$$

The autophosphorylation of a free (unbound) cognate sensing protein at a rate k_A:

$$S_f + ATP \overset{k_A}{\rightarrow} S_{fp} + ADP + H^+ \tag{5.14}$$

The autophosphorylation of a ligand-bound cognate sensing protein at a rate k_B:

$$S_b + ATP \overset{k_B}{\rightarrow} S_{bp} + ADP + H^+ \tag{5.15}$$

The phosphorylation of a transcription factor by its free, phosphorylated cognate sensing protein at a rate k_C:

$$S_{fp} + T \overset{k_C}{\rightarrow} S_f + T_p \tag{5.16}$$

The phosphorylation of a transcription factor by its bound, phosphorylated cognate sensing protein at a rate k_D:

$$S_{bp} + T \overset{k_D}{\rightarrow} S_b + T_p \tag{5.17}$$

The auto-phosphatase activity of a transcription factor at a rate k_E:

$$T_p + H_2O \overset{k_E}{\rightarrow} T + P_i \tag{5.18}$$

By assuming mass-action kinetics, we can represent this system mathematically using ordinary differential equations. Similar to the one-component systems, dissociation constants are curated

from the literature when possible or assumed to correspond to a half-maximal binding at 1 μM. Many of the reaction rates can be found in [113], though we have had to adapt them to reach steady-state behavior on the time scale of a cell cycle. Ligand binding is simulated in a fashion identical to the one-component systems and the rest of the sub-model is simulated using a numerical ODE integrator.

Zero-component systems

Because we do not know enough about the regulation of these systems, we assume all transcription factors will bind to available promoter sites.

5.4.2 Modeling the modulation of RNA Polymerase recruitment

After modeling transcription factor activation, we need to model the probability that the transcription factor is bound to DNA, P_T, and, when the transcription factor is DNA-bound, its effect on RNA Polymerase recruitment to the promoter site, Δr. Recalling the notation used in the *Background* section, we want to modulate the j'th entry in the v_{synth} vector of RNA Polymerase initiation probabilities such that:

$$v_{\text{synth},j} = \alpha + P_T \Delta r \tag{5.19}$$

where α represents basal recruitment of RNA Polymerase and the second term is dependent on transcription factor activity. To represent more than one transcription factor regulating a gene, we denote the probability that the i'th transcription factor is DNA-bound as $P_{T,i}$, and the recruitment effect of the i'th transcription factor on the j'th gene as Δr_{ij}. To be consistent in our notation, if we let α_j represent the basal RNA Polymerase recruitment to the j'th gene, we have:

$$v_{\text{synth},j} = \alpha_j + \sum_i P_{T,i} \Delta r_{ij} \tag{5.20}$$

Below we will discuss how to compute $P_{T,i}$. Treating $v_{\text{synth},j}$ and $P_{T,i}$ values as constants (because we can compute both of them), we have a linear system and need to solve for α_j and Δr_{ij} terms—terms that represent RNA Polymerase recruitment. If we consider a system in which 1 gene is regulated by 1 transcription factor, we have:

$$v_{\text{synth},1} = \alpha_1 + P_{T,1} \Delta r_{11} \tag{5.21}$$

which is underdetermined (note that we have set $j = 1$ in the above equation). However, if we have $v_{\text{synth},1}$ in two conditions c_0 and c_1 (say, from our expression profiles computed from fold-change data), as well as $P_{T,1}$ values in each condition, we have:

$$\begin{aligned} v_{\text{synth},1}^{c_0} &= \alpha_1 + P_{T,1}^{c_0}\Delta r_{11} \\ v_{\text{synth},1}^{c_1} &= \alpha_1 + P_{T,1}^{c_1}\Delta r_{11} \end{aligned} \tag{5.22}$$

Armed with two equations, we can solve for α_1 and Δr_{11}. In reality, due to the noisiness of data, the procedure for finding α's and Δr's is more complicated. Before discussing that procedure in depth, though, and with this bird's eye view of the problem, we turn our attention to determining P_T, the probability that a transcription factor is DNA-bound.

Estimating P_T: an initial attempt

If we allow a slight abuse of notation (by re-using variables from the previous section) and let T represent the concentration of an active transcription factor (which, depending on the type of transcription factor, may be a ligand-bound transcription factor, a free transcription factor, or a phosphorylated transcription factor), P represent the concentration of promoter sites, and C the concentration of TF-bound promoter sites, we can have a system described by dissociation constant K_d:

$$T + P \rightleftharpoons C \tag{5.23}$$

Additionally, with conservation equations:

$$\begin{aligned} T_t &= T + C \\ P_t &= P + C \end{aligned} \tag{5.24}$$

we can write, given that $K_d = TP/C$:

$$\begin{aligned} C &= \frac{TP}{K_d} \\ K_d C &= (T_t - C)(P_t - C) \\ 0 &= (C)^2 - (T_t + P_t + K_d)(C) + (T_t P_t) \\ C &= \frac{(T_t + P_t + K_d) \pm \sqrt{(T_t + P_t + K_d)^2 - 4(T_t P_t)}}{2} \end{aligned} \tag{5.25}$$

Then, the probability that a transcription factor is promoter bound, P_T, is:

$$\begin{aligned} P_T &= \frac{C}{P_t} \\ P_T &= \frac{(T_t + P_t + K_d) \pm \sqrt{(T_t + P_t + K_d)^2 - 4(T_t P_t)}}{2P_t} \end{aligned} \tag{5.26}$$

(the solution for P_T is chosen to be between 0 and 1)

We compute P_t from the number of promoter sites on the chromosome, and, using the cell's volume, we then convert that to a concentration. Similarly, T_t is the count of transcription factor converted to a concentration. We obtain K_d values from the literature. Figure 5.3 shows that many transcription factors have nano or sub-nanomolar affinities for DNA, indicating that they bind tightly.

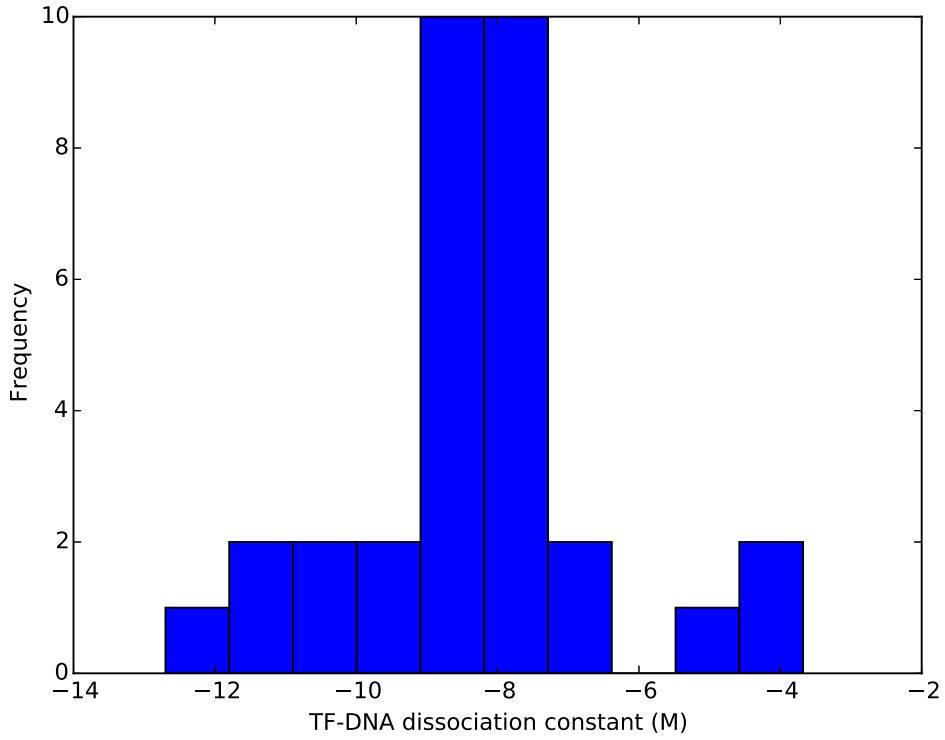


Figure 5.3: Transcription factor-DNA affinities. Most are in the nano or sub-nanomolar range.

Unfortunately, this formulation for computing P_T is very sensitive to the number of transcription factors. For example, assuming $P_t = 10$ nM, $K_d = 10$ nM, if $T_t = 5$ nM (corresponding to roughly 5 molecules in an *E. coli* cell) then $P_T = 22\%$. With those same values of P_t and K_d, if T_t is instead 10 nM (corresponding to roughly 10 molecules in an *E. coli* cell), then $P_T = 38\%$—nearly double. As we can see, for low-copy number transcription factors (e.g., with expected counts of around 10), this system is incredibly sensitive to stochastic fluctuations in counts. Recapitulating the expected gene expression for target genes is then virtually impossible to obtain. Faced with this, we propose a similar model that is much more robust to counts of molecules.

Estimating P_T: a second attempt

Consider a system where T^* represents the concentration of active transcription factor that can bind DNA with affinity K_d and modulate RNA Polymerase recruitment. Let T represent the concentration of inactive transcription factor that can also bind DNA with affinity K_d, but not modulate RNA Polymerase recruitment (transcription factors argP and ntrC are in fact known to bind DNA in both their active and inactive states [114,115]). Let P again represent the concentration of promoter sites and let C and D be concentrations of complexes formed by the following reactions:

$$\begin{aligned} T^* + P &\rightleftharpoons C \\ T + P &\rightleftharpoons D \end{aligned} \tag{5.27}$$

Defining the conservation equation $P_t = P + C + D$, we again want to find $P_T = C/P_t$. Knowing that $K_d = T^*P/C$ and $K_d = TP/D$, we can write:

$$\begin{aligned} P_t &= P + C + D \\ P_t &= \frac{K_d C}{T^*} + C + \frac{T}{T^*}C \\ P_t &= C\left(1 + \frac{K_d}{T^*} + \frac{T}{T^*}\right) \\ P_t &= C\left(\frac{T + T^* + K_d}{T^*}\right) \\ C &= P_t \frac{T^*}{T + T^* + K_d} \\ \frac{C}{P_t} &= \frac{T^*}{T + T^* + K_d} \\ P_T &= \frac{T^*}{T + T^* + K_d} \end{aligned} \tag{5.28}$$

Since most of the K_d values in Figure 5.3 are in the nano or sub-nanomolar range, we can approximate this as:

$$P_T \approx \frac{T^*}{T + T^*} \tag{5.29}$$

If we recall Equation 5.9 and the accompanying discussion, we know that $T^*/(T+T^*)$ is constant over time, assuming ligand concentrations are maintained at a steady level for one-component systems (which is the case in our simulations). This ensures that the system is relatively robust to the low-count stochasticity that plagued us in the previous formulation for estimating P_T. We note that for zero-component systems, since we define all of the transcription factors to be active, $P_T = 100\%$. For two-component systems, to model them in a fashion consistent with one-component systems, we also use Equation 5.29 to determine their activity (where T^* represents the phosphorylated form of

the transcription factor). This is summarized in Table 5.2.

Transcription factor type	Promoter-bound probability
Zero-component system	$P_T = 1$ if TF is present, 0 otherwise
One-component system	$P_T = (T^*)/(T^* + T)$
Two-component system	$P_T = (T_p)/(T_p + T)$

Table 5.2: Formulas used to compute the probability that a transcription factor is promoter-bound. T^* is the active form of a one-component system transcription factor, while T_p is the phosphorylated form of a two-component system transcription factor, and T is the inactive or unphosphorylated form of a transcription factor.

Inferring RNA Polymerase recruitment parameters: an initial attempt

We now return to our earlier discussion of a one-gene system with expression data and transcription factor activity data in two conditions. We end up with a system described by Equation 5.22 which we present here again for the reader's convenience:

$$v^{c_0}_{\text{synth},1} = \alpha_1 + P^{c_0}_{T,1}\Delta r_{11}$$
$$v^{c_1}_{\text{synth},1} = \alpha_1 + P^{c_1}_{T,1}\Delta r_{11}$$

Because we can compute $v_{\text{synth},1}$ in every condition, and because we can use Table 5.2 to compute P_T in every condition, we have a perfectly determined system of two equations and two unknowns—the unknowns being α_1 and Δr_{11}. To generalize this to a larger system, we can write this problem in matrix form. For our one-gene system, this is:

$$\begin{bmatrix} v^{c_0}_{\text{synth},1} \\ v^{c_1}_{\text{synth},1} \end{bmatrix} = \begin{bmatrix} 1 & P^{c_0}_{T,1} \\ 1 & P^{c_1}_{T,1} \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \Delta r_{11} \end{bmatrix} \tag{5.30}$$

For a two-gene system with similar regulation, this would look like:

$$\begin{bmatrix} v^{c_0}_{\text{synth},1} \\ v^{c_1}_{\text{synth},1} \\ v^{c_0}_{\text{synth},2} \\ v^{c_1}_{\text{synth},2} \end{bmatrix} = \begin{bmatrix} 1 & P^{c_0}_{T,1} & 0 & 0 \\ 1 & P^{c_1}_{T,1} & 0 & 0 \\ 0 & 0 & 1 & P^{c_0}_{T,1} \\ 0 & 0 & 1 & P^{c_1}_{T,1} \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \Delta r_{11} \\ \alpha_2 \\ \Delta r_{12} \end{bmatrix} \tag{5.31}$$

For our entire regulatory network, we can write this more concisely as:

$$v^{c_x}_{\text{synth}} = Gr \tag{5.32}$$

where $v^{c_x}_{\text{synth}}$ is a vector representing the synthesis probabilities across all conditions, G is a matrix describing the probabilities that promoters are bound by transcription factors in these conditions, and r is a vector representing the RNA polymerase recruitment strengths of each transcription

factor. By including a sufficient number of conditions (i.e., a sufficient amount of data), we design G to be square and full rank. Thus, we can solve explicitly for r:

$$r = G^{-1} v_{\text{synth}}^{c_x} \tag{5.33}$$

Having solved for the RNA Polymerase recruitment strengths stored in r, we put these values into an appropriately-shaped matrix R for use in simulation. For our two-gene system shown in Equation 5.31, R would be:

$$R = \begin{bmatrix} \alpha_1 & \Delta r_{11} & 0 & 0 \\ 0 & 0 & \alpha_2 & \Delta r_{12} \end{bmatrix} \tag{5.34}$$

In our simulations, we additionally have a vector p, computed at every time step, which describes the promoter-bound state of all transcription factors. Again, for our two-gene system regulated by a transcription factor, p would be:

$$p = \begin{bmatrix} 1 \\ p_{11} \\ 1 \\ p_{12} \end{bmatrix} \tag{5.35}$$

where $p_{11} \in \{0, 1\}$ and $p_{12} \in \{0, 1\}$ represent whether or not those specific promoter sites are bound (which occurs, for a condition c_x, with probability $P_{T,1}^{c_x}$).

The normalized matrix-vector product Rp is then used as a generalization of v_{synth} described in Equation 5.5 in the *Background* section. We note that in the situation where no transcriptional regulation is described, R is precisely a diagonal matrix with entries of v_{synth} and p is a vector whose entries are all 1—the product of Rp is thus v_{synth}. Thus, the matrix-vector product Rp is a logical generalization of v_{synth}.

Unfortunately, there is a slight problem in this formulation. The elements that correspond to α values in the R matrix are all 1. Entries that correspond to a transcription factor being promoter bound, however, are 1 with probability P_T (i.e., the time-averaged value in that entry is P_T). Returning to our example two-gene system, it may be the case that

$$\alpha_1 + \Delta r_{11}$$

is less than zero or greater than one (even if $0 \leq \alpha_1 + P_T \Delta r_{11} \leq 1$). This has the interpretation that either a negative number of RNA Polymerases will be recruited, or a number greater than the total number of active RNA Polymerases in the cell will be recruited to the promoter site—a nonsensical result. In early simulations that I ran, I saw exactly this behavior and thus had to more carefully formulate a solution to infer parameters.

Inferring RNA Polymerase recruitment parameters: a second attempt

The procedure for computing a matrix R that stores RNA Polymerase recruitment parameters by solving $G^{-1}v_{\text{synth}}^{c_x}$ and appropriately reshaping the result was flawed. One potential reason is that the experimental data used in the G matrix and in the $v_{\text{synth}}^{c_x}$ vector likely has some un-quantifiable error in it that does not match our model (e.g., perhaps due to noisiness of the associated measurements). We thus generalize the procedure for computing the matrix R so that RNA Polymerase capacity constraints won't be violated. To accomplish this, we create two linear programs (described below) that are iteratively executed until parameters converge.

The first linear program takes our G matrix and our $v_{\text{synth}}^{c_x}$ vector and computes r while (1) ensuring that RNA Polymerase capacity is not violated (e.g., no recruiting negative amounts of RNA Polymerase) and (2) ensuring that transcription factor directionality is maintained (i.e., an inducer has a positive Δr value while a repressor has a negative Δr value). Formally, it is stated as:

$$\begin{aligned} \text{minimize} \quad & \|Gr - v_{\text{synth}}^{c_x}\|_1 \\ \text{subject to} \quad & \mathbf{0} \leq Cr \leq \mathbf{1} \\ & \mathbf{0} \leq Tr \end{aligned} \tag{5.36}$$

For our two-gene, one transcription factor example system, assuming the transcription factor is an inducer, this problem explicitly looks like:

$$\begin{aligned} \text{minimize} \quad & \left\| \begin{bmatrix} 1 & P_{T,1}^{c_o} & 0 & 0 \\ 1 & P_{T,1}^{c_1} & 0 & 0 \\ 0 & 0 & 1 & P_{T,1}^{c_o} \\ 0 & 0 & 1 & P_{T,1}^{c_1} \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \Delta r_{11} \\ \alpha_2 \\ \Delta r_{12} \end{bmatrix} - \begin{bmatrix} v_{\text{synth},1}^{c_0} \\ v_{\text{synth},1}^{c_1} \\ v_{\text{synth},2}^{c_0} \\ v_{\text{synth},2}^{c_1} \end{bmatrix} \right\|_1 \\ \text{subject to} \quad & \mathbf{0} \leq \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \Delta r_{11} \\ \alpha_2 \\ \Delta r_{12} \end{bmatrix} \leq \mathbf{1} \\ & \mathbf{0} \leq \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \Delta r_{11} \\ \alpha_2 \\ \Delta r_{12} \end{bmatrix} \end{aligned} \tag{5.37}$$

We remark that our solution from our initial attempt at inferring parameters described in the previous section

$$r = G^{-1}v_{\text{synth}}^{c_x}$$

would be an optimal solution to this problem if r satisfied the two constraints (the objective value would be zero).

At this point, we could stop and use the value for r returned by the linear solver. However, this assumes that there is no error in any of the $P_T^{c_x}$ values placed in the G matrix—it assumes that our estimates for transcription factor activity are perfect, that the data going into those calculations (namely dissociation constants and ligand concentrations) have no error. Because this is unlikely, we create a second linear program which refines the $P_T^{c_x}$ values. We can then take the refined $P_T^{c_x}$ values and update our estimates of the dissociation constants and ligand concentrations.

Thus, the second linear program takes the r vector and reshapes it into a matrix $\hat{R}$ (it is different than the matrix R which is used in the simulation) and solves for an appropriately defined vector of promoter-bound probabilities $\hat{p}$ (it is different than the vector p used in the simulation) that is incentivized to be close to the initial estimate $\hat{p}_0$ and constrained to be between zero and one. Formally, it is stated as:

$$\begin{aligned} &\text{minimize} && \|\hat{R}\hat{p} - v_{\text{synth}}^{c_x}\|_1 + \lambda\|\hat{p} - \hat{p}_0\|_1 \\ &\text{subject to} && \mathbf{0} \leq \hat{p} \leq \mathbf{1} \end{aligned} \tag{5.38}$$

(Note, in our implementation, we set $\lambda = 10^{-3}$)

For our example two-gene, one transcription factor system, this looks like:

$$\begin{aligned} &\text{minimize} && \left\| \begin{bmatrix} 1 & 0 & \Delta r_{11} & 0 \\ 1 & 0 & 0 & \Delta r_{11} \\ 0 & 1 & \Delta r_{12} & 0 \\ 0 & 1 & 0 & \Delta r_{12} \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ P_{T,1}^{c_o} \\ P_{T,1}^{c_1} \end{bmatrix} - \begin{bmatrix} v_{\text{synth},1}^{c_0} \\ v_{\text{synth},1}^{c_1} \\ v_{\text{synth},2}^{c_0} \\ v_{\text{synth},2}^{c_1} \end{bmatrix} \right\|_1 + \lambda \left\| \begin{bmatrix} 1 \\ 1 \\ P_{T,1}^{c_o} \\ P_{T,1}^{c_1} \end{bmatrix} - \begin{bmatrix} 1 \\ 1 \\ (P_{T,1}^{c_o})_0 \\ (P_{T,1}^{c_1})_0 \end{bmatrix} \right\|_1 \\ &\text{subject to} && \mathbf{0} \leq \begin{bmatrix} 1 \\ 1 \\ P_{T,1}^{c_o} \\ P_{T,1}^{c_1} \end{bmatrix} \leq \mathbf{1} \end{aligned} \tag{5.39}$$

We iterate between the optimization problems described by Equation 5.36 and Equation 5.38. In practice, with our full regulatory network, it takes between 5-10 iterations (and about 10 seconds per iteration) to converge to a set of RNA Polymerase recruitment parameters r and refined promoter-bound probabilities $\hat{p}$. We reshape the r vector to obtain the R matrix used in simulations in the same manner described in the previous section. Using the values in $\hat{p}$ and Equation 5.12 we can update our estimates of dissociation constants and ligand concentrations for use in the simulation.

5.4.3 Model statistics

Table 5.3 shows the transcription factors whose functions are implemented, the conditions to modulate their activity, and the number of target genes they regulate: a total of 438 quantified interactions regulating 355 genes.

TF	TF type	Condition to modulate activity	Number of target genes
TrpR	one-component	minimal + amino acids	7
ArgR	one-component	minimal + amino acids	22
ArgP	one-component	minimal + amino acids	6
Lrp	one-component	minimal + amino acids	16
TyrR	one-component	minimal + amino acids	6
PutA	one-component	minimal + amino acids	1
AraC	one-component	minimal + arabinose	4
BglJ	zero-component	BglJ knockout	4
DnaA	one-component	DnaA knockout	3
Fis	zero-component	Fis knockout	30
Hns	zero-component	Hns knockout	76
IhfA	zero-component	IhfA knockout	49
LeuO	zero-component	LeuO knockout	11
LexA	zero-component	LexA knockout	14
MetJ	one-component	minimal + SAM	5
CytR	one-component	minimal + cytidine	3
Fnr	one-component	anaerobic	84
ArcA	two-component	anaerobic	49
BaeR	two-component	minimal + indole	5
BasR	two-component	minimal + ferric	5
DcuR	two-component	minimal + succinate	9
NarL	two-component	minimal + nitrate	29
TOTAL			438

Table 5.3: Transcription factors implemented in the model. While there are a total of 438 transcription factor-target interactions, some targets are regulated by multiple transcription factors, resulting in 355 genes being modulated by transcription factors.

One word of note: from examining Table 5.3, we see that DnaA and LexA are modulated via gene deletion. Because both *dnaA* and *lexA* are essential genes [4], simulating their knockout conditions is potentially fraught with confounding factors. However, their essential functions are not currently included in the model (e.g., the sub-model of DNA replication initiation depends on the cell reaching a certain critical mass rather than on the explicit role of DnaA), so they are functionally treated as non-essential genes in the current implementation. Once their essential functions are modeled more explicitly, their mechanism of regulation will also need to be modeled more carefully—they will no longer be able to be simulated well as zero-component systems.

5.5 Simulation Results

Figure 5.4 shows simulated results of the activity of the transcription factor TrpR when amino acids are added to the media. We see the entire cascade of events, from the increase in stimulus concentration, all the way down to the down-regulation of a target gene! This figure showcases an incredible amount: (1) the ability to simulate multiple generations (generation boundaries are clearly visible in the fourth panel as discontinuities in TrpA counts), (2) the ability to transition to a different environment which yields a different growth rate and cell composition, (3) physiological effects linking an environmental stimulus to transcription factor promoter occupancy to downstream gene expression. All of this from one unified model where we can quantitatively simulate the dynamics of single cells! While the model details and mechanisms aren't perfect (discussed below), I am not aware of any other models that can produce this output at the genome scale.

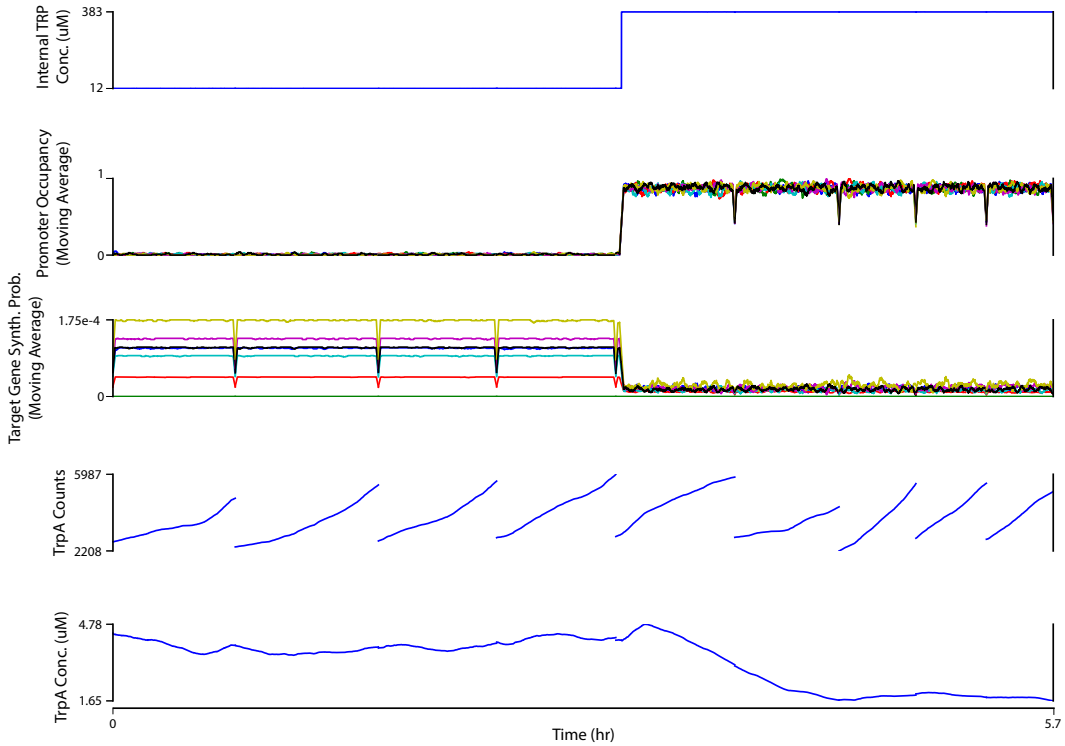


Figure 5.4: Nine generation simulation showing the effects of the addition of amino acids to the media during the fifth generation. (First panel) The internal tryptophan concentration. (Second panel) Moving average of the TrpR promoter occupancy of the 7 target genes. (Third panel) Moving average of the synthesis probabilities of the 7 target genes. (Fourth panel) Counts of TrpA, one of the targets of TrpR over multiple generations. (Fifth panel) Concentration of TrpA over multiple generations.

One interesting observation we see in the fourth panel of Figure 5.4 is that the counts of TrpA

do not change significantly before and after the addition of amino acids to the media—somewhat surprising given that it should be down-regulated. In the fifth panel, though, we reassuringly see its concentration decreases roughly three-fold. The reason for this is that cells growing with amino acids have a much faster doubling time (24 min) than cells growing without amino acids (44 min), and are considerably larger. To sustain the faster doubling time, more transcription (and translation) must occur globally, so protein counts of TrpA are relatively constant. However, the fraction of all transcription (and translation) that is devoted to TrpA decreases, and so we see a decrease in its concentration.

While Figure 5.4 demonstrates the behavior of one transcription factor, there are 21 others to assess. Figure 5.5 shows, for each transcription factor in its active and inactive condition: (1) the probability that it is promoter bound (top panel), and (2) the synthesis probabilities of each target gene (bottom panel). Prior to simulation, we compute estimates of each of these values, which we compare to values obtained from running simulations. We see that, in general, these values are in very good agreement and fall on the $y = x$ line, except for (1) in the top panel, small probabilities of being promoter bound which demonstrate noise we expect to see with rare events and (2) in the bottom panel, synthesis probabilities of genes regulated by two-component systems, which, due to their model of signal transduction, are noisier than the zero- and one-component systems.

5.6 Limitations

While our integrated regulatory model is an improvement over previous work, it nonetheless has a number of limitations that arise from data and modeling considerations:

- **Regulation at the gene level rather than operon level**. We model transcription—and therefore transcriptional regulation—at the gene level rather than the operon level. Every gene has a pseudo-promoter site from which RNA Polymerase can initiate. In the baseline condition, I performed RNA-seq and mapped reads back to individual genes because, to my surprise, *E. coli* does not have a well-annotated transcriptome: while there are nearly 4000 annotated promoters [116], there less than 300 annotated terminator sites [117]. Even with a good reference transcriptome, all of the fold-change data measuring expression in different conditions is reported on a per-gene basis, and, as far as I'm aware, there is no good way to convert these fold-changes to be per-operon without having detailed knowledge of potential confounding factors such as nested operons and the effects of transcriptional attenuation.

  To address this issue, in principle all of the differential gene expression measurements could be redone using a long-read sequencing technology that measured operon-level expression. We could then replace our current per-gene data with this new data set and use our same mathematical framework to model transcription and transcriptional regulation. Would funding agencies be keen on allocating their resources for this purpose?

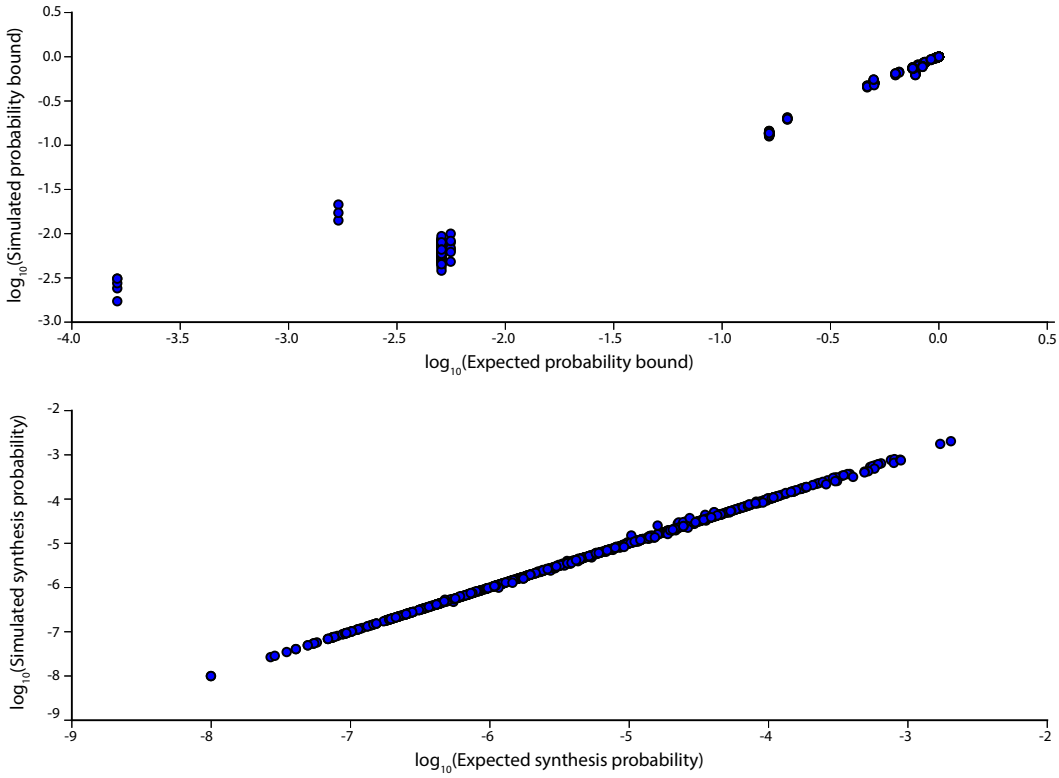


Figure 5.5: Transcription factors function as expected. (Top panel) The expected probability, computed prior to simulation, that a transcription factor is promoter bound in a given condition compared to simulated probabilities. (Bottom panel) Expected synthesis probabilities of regulated target genes, computed prior to simulation, compared to simulated values.

- **Auto-regulation is not included**. The studies that provide the bulk of our fold-change data (e.g., [118]) over-express transcription factors on a plasmid and thus do not properly measure auto-regulation. To quantify auto-regulation we would want to mutate the promoter site of the endogenous copy of the transcription factor, abolishing its ability to bind. This should be possible using CRISPR technology [119].

- **For a given transcription factor, all transcription factor-DNA affinities are assumed equal**. When a simulated transcription factor is active and binds a promoter for a target gene, it selects a promoter at random from an unweighted uniform distribution. Perhaps by using MITOMI technology [120] being developed in the Fordyce lab, we can weight this distribution in a sequence-specific manner.

- **Effects of other RNA regulation are lumped in**. Since we lack quantitative data on

attenuation and (anti-)termination, these mechanisms of regulation are not currently implemented.

- **Sigma factors are not included**. While it is well-recognized that sigma factors play an important role in gene regulation, they were not characterized at the genome-scale when we started this project and explored whether we could incorporate their functionality.

- **DNA structure and DNA copy number is not included**. We do not currently have a genome-scale model of DNA structure in our *E. coli* model and so we do not mechanistically model transcriptional bursting [121]—the average rate of RNA Polymerase initiation at a promoter is constant. Even with such a model, it may be hard to tune expression parameters to sustain growth (see Chapter 8 for further discussion on this point). DNA copy number should, in principle, be easier to handle. Naively, from a reported growth rate, we can infer average chromosome copy number in a straightforward manner and then, using a gene's position, estimate its copy number. Unfortunately, most of our differential expression data sets do not report growth rate. However, even if growth rate were reported, the calculation wouldn't be that straightforward—it could easily be confounded by (auto-)regulation, for example.

The aforementioned limitations are largely limitations of data. In general, models in systems biology can (and do) become more mechanistic and quantitative when new data sets become available. These model limitations can, perhaps, be used as motivation or justification to generate new data sets or to develop the requisite technologies.

Chapter 6

A kinetically constrained model of metabolism in *E. coli*

6.1 Background

Metabolism imports and transforms raw materials from the environment into energy and structural building blocks for the cell. Given this central role in cell physiology, the scientific community has devoted considerable effort to characterizing biochemical pathways experimentally and mathematically modeling their interactions as part of a larger network. While there have been efforts to simulate small networks using ordinary differential equations (ODEs) (e.g., [122]), scaling these efforts to larger systems has proven challenging. Since there is no high-throughput technology to characterize these systems, obtaining parameters is difficult. Furthermore, because many kinetic parameters are strong functions of the environment (e.g., salt concentrations, pH, temperature) in which they are measured, reconciling them in one unified model is computationally challenging, particularly because of the non-linearities present in these systems. To circumvent these challenges, constraint-based models have been developed which account for hundreds or thousands of reactions and only need a handful of measurements to parameterize them (e.g., uptake rates) [123]. In order to achieve this feat, these models make two major assumptions: (1) that the metabolic network is at steady state, and (2) that the metabolic network is optimizing an objective function, usually to maximally produce cellular components ("biomass"). Importantly, when data is available to constrain fluxes, that data can be incorporated into the network with relative ease [18, 124]. Given these benefits of constraint-based models, it was logical to incorporate one as the sub-model for metabolism of *M. genitalium*.

Concretely, a constraint-based problem for a metabolic model often has the form:

$$\begin{array}{ll} \text{maximize} & \text{production of cellular resources} \\ \text{subject to} & \text{the metabolic network is at steady-state} \\ & \text{uptake rates of nutrients do not exceed measured values} \end{array} \tag{6.1}$$

More mathematically, this is often written as:

$$\begin{array}{ll} \text{maximize} & v_{bio} \\ \text{subject to} & Sv = 0 \\ & v_l \leq v \leq v_u \end{array} \tag{6.2}$$

where the matrix S describes the stoichiometry of every reaction in the metabolic network (usually a "fat" matrix with many hundreds of rows and over a thousand columns), the vector v (to be solved for) stores the fluxes for every reaction, the element v_{bio} of vector v corresponds to a column in S specifying the cellular resources to be maximized in proportion to one another, and v_l and v_u are constraints on v that incorporate measured uptake rates (e.g., of glucose and oxygen) as well as reaction directionality determined by thermodynamic considerations. For a reversible reaction without any constraints, the corresponding entries in v_l and v_u would be $-\infty$ and $+\infty$, respectively. An irreversible reaction would have an entry in v_l of 0 and an entry in v_u of $+\infty$.

This formulation of the problem was used in the *M. genitalium* simulations to compute cellular resource production at every time step with one major additional piece of information. Because the whole-cell model computes protein levels, if a maximal kinetic rate (k_{cat}) is known for an enzyme, its corresponding entry in v_u may be set to $k_{\text{cat}} \cdot E$ (where E is the amount of enzyme) to constrain the internal flux of the reaction catalyzed by that enzyme. Kinetic parameters were curated from the literature for a variety of organisms based on enzyme homology, but it was found—perhaps not surprisingly—that when these parameters were measured for the native *M. genitalium* enzymes, growth rate predictions of gene-deletion strains were more accurate [10]

There are a few issues in using this formulation of a constraint-based problem in a whole-cell modeling context. One is that the cellular resources produced at each time step are an average of the resources needed over an entire cell cycle. As a result, cell cycle-dependent metabolite usage (e.g., production of the dNTPs needed for DNA replication) is not well-modeled. Another issue is that if any single cellular resource cannot be produced at a given time step, none of them will be produced—one can easily imagine that this could have catastrophic consequences for the cell. Jonathan Karr had to fit parameters controlling enzyme levels to avoid this scenario from occurring in simulations. My labmate Elsa Birch encountered similar issues when modeling the effects of T7 infection on metabolism [125] and set out to explicitly address this "all-or-none" behavior by developing a flexible objective function that would allow the network to be robust to deficiencies in any single cellular resource [126].

Early versions of the *E. coli* model used Elsa's flexible objective until John Mason realized that for *E. coli* in particular, we have an extra piece of data that would also help us solve the issue of time-dependent/cell-cycle dependent resource usage: measurements of metabolite pools for nearly 100 molecular species [40]. This led to the development of a homeostatic objective. Additionally, measurements of metabolite pools enable us to incorporate more detailed kinetic parameters: not just k_{cat} values but also K_m values that describe enzyme saturation. Below we describe both the development of a homeostatic objective as well as the incorporation of detailed kinetic parameters into a constraint-based model of metabolism.

6.2 Homeostatic Objective

In an *E. coli* whole-cell modeling context, with concentrations of metabolite pools at our disposal, we can state our constraint-based problem as follows:

$$\begin{array}{ll} \text{minimize} & \text{the difference in metabolite pool concentrations from their setpoints} \\ \text{subject to} & \text{the metabolic network is at steady-state} \\ & \text{uptake rates of nutrients do not exceed measured values} \\ & \text{constraints on internal reaction fluxes are not violated} \end{array} \tag{6.3}$$

More formally, if we let $c_{o,i}$ denote the measured concentration of the i'th metabolite (and let this be the setpoint), and if we let $v_{c,i}$ denote a flux for production of the i'th metabolite within a time step (so that $v_{c,i}$ also has units of concentration), then we state our constraint-based problem as:

$$\begin{array}{ll} \text{minimize} & \displaystyle\sum_i \left| 1 - \frac{v_{c,i}}{c_{o,i}} \right| \\ \text{subject to} & Sv = 0 \\ & v_l \leq v \leq v_u \end{array} \tag{6.4}$$

This problem incentivizes the metabolic network to maintain concentrations of metabolite pools. These metabolite pools (e.g., of NTPs, dNTPs, amino acids) are the cellular resources that get used by other physiological processes (e.g., transcription will use NTPs, DNA replication will use dNTPs, translation will use amino acids). The metabolic model will essentially replenish what these other simulated processes used in the previous time step, subject to the constraints v_l and v_u placed on fluxes.

After testing this formulation, we noticed that metabolism would, depending on its precise inputs, secrete mass (e.g., in the form of acetate) from the cell—so much mass that the cell would have

a negative growth rate. To disincentivize this behavior, we added a penalty term, making the constraint-based problem:

$$\begin{aligned} \text{minimize} \quad & \sum_i \left| 1 - \frac{v_{c,i}}{c_{o,i}} \right| + \gamma\, m_s \\ \text{subject to} \quad & Sv = 0 \\ & v_l \leq v \leq v_u \end{aligned} \tag{6.5}$$

where m_s is the mass secreted from the cell and γ is a tunable hyper-parameter. Values of γ as low as 10^{-5} are sufficient to avoid egregious mass secretions, while values above 10^{-2} do not allow the network to import sufficient nutrients to sustain cell growth. Somewhat surprisingly, with the upper bound on glucose uptake set to a relaxed value of 20 mmol/gDCW/hr, the actual glucose uptake rate is 10.3 mmol/gDCW/hr, which is in very close agreement with the literature-reported value (for FBA simulations) of 10 mmol/gDCW/hr [18, 127]. This implies that our whole model is not using an unrealistic amount of energy to fuel growth.

At this point, the only constraints on v_l and v_u describe nutrient uptake rates and reaction reversibility. Below we describe the steps necessary to incorporate Michaelis-Menten parameters into this model.

6.3 Data Curation

As mentioned earlier, because we are modeling *E. coli*, we have a unique opportunity to incorporate detailed organism-specific kinetic parameters. At first we attempted to simply download these parameters in bulk from the BRENDA database [78]. Upon initial inspection, though, we observed that much of the data characterized mutant enzymes and/or non-canonical substrates. We were further troubled by a report from Ron Milo's group that stated, "We found up to 20% of the values in the Brenda database do not correspond to the values reported in the corresponding reference papers. These discrepancies are the result of erroneous copying and unit mismatch. In several cases, the values in the database were orders of magnitude higher/lower than those given in the original works." [128] (quoted text is in the supplement to the paper). Thus, to ensure that the highest-quality data would go into the model, Markus decided to read each reference paper and record the reported kinetic parameters.

From an initial list of approximately 12,000 papers referenced by BRENDA, we filtered out papers that lacked a k_{cat}, did not use a lab strain of *E. coli*, or did not involve enzymes in our metabolic network. From the remaining ˜1, 200 papers that were manually curated, we obtained 400 constraints for 340 reactions (some of the reactions have multiple constraints—for example K_m values for multiple substrates). We adjust reported k_{cat} values for temperature using a scaling factor given by:

$$2^{\frac{37-T}{10}} \tag{6.6}$$

where T is the reported temperature (in °C) for the experimental conditions—this increases the kinetic rate by a factor of 2 for every 10°C away from 37°C. Of the 400 constraints, 181 include both k_{cat} and K_m information, while 219 include only k_{cat} information. In the simulation, at each time step, we select the most relaxed constraint for each of the 360 constrained reactions.

6.4 Incorporation of kinetic targets into the objective

For the j'th reaction with a constraint, if we have a K_m parameter, we set the flux target $v_{t,j}$ as follows:

$$v_{t,j} = k_{\text{cat}} E \frac{S}{S + K_m} \tag{6.7}$$

where E is the enzyme concentration, S is the substrate concentration, k_{cat} is the enzyme's maximal rate, and K_m (in conjunction with with substrate concentration) determines the enzyme saturation level.

If we lack data for S and/or K_m, then we write:

$$v_{t,j} = k_{\text{cat}} E \tag{6.8}$$

Incorporating the kinetic targets as soft-constraints, we then formulate our constraint-based problem as:

$$\begin{aligned} \text{minimize} \quad & \sum_i \left| 1 - \frac{v_{c,i}}{c_{o,i}} \right| + \lambda \sum_j \left| 1 - \frac{v_j}{v_{t,j}} \right| + \gamma \, m_s \\ \text{subject to} \quad & Sv = 0 \\ & v_l \leq v \leq v_u \end{aligned} \tag{6.9}$$

Here, λ is a tunable hyper-parameter. When it is set to 10^{-7}, the entire kinetics term makes an approximately equal contribution to the objective value as the homeostatic term, so we have it set at this value in our simulations. When it is as high as 10^{-4}, we observe growth deficiencies. Additionally, if the enzyme concentration E ever reaches zero for a reaction, we set the corresponding values in v_l and v_u to 0, enforcing that the reaction has zero flux.

6.5 Results

Simulation results are shown in Chapter 7, Figure 7.3B.

6.6 Limitations

The main limitations of this method I think generally apply to any constraint-based metabolic model. Is the steady-state assumption valid, particularly at the 1 second time scale? Is the objective function reasonable? Ideally, the objective function (whether it is to maximize biomass production or maintain homeostasis) would emerge naturally from enzymatic parameters and substrate concentrations rather than get set in a top-down fashion. However, until we have technologies to characterize and quantify enzyme function and regulation *in vivo* in a high-throughput fashion, constraint-based methods will continue to play a significant role in whole-cell modeling efforts.

Chapter 7

Crick's "complete solution of *E. coli*," 40 years later

Four decades ago, Francis Crick advocated for a coordinated worldwide scientific effort to determine a "complete solution" of *Escherichia coli*. Since then, millions of measurements have been published using this organism. To what extent do these data constitute a "solution"? Here, we use large-scale modeling to simultaneously evaluate a massive set of heterogeneous experimental results in *E. coli*. We show that these data are strikingly self-consistent with respect to gene and protein expression and cellular metabolism as well as replication and growth physiology. Surprisingly, the totality of the data suggests that a clear majority of *E. coli* genes are expressed less than once per generation—including the genes associated with antibiotic resistance and persistence—and yet the cell is robust to this behavior. Our findings cross-validate experimental measurements made largely independently across the world and over many decades, and suggest several lines of fruitful inquiry for the future.

Crick recommended this coordinated effort be performed both for "the intellectual satisfaction of having a single living cell 'completely' explained", as well as in order to make major advances in biology in the most efficient way [129]. He suggested that the best way to accomplish this would be to establish a central laboratory which could coordinate work across nations, standardize biological reagents and materials, and produce vast libraries. Although such an approach was never adopted, the scientific community has performed and published millions of measurements, of many different kinds and in hundreds of laboratories, using *E. coli* over the intervening decades. To what extent,

Manuscript submitted to *Nature* on May 3, 2017. Author list: Derek N. Macklin*, Nicholas A. Ruggero*, Javier Carrera*, Heejo Choi, Travis A. Horst, John C. Mason, Mialy M. DeFelice, Inbal Maayan, Morgan L. Paull, Sajia Akhter, Samuel R. Bray, Daniel S. Weaver, Ingrid M. Keseler, Peter D. Karp, Markus W. Covert

if any, do these data constitute a "solution"?

The answer to this question has implications beyond *E. coli* or even microbiology, as high-profile studies published recently have led this journal and others to question the reproducibility of scientific results in multiple scientific fields [130, 131]. If this is the case, then we should worry about whether what we "know" about *E. coli* is actually correct in two ways. First, are the data reproducible (i.e., does a repeated study produce the same measured outcomes)? Second, and more deeply, are the data cross-verifiable—meaning, does a multiplicity of heterogeneous data all point to the same conclusion?

This second and more important question requires us to consider all of the data together in their biological context, integrating and evaluating them as a whole—which in turn depends on theoretical approaches based on mathematical representations of known or inferred biological mechanisms. The application of theory to understand, consolidate, and even verify data is not new. The most famous expression of the latter point is attributed to Sir Arthur Eddington: "I hope I shall not shock the experimental physicists too much if I add that it is also a good rule not to put overmuch confidence in the observational results that are put forward *until they are confirmed by theory*" (italics his) [132]. In fact, this motto guided Watson and Crick's approach to developing a model of the double helical structure of DNA, for which Crick asserted that "people don't realize that not only can data be wrong in science, it can be *misleading*" (italics his) [133]. As a result, some data were set aside if they were inconsistent with the theory of the double helical structure.

Per Eddington's remark, our goal is to use large-scale, integrative modeling of *E. coli*'s fundamental biological processes as a theory, to confirm (or at least evaluate) the observational results that have been reported in *E. coli* over the past century. Obviously, the success of this effort depends on a mathematical approach which can both represent these biological processes mechanistically, as well as accommodate many millions of data points, which exhibit an extraordinary amount of heterogeneity. Attempts to mechanistically model cell behavior at the large scale also span several decades [134–138]. We recently demonstrated a large-scale modeling approach which was capable of integrating all of the known functions in the simplest culturable bacterium, *Mycoplasma genitalium* [7]. Notably, the model successfully reproduced many measured data, and even predicted previously unmeasured parameters which were subsequently verified experimentally [10].

Encouraged by this success, here we apply our approach to *E. coli*, arguably the best characterized organism (see Figure 7.1). *E. coli* has nearly ten times more genes than *M. genitalium*, is comprised of roughly 50-100 times as many molecules that interact and react, can readily grow in a wide variety of environmental conditions, and exhibits extensive self-regulation and control, all of which pose significant challenges to whole-cell modeling. However, one of the most exciting aspects of modeling *E. coli* at the large scale is the enormous effort in data generation that has already been performed—and in many cases, stored in an accessible format [139]. Thus, whereas only 27.5% of the parameter values used in our *M. genitalium* model were actually measured in that organism,

the model we describe here contains parameter values that are 100% measured in *E. coli*. This fact provided us an opportunity to assess the data against itself to a degree that would not be possible in any other organism.

In the Supplemental Materials, we describe in detail a model that fully integrates the Central Dogma together with carbon and energy metabolism, and in the context of balanced growth. Functionally, 1,214 genes (or 43% of the well-annotated genes) have been included to represent these processes—which are also those for which the lion's share of the existing *E. coli* data has been generated. The model has been optimized for three different environments, which are relatively well-characterized experimentally and exhibit diverse phenotypic behaviors: a minimal (M9 salts plus glucose, aerobic conditions), rich (minimal + all amino acids), and minimal anaerobic medium. From the totality of the data, over nineteen thousand parameters were identified and included in the model, while <1% were fit—which underscores that it is the data that are being tested by the theory, not vice versa. We then applied this model to assess the cross-verifiability of this massive dataset. We decided on three evaluative criteria: (1) *Equations* – can the data be encapsulated mathematically? (2) *Simulations* – does the model output "make sense" with respect to known cell behaviors? (3) *Validation* – are the data consistent with one another?

We first evaluated these criteria in terms of what Crick called "the Central Dogma", namely the expression of genes in terms of mRNA and protein. Gene expression has been studied in extensive detail, with datasets that characterize mRNA expression under a variety of environmental conditions (including an RNA-Seq dataset that we generated for this study, see Supplemental Materials), mRNA and protein half lives, kinetics of RNase activity, experimentally-determined RNA-polymerase and transcription factor binding sites, dissociation constants for proteins bound to DNA binding sites or other cellular and environmental ligands, transcription unit structure, translational efficiencies of mRNA transcripts, and cellular growth rate and chemical composition measurements (see Supplemental Materials for complete descriptions of all data used). In terms of our Equation criterion, we found that these datasets can indeed be integrated mathematically, beginning with a basis of thousands of ordinary differential equations (shown for a representative mRNA and protein in Figure 7.2A) that we then implemented as stochastic simulations (see Supplemental Materials for a more detailed description of the modeling approach).

To address the Simulation criterion, we ran simulations of single cell life cycles and observed whether the parameters could yield plausible trajectories of mRNA and protein production. Representative traces for genes encoding both high- and low-stability mRNAs are shown in Figure 7.2B; these exhibited characteristic expression dynamics wherein mRNA was produced at random times, leading to a concomitant sharp increase in protein production, as has been observed experimentally [140].

Using the model, we also identified a number of ways in which these data could be cross-validated against each other (Figure 7.2C). We first compared expression fold changes that were determined

from independent microarray and RNA-Seq datasets (based on an earlier compilation [141]), by identifying experiments that would be expected to have the same or similar gene expression outcomes (e.g., when an environmental stimulus experiment and a separate genetic perturbation experiment modulate the activity of the same transcription factor). In such cases, the mRNA fold changes were strongly correlated (72% of pair shifts have PCC > 0.7 or higher, Figure 7.2D). We then compared the same fold changes to the known regulatory topology extracted from the RegulonDB and EcoCyc databases [139,142]. This analysis indicated that over 76% of the fold-changes determined from array or RNA-Seq data are consistent with the topology of the transcriptional regulatory and signaling networks (Figure 7.2E). For a third test, we compared the total simulation output for protein counts in a cell to an experimentally-determined proteome which was not used to parameterize the model [143], and found the agreement to be statistically significant (Figure 7.2F, PCC = 0.75, $p < 10^{-20}$, $n = 2,233$).

Finally, we recognized that under steady-state conditions, the rate of protein synthesis should equal the rate of decay. This proved to largely be the case in our simulations, with 85% of the production rates within an order of magnitude of the decay rate (Figure 7.2G). This was particularly surprising given that protein decay rates are very lightly characterized, and are thus usually estimated by the "N-end rule", whereby protein half-lives are assumed to have values of ˜2 minutes or ˜10 hours, based on their N-terminal amino acid [144]. We wondered whether some of the outliers in our comparison might be due to a more nuanced or specific value for the protein half-life. To test this hypothesis, we determined the half-lives of six outlier proteins experimentally, and found that in all cases, the half-life predicted by our model was a better predictor of the data than the N-end rule (Figure 7.2H). In three of the cases, the new half-life was sufficient to explain the discrepancy between the protein production and decay rates; for the remaining three, other as-yet unknown factors are also likely to play a role. In total, all of our validation tests and follow-on experiments suggested that the datasets included in this model are highly self-consistent with regard to gene expression and the Central Dogma.

We next considered our evaluative criteria in terms of *E. coli*'s metabolic network. This network accounts for 962 enzymes, and has been investigated as part of thousands of studies which determined the stoichiometry for each chemical reaction (as compiled by others [145, 146]), identified and characterized virtually all of the enzymes and transport proteins in the network, and measured concentrations of > 100 small molecules internal to the cell [147], as well as physiological properties (e.g., chemical composition of the cell, maximum uptake and secretion rates of carbon sources and by-products), key fluxes in central carbon metabolism, and detailed kinetics for many enzymes. The processes associated with the Central Dogma are also deeply relevant here, as the concentrations of enzymes are used to determine kinetic constraints.

As before, it is possible to represent these networks mathematically by writing thousands more

ordinary differential equations which describe the concentrations of small molecules over time (Figure 7.3A) and making subsequent simplifying assumptions in order to model flux through a metabolic network using linear optimization and flux balance analysis [148]. We have found that with significant modifications—particularly with regard to the incorporation of regulation of enzyme expression [149], the inclusion of enzyme kinetic constraints (e.g., based on metabolite concentrations and Michaelis-Menten parameters) [7,150], and substantial innovation with regard to the objective function and bounds [151] (see Supplemental Material for details)—this method can be incorporated into whole-cell modeling.

Referring to the Simulation criterion, we examined the output for the metabolic network. Figure 7.3B depicts a selection of model output (including the flux distribution over time for metabolic fluxes related to central carbon metabolism and amino acid biosynthesis) for the shift that occurs when cells initially growing in an aerobic glucose minimal medium are presented with the full range of amino acids. This perturbation leads to two major shifts in flux. First, the fluxes which normally synthesize amino acids are dramatically reduced. Second, the addition of amino acids leads to a higher growth rate (discussed in more detail below), which in turn drives another set of enzymes to produce more DNA precursors. Both these and the other model outputs are highly consistent with what we would expect.

We then performed validation tests of the metabolic data to determine whether the totality of the constraints imposed on the metabolic network by Michaelis-Menten constants and small molecule concentrations was internally consistent. This required a modification to our linear optimization approach which lightly penalized flux distributions which diverged from these kinetic constraints (see Supplemental Material). We found that incorporation of these constraints (400 in total, constraining 340 metabolic reactions) led to a growth yield that was lower than has been measured experimentally. Further investigation identified two metabolic reactions from the Citric Acid cycle for which the incorporation of kinetic constraints reduced the yield (succinate dehydrogenase and fumarate reductase, see Supplement). With the constraints on these reactions relaxed, we ran simulations and compared the output metabolic fluxes to our kinetic constraints (Figure 7.3C). Twenty-eight of the fluxes were unused by the simulation (i.e., had a zero value), due to the fact that the model is not yet whole-cell. The remaining fluxes exhibited a remarkable degree of consistency, where 87% were within an order of magnitude of the constraint value (PCC = 0.86, $p < 10^{-100}$, $n = 356$). Next, we compared simulated results to another independent dataset that was withheld from our model parameterization (Figure 7.3D) [152]. With two exceptions, the simulation and data were highly correlated (PCC = 0.80, $p < 10^{-4}$, $n = 21$). Interestingly, succinate dehydrogenase was one of the proteins for which the kinetic constraint was relaxed as mentioned above, and the other (isocitrate dehydrogenase) is a near neighbor in the Citric Acid cycle, which invites further investigation. Overall, these results indicate a strong consistency between all of the datasets we compiled relative to metabolism.

Our final evaluation concerned measurements associated with cell growth and physiology. We were able to encapsulate data on cell physiology across multiple growth conditions in a set of equations describing the relationship between cell size, growth rate, ribosome concentration, ribosome elongation rate, the rate of DNA initiation, and other cell cycle parameters (Figure 7.4A). Considering our Simulation output and experimental Validation criteria, we examined the physiological response to the same environmental shift depicted in Figure 7.3B for metabolism (i.e., adding amino acids to a defined minimal medium), and found that this simulated up-shift in nutrient availability led to a steady-state increase in cell size, growth rate and ribosome concentration as expected (Figure 7.4B). The dynamics of the response also met expectations, with the RNA fraction's instantaneous growth rate changing to the new expected growth rate relatively quickly, while other mass fractions and the total cell growth rate converged more slowly (Figure 7.4C), as others have observed [153]. These simulation outputs were also consistent with experimental measurements of the RNA mass per cell, ribosome elongation rate and stable RNA synthesis rate at three different doubling times (Figure 7.4D) [154].

Next, we examined the dynamics of simulated chromosome replication and its response to the same environmental shift. We modeled the initiation of chromosome replication as occurring at a fixed ratio of cell mass per origin of replication [155], which produces multiple overlapping rounds of replication at faster growth rates [156]. This approach correctly reproduced expected replication fork dynamics, including periods where no DNA polymerization is occurring at slow growth rates, and multiple simultaneous rounds of replication at fast growth rates (Figure 7.4E). Furthermore, the rate of DNA replication initiation correctly scaled to match the growth rate both before and after the shift in environment, and was also confirmed by population measurements of the number of origins of replication per cell upon DNA replication initiation (Figure 7.4D).

We then coupled each chromosome replication initiation event to a cell division event which occurred after a fixed period of time for DNA replication and cytokinesis (occurring zero, one or two generations in the future depending on the growth rate), as inspired by recent work [157] (Figure 7.4E). The interaction of (1) variation in the growth rate between individual cell simulations and (2) coupling cell division to a cell cycle event at a fixed mass led to two possible effects with regard to mass added per cell cycle, depending on the growth rate. In fast growing cells, the cell mass added over the life cycle was uncorrelated with the initial cell mass—a phenomenon referred to as "adder" behavior [158,159], whereas for slower growing cells the added and initial cell masses were correlated ("sizer" behavior) [157] (Figure 7.4F). Integrating the responses of growth rate and chromosomal replication to media conditions therefore enabled us to simulate cellular behavior over many generations with stable cell size distributions—in dramatic contrast to our original *M. genitalium* model, which could only grow stably for one cell cycle [7]

Finally, having access to all of these data simultaneously in a single integrated model enabled us to ask the question: can these data tell us something *in toto* that wouldn't have been obvious from

any individual experiment? In this regard, the most striking observations we made relate to the fact that, although many genes are transcribed multiple times as a typical cell grows and divides, a clear majority of the genes in *E. coli* are transcribed at a rate of less than one per cell cycle (Figure 7.5A,B). In the case of growth on minimal glucose medium, 35.7% of the genes are transcribed at least once per cell cycle, and 4.6% are essentially never expressed in this environment. The remaining 59.7% are transcribed with a frequency that is distributed almost uniformly between 0% and 100% (Figure 7.5B). This leads to dramatic consequences for the dynamics of protein expression, where a protein might be expressed only once in ten generations—but with a much higher fold change—and otherwise simply be diluted in concentration as the cell grows and divides (Figure 7.5A,C,D).

Investigating this phenomenon further, we found that the bulk of essential genes are found in the group with 100% cell cycle expression (Figure 7.5E), and most genes with unknown functionality or poor annotation are in the group of genes that were expressed less than once per cell cycle (Figure 7.5F). More surprising was the finding that certain functional categories are also predominantly in certain groups. For example, 74.3% of the genes associated with antibiotic resistance or persistence are found to be expressed less than once per cycle (Figure 7.5G), a compelling finding given that persistent bacteria appear to be phenotypically different from their siblings in culture [160,161].

Considering these low-expression genes with respect to the rest of our compiled mRNA expression data, we were able to divide them into two categories. First, we found that one group of 1,945 genes was *inducible*—they were expressed at low expression levels in one or more environmental conditions, but at 100% cell cycle expression levels in other conditions. The other group of genes, 1,181 in total, were found to be *constitutively* expressed less than once per cycle, with a low probability of synthesis under all of the environmental conditions we considered (Figure 7.5H).

This finding implies that many proteins required for cell survival and growth may be absent from the cell for periods of time. We therefore wondered, how does the cell compensate for the temporary loss of an important enzyme due to very low expression rates? For example, O-succinylbenzoate-CoA ligase is an enzyme involved in menaquinone (sometimes known as Vitamin K_2, important for respiration and electron transfer in bacteria) biosynthesis. This ligase is encoded by the *menE* gene, which is transcribed with a frequency of ˜1.1 times per cell cycle (Figure 7.5I), producing an average of 5.2 proteins per generation. The ligase is only active as a tetramer, and thus the average count of active complex in the cell is ˜7.2. Our simulations exhibit periods of time in which no tetramer exists (gray regions). During these $MenE_4$ starvation phases, when the tetramer is completely absent, the internal concentration of menaquinone is reduced over time; however, following a new round of $MenE_4$ expression, the menaquinone is rapidly re-synthesized. Sufficient internal metabolite pools and rapid enzyme kinetics is therefore one mechanism that can make cell growth robust to the loss of a key enzyme for periods of time.

In conclusion, the integration of experimental data using a large-scale theoretical framework allowed us to compile a massive, self-consistent dataset. Our findings cross-validate hundreds of

thousands of experimental measurements made largely independently across the world and over many decades. These findings stand in stark contrast to the reports on reproducibility cited above, and we believe that with respect to *E. coli* at least, we can have confidence in the scientific community's output.

Importantly, this work depended on the use of theory to validate the experimental data, and not vice versa. Such an application of theory has been rare in the biological sciences—possibly because the large-scale integrated modeling approach we use is somewhat new—but this work demonstrates its potential, with regard to understanding, interpreting and cross-validating large data sets, and even suggesting lines of fruitful experimental inquiry for the future.

Returning to Crick's original monograph, does the totality of the data presented here, overlaid with a theoretical framework, constitute a "complete solution"? Crick gave a more detailed definition—"By 'complete' one means complete in the intellectual sense, implying that nothing appears to remain which further experiment could not easily explain using well-established facts and ideas" [129]. By that metric, our work clearly falls short—not only because of the limited number of genes, molecules and environments that are accounted for, but also because the correlations and comparisons we show here, while strong, could yet be improved and expanded by directed future experimentation.

Though not complete, the data and model may still represent a solution of a certain kind. In advanced mathematics, theorists recognize a distinction between particular and general solution types, and know that often the identification of particular solutions can lead to characterization of the general. In that context, we argue that this work represents a particular solution of *E. coli*—a solution that is only relevant to specific conditions, does not yet represent all known cellular processes, and incorporates a dataset that though extensive, is still a sample of all of the possible data which could be included—but which also provides a foundation upon which other functionalities, environments and datasets can readily be built. We therefore hope and anticipate that this work may provide an alternative to a "central lab", which can encourage, motivate and support both theorists and experimental scientists—in the light of four decades of accomplishment—to reassess and eventually surmount Crick's grand challenge.

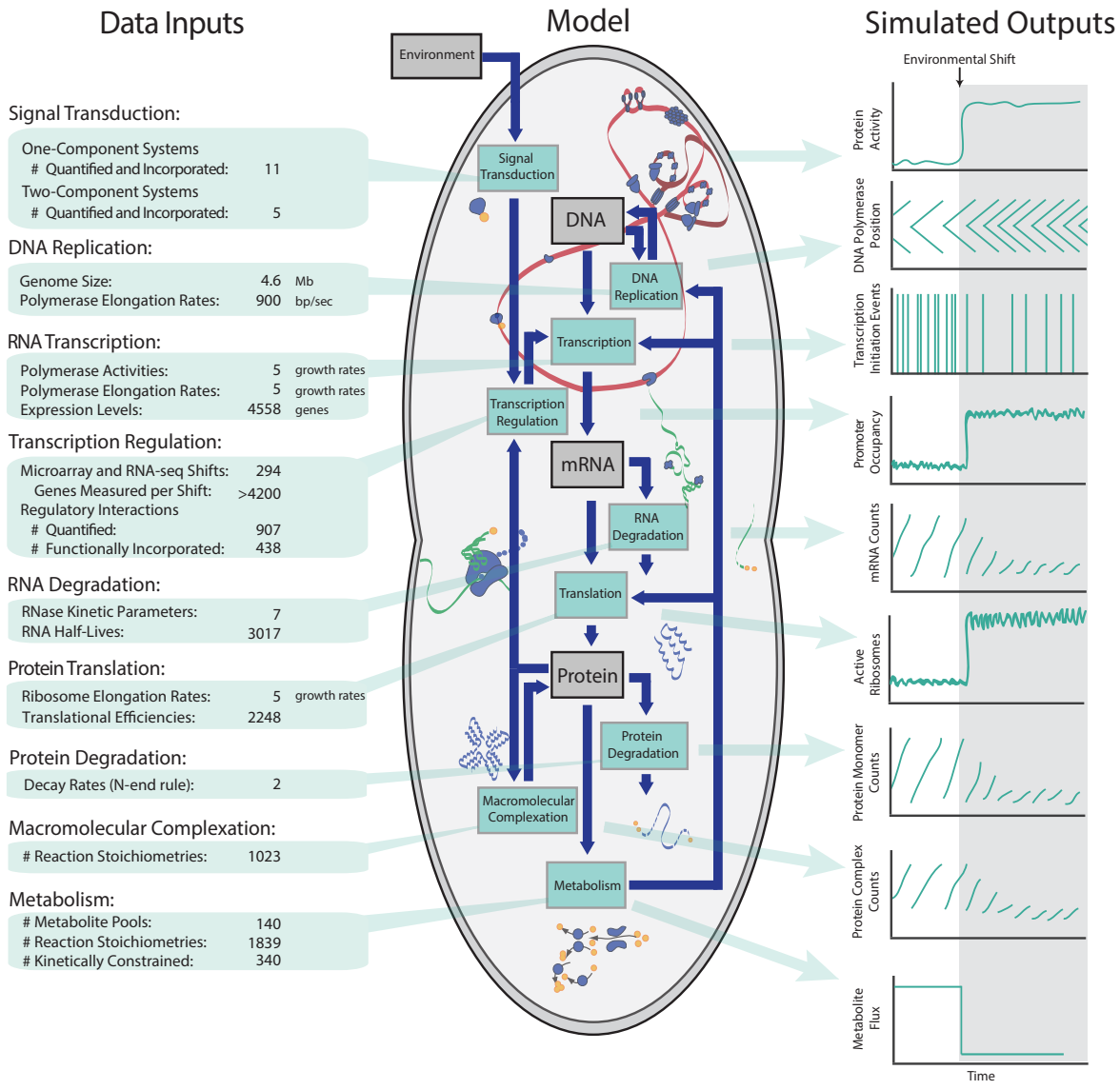


Figure 7.1: A computational framework synthesizes several decades' worth of heterogeneous data sets from a single organism into an internally consistent model. The model can simulate the dynamics of multiple aspects of physiology across three environmental conditions.

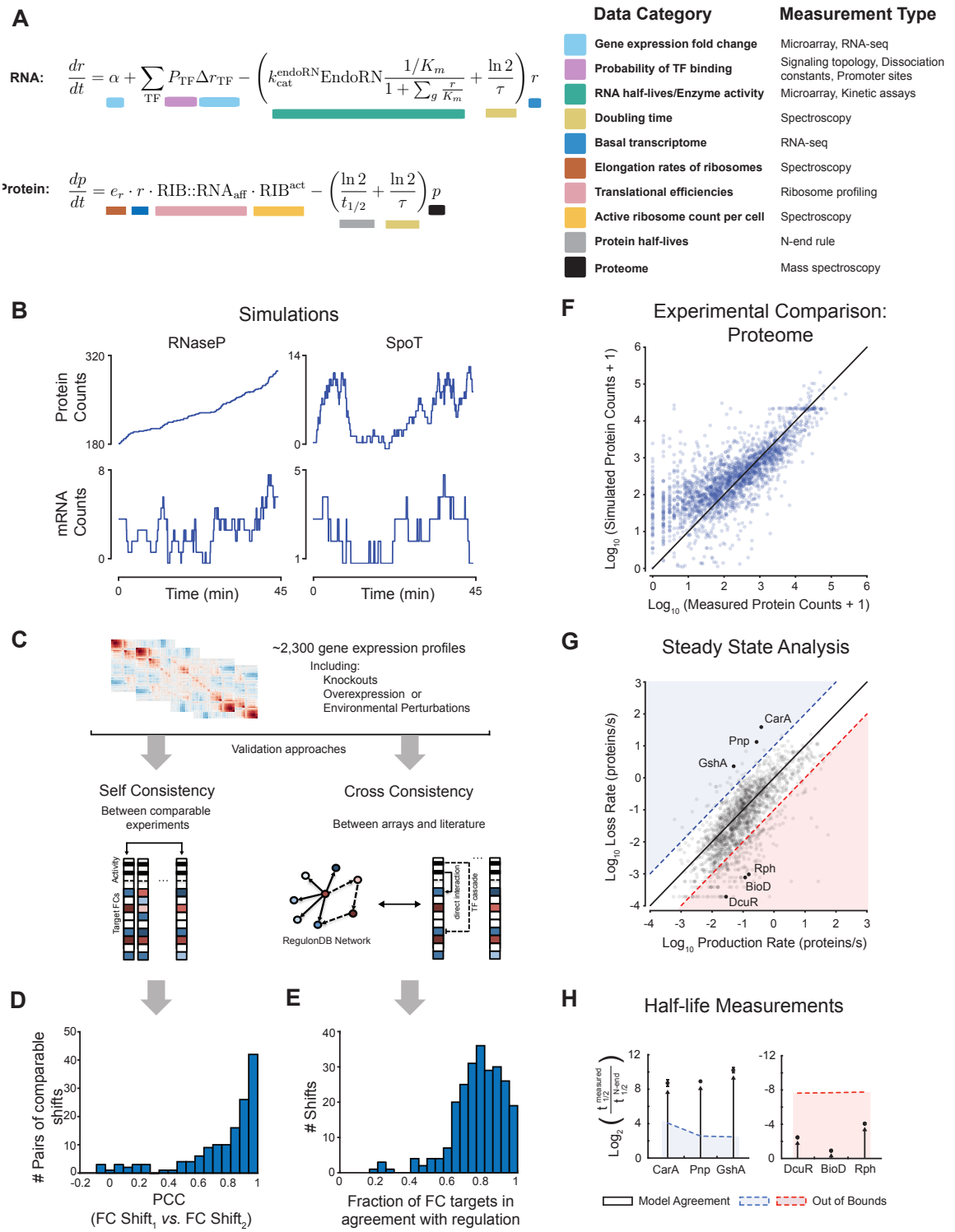

A
RNA:
Protein:
Data Category
Measurement Type
Gene expression fold change
Microarray, RNA-seq
Probability of TF binding
Signaling topology, Dissociation constants, Promoter sites
RNA half-lives/Enzyme activity
Microarray, Kinetic assays
Doubling time
Spectroscopy
Basal transcriptome
RNA-seq
Elongation rates of ribosomes
Spectroscopy
Translational efficiencies
Ribosome profiling
Active ribosome count per cell
Spectroscopy
Protein half-lives
N-end rule
Proteome
Mass spectroscopy
B
Simulations
RNaseP
SpoT
Protein Counts
mRNA Counts
Time (min)
C
~2,300 gene expression profiles
Including:
Knockouts
Overexpression or
Environmental Perturbations
Validation approaches
Self Consistency
Between comparable experiments
Cross Consistency
Between arrays and literature
RegulonDB Network
D
Pairs of comparable shifts
PCC
(FC Shift1 vs. FC Shift2)
E
Shifts
Fraction of FC targets in agreement with regulation
F
Experimental Comparison: Proteome
Log10 (Simulated Protein Counts + 1)
Log10 (Measured Protein Counts + 1)
G
Steady State Analysis
CarA
Pnp
GshA
Rph
BioD
DcuR
Log10 Loss Rate (proteins/s)
Log10 Production Rate (proteins/s)
H
Half-life Measurements
CarA Pnp GshA
DcuR BioD Rph
Model Agreement
Out of Bounds

Figure 7.2: Model-driven analysis and cross-validation of the data associated with Central Dogma-related processes. (A) Representative equations and data sources describing RNA and protein expression for all of the RNAs and proteins in *E. coli*. These equations were the starting point for our stochastic single-cell simulations. (B) Simulated dynamics of mRNA and protein expression for two genes. (C) A pipeline for assessing the consistency of a set of ˜2,300 RNA expression profiles in terms of self-consistency and cross-consistency. The self-consistency was determined by identifying 144 pairs of experiments that would be expected to have the same or similar results, and comparing the gene expression shifts observed in each, while the cross-consistency was found by comparing the set of expression profiles with the published literature on transcription as compiled in RegulonDB [142], where a given transcription factor has been determined to activate or repress transcription of a given target gene. (D) Histogram showing the Pearson Correlation coefficients between paired experimental outcomes. (E) Histogram showing the fraction of gene expression fold-changes shown to be in agreement with the published literature. (F) A comparison of simulation and experimental results [143] with regard to the number of proteins expressed per cell for each gene. (G) A comparison of calculated protein production rates against protein synthesis rates for each gene. Six outliers are highlighted because we determined their protein decay rates experimentally. (H) Comparison of the N-end rule to new measurements of protein half-lives. The experimental data are represented as a log2 ratio (where the N-end rule half-life is the denominator). The top three outliers from Panel G (in the blue region) had half-lives that were many-fold higher than indicated by the N-end rule; the other three (red) had half-lives that were lower than indicated. In all cases, the model prediction (i.e., white region beyond the dashed line) was closer to the experimentally determined value.

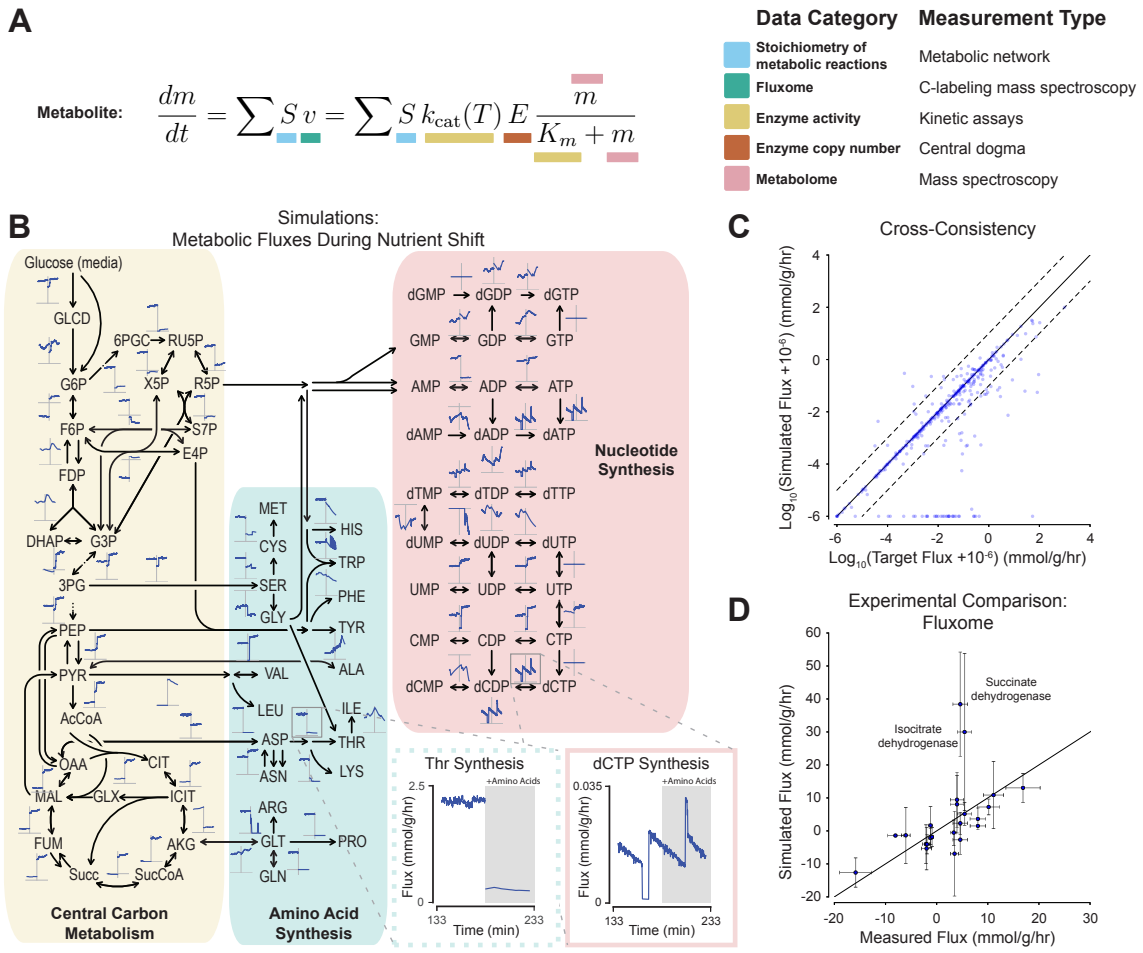


Figure 7.3: Model-driven analysis and cross-validation of the data associated with metabolic processes. (A) Representative equations and data sources describing concentrations for all of the metabolites in our model. These equations were the foundation of a linear optimization-based model. (B) Simulated dynamics of selected metabolic fluxes during a shift from minimal media to minimal media supplemented with amino acids. Two representative plots of flux dynamics are enlarged for illustrative purposes. (C) A comparison of the model's simulated flux results with flux values that were calculated from the kinetic data we curated. (D) A direct comparison of simulated flux values to experimental measurements [152]. Two outliers are highlighted for discussion in the main text. Error bars indicate standard deviation ($n = 3$ for measured flux and $n = 32$ for simulated flux).

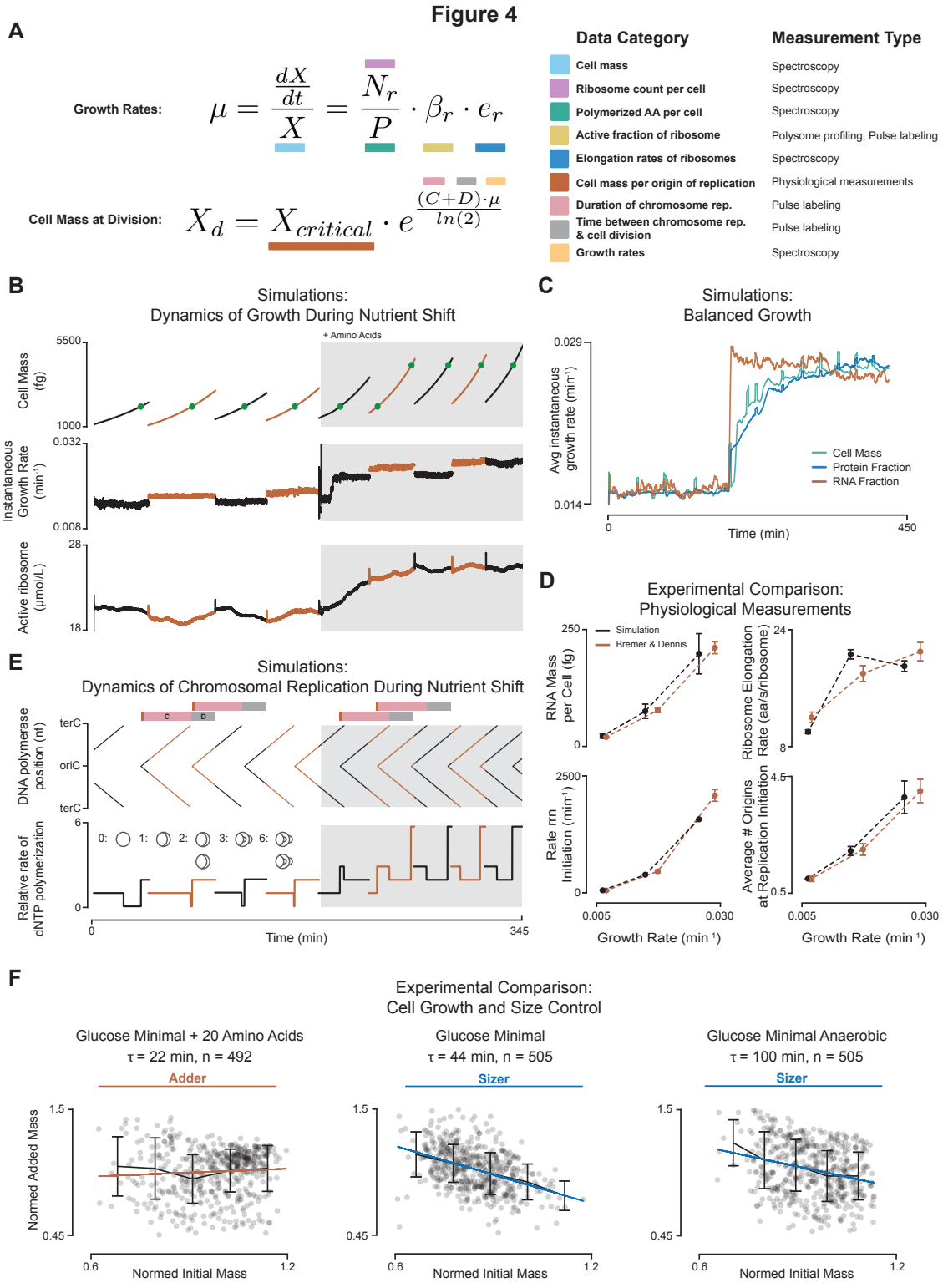


Figure 4
A
Growth Rates:
$\mu = \frac{\frac{dX}{dt}}{X} = \frac{N_r}{P} \cdot \beta_r \cdot e_r$
Cell Mass at Division:
$X_d = X_{critical} \cdot e^{\frac{(C+D)\cdot\mu}{ln(2)}}$
Data Category
Measurement Type
Cell mass
Spectroscopy
Ribosome count per cell
Spectroscopy
Polymerized AA per cell
Spectroscopy
Active fraction of ribosome
Polysome profiling, Pulse labeling
Elongation rates of ribosomes
Spectroscopy
Cell mass per origin of replication
Physiological measurements
Duration of chromosome rep.
Pulse labeling
Time between chromosome rep. & cell division
Pulse labeling
Growth rates
Spectroscopy
B
Simulations:
Dynamics of Growth During Nutrient Shift
+ Amino Acids
Cell Mass (fg)
5500
1000
Instantaneous Growth Rate (min⁻¹)
0.032
0.008
Active ribosome (μmol/L)
28
18
C
Simulations:
Balanced Growth
Avg instantaneous growth rate (min⁻¹)
0.029
0.014
Cell Mass
Protein Fraction
RNA Fraction
0
450
Time (min)
D
Experimental Comparison:
Physiological Measurements
Simulation
Bremer & Dennis
RNA Mass per Cell (fg)
250
0
Ribosome Elongation Rate (aa/s/ribosome)
24
8
Rate rrn Initiation (min⁻¹)
2500
0
Average # Origins at Replication Initiation
4.5
0.5
0.005
0.030
Growth Rate (min⁻¹)
E
Simulations:
Dynamics of Chromosomal Replication During Nutrient Shift
DNA polymerase position (nt)
terC
oriC
terC
C
D
Relative rate of dNTP polymerization
6
0
0: 1: 2: 3: 6:
0
345
Time (min)
F
Experimental Comparison:
Cell Growth and Size Control
Glucose Minimal + 20 Amino Acids
τ = 22 min, n = 492
Adder
Glucose Minimal
τ = 44 min, n = 505
Sizer
Glucose Minimal Anaerobic
τ = 100 min, n = 505
Sizer
Normed Added Mass
1.5
0.45
0.6
1.2
Normed Initial Mass

Figure 7.4: Model-driven analysis and cross-validation of the data associated with growth and DNA replication. (A) Representative equations and data sources describing key growth rate parameters and division requirements. These equations form the basis of the growth laws implemented in our simulations. (B) Simulated dynamics of cell mass, instantaneous growth rate, and active ribosome concentration over several generations (depicted using alternating colors of black and orange), both before and after supplementing a minimal medium with amino acids. The green circles indicate DNA replication initiation events. (C) Simulated average instantaneous growth rate of the overall cell mass as well as the RNA and protein fractions alone. After an environmental shift, the RNA mass fraction is the first to reach the new growth rate, but the other fractions also reach the new rate over time. (D) Comparison plots of cellular properties calculated from the simulations, together with their counterparts reported in the literature [154]. (E) DNA polymerase position and relative rate of dNTP polymerization, across generations and during a shift from minimal media to minimal media supplemented with amino acids. The pink and gray bars on top indicate the so-called C and D periods of replication (named for the processes of chromosome replication and cytokinesis, respectively) [157]; the inset drawings highlight the fact that the relative rate of dNTP polymerization can be thought of as analogous to the number of replication forks, as shown. (F) Comparison plots of the normed initial and added mass indicates that the simulations reproduce adder behavior in one condition, and sizer behavior in others, as would be expected [157–159].

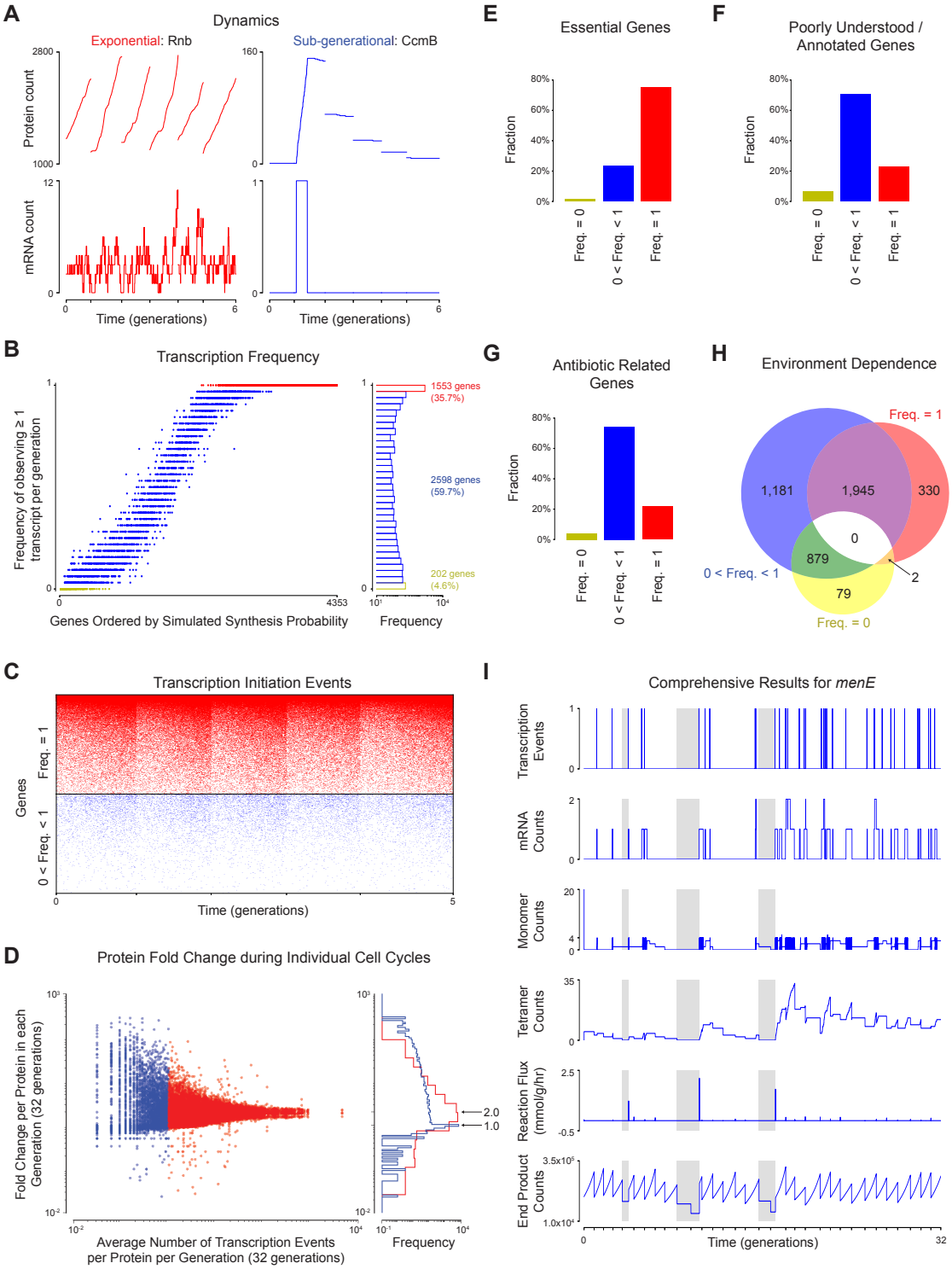

A
Dynamics
Exponential: Rnb
Sub-generational: CcmB
Protein count
mRNA count
Time (generations)
B
Transcription Frequency
Frequency of observing ≥ 1 transcript per generation
Genes Ordered by Simulated Synthesis Probability
Frequency
1553 genes (35.7%)
2598 genes (59.7%)
202 genes (4.6%)
C
Transcription Initiation Events
Genes
Freq. = 1
0 < Freq. < 1
Time (generations)
D
Protein Fold Change during Individual Cell Cycles
Fold Change per Protein in each Generation (32 generations)
Average Number of Transcription Events per Protein per Generation (32 generations)
Frequency
2.0
1.0
E
Essential Genes
Fraction
Freq. = 0
0 < Freq. < 1
Freq. = 1
F
Poorly Understood / Annotated Genes
G
Antibiotic Related Genes
H
Environment Dependence
Freq. = 1
1,181
1,945
330
0
879
2
0 < Freq. < 1
79
Freq. = 0
I
Comprehensive Results for menE
Transcription Events
mRNA Counts
Monomer Counts
Tetramer Counts
Reaction Flux (mmol/g/hr)
End Product Counts
Time (generations)

Figure 7.5: A large fraction of *E. coli* genes are expressed less than once per cell cycle, which has physiological consequences. (A) Simulations of mRNA and protein expression over multiple generations for genes that are expressed at high (left, in red) and low levels (right, in blue; note that colors are conserved to preserve meaning throughout the figure) transcriptional frequencies. Counts are shown for a representative six-generation long window, with an arbitrarily chosen zeroth starting generation. (B) Frequency of observing at least one gene transcript per generation over a 32-generation simulation. Histograms show that 1,553 genes are transcribed at least once per cell cycle (red), 202 genes are essentially never expressed in this environment (yellow), and the remaining 2,598 genes are transcribed with a frequency between zero and one (blue). (C) Simulated transcription initiation events for each gene during the first five generations. (D) A plot of the fold change in protein count for each individual protein species over every cell cycle. This shows that proteins that are always expressed in every cell cycle usually double in number (histogram peak at ˜2), whereas proteins which are expressed less than once per cell cycle often remain at the same protein count (histogram peak at ˜1), but sometimes experience a very large fold change. (E) Expression frequency analysis of essential genes, (F) poorly understood or annotated genes, and (G) genes related to antibiotic resistance and persistence. (H) Venn diagram indicating genes that are constitutively expressed in every cell cycle regardless of environmental conditions (red), genes that are always expressed less than once per cell cycle (blue), or not expressed (yellow), as well as genes that can be environmentally induced from one group to another (purple, orange, green). (I) A set of plots that indicate the transcription, translation, complexation and metabolic activity of the $MenE_4$ tetramer, which catalyzes a reaction responsible for producing menaquinone and demethylmenaquinone (represented in the final plot as a sum total). Each new generation is indicated with a tick mark along the x-axis; the gray areas highlight periods of time in which $MenE_4$ is not present in the cell.

Chapter 8

Conclusion

I've been told that good avenues of research always leave you with more questions than answers, more and more directions to pursue—more research to do.

By that metric, whole-cell modeling is an excellent avenue of research. Experimental, theoretical, and engineering challenges abound. While we've certainly made progress, by no means have we "solved the cell". In this chapter, I lay out opportunities that I think are worth pursuing, but I also caution the reader with the challenges that lay ahead.

8.1 Future Work

In building the core of an *E. coli* whole-cell model, I have become acutely aware of areas where our modeling efforts were hindered by a lack of data—data that we should be able to collect with existing measurement techniques. Acquiring this data will further our understanding of the organism and enable more detailed modeling efforts. I enumerate these areas in the *Experiments* sub-section below. Despite these hindrances, given that we have created the core of an *E. coli* whole-cell model, we can now expand the model into exciting new territory, which I outline in the *Modeling* sub-section that follows. Of course, existing sub-models can also be improved as new data becomes available in a manner amenable to integration. Finally, in the last sub-section, I outline the *Engineering* challenges.

8.1.1 Experiments

Gene Expression

While building the *E. coli* model of transcription, I was frustrated by my inability to find a well-annotated transcription unit structure. The annotation provided by Ecocyc [162], which (at the time of writing) boasts 3556 transcription units [116], does not have precisely defined transcription start

and stop sites for all transcription units (see the transcription unit definition at: [163]). In fact, while there are 3845 promoters listed [116], there are only 283 annotated terminator sites [117]. Using long-read sequencing technologies, it should be possible to properly annotate *E. coli*'s transcription unit structure.

In addition to annotating *E. coli*'s transcription unit structure, we would like to properly quantify *E. coli*'s transcriptome. Primarily for logistical reasons, my experiments in Chapter 5 used short (75 bp) reads which are not guaranteed to unambiguously map back to a reference transcriptome, particularly when nested transcripts are present. While long-read technologies might alleviate this issue, amplification biases during library preparation would need to be minimized [164]. Ideally, to properly understand transcriptional regulation (e.g., to see the effects of transcriptional attenuation), many of the classic microarray-based experiments that measured differential expression (e.g., those used to build the list of fold-change effects in Chapter 5) would be repeated and quantified using a long-read technology. This would likely be expensive, but it would give a more detailed view into regulation and could possibly suggest mechanism.

E. coli Composition Data

To reconstruct our *E. coli* cell at different growth rates, we relied heavily on data from Dennis and Bremer [165]. This data, collected a few decades ago, reports characteristics of an *E. coli* B strain rather than the currently in vogue K-12 MG1655 and BW25113 strains. My measured growth rates of MG1655 differed considerably from the reported B strain data. For example, in minimal media, Dennis and Bremer report a 44 min doubling time, whereas I observed an 80 minute doubling time for MG1655 (see Chapter 5). Unfortunately, due to lack of strain standardization over time, our current model represents an amalgam of all of these strains. While this is a considerably better situation than the *M. genitalium* model, which only has 27.5% of its parameters from its namesake organism (and a total of 55% of its parameters from any *Mycoplasma*) [81], ideally all of the information would emanate from a single strain. Reproducing Tables 2 and 3 in Dennis and Bremer's work [165] for K-12 MG1655 would be an excellent starting point.

Transcription factor binding

When building the model of transcription factor binding in *E. coli*, Javier Carrera and I curated affinity data for 32 transcription factor-DNA binding interactions. This represents only 15% of *E. coli*'s 208 transcription factors. Furthermore, in many cases, the reported affinity is for a consensus DNA sequence rather than a promoter-specific sequence. Ideally we would increase the coverage of this data, to both include more transcription factors and to include sequence-specific affinities. The MITOMI technology [120] being developed in Polly Fordyce's lab is a compelling candidate for this task.

Uncharacterized gene functionality

When we state that our goal is to construct a gene-complete whole-cell model of an organism, we implicitly mean that the model will incorporate the function of every well-annotated gene. When Nick Ruggero and I surveyed the list of *E. coli*'s genes and clustered them based on functionality, we found that approximately 30% of *E. coli*'s genes are not annotated in sufficient detail to model. In the *M. genitalium* whole-cell model, 410 out of the organism's 525 genes (78%) were functionally incorporated. In the Venter Institute's synthetically-constructed minimally viable cell, 32% of the 473 genes were unannotated yet essential [166]. Clearly, as evidenced by both modeling and synthetic biology efforts, our understanding of bacterial physiology is lacking.

How do we figure out the functions of these uncharacterized genes? In some preliminary work, a few of my labmates have observed differential expression of some of *E. coli*'s unannotated genes in response to bacteriophage exposure. It's possible that many of the unannotated genes respond to stimuli that we simply haven't rigorously explored yet. Admittedly, knowing that a gene is differentially expressed in response to a stimulus does not explain the gene's function, but it provides a starting point.

8.1.2 Modeling

Bacteriophage infection

Given our lab's—and my own—interest in phage infection, I think it would be fascinating to incorporate a phage model into a whole-cell model. Perhaps the most obvious choice from a logistical perspective would be to extend Elsa Birch's model of T7 infection that used a hybrid of ordinary differential equations and flux balance analysis [125]. More complicated would be the incorporation of a lambda phage model. Adam Arkin's work [167] may serve as a starting point, but considerable detail may need to be added in order to correctly predict outcomes of our lab's previously published phage infectivity screen [3]. Nonetheless, I have no doubt that comparing model predictions to experimental measurements would lead to novel and exciting insights into host-virus interaction.

Heterologous gene expression

One of my primary motivations for constructing a whole-cell model of *E. coli* was to use it as an engine for computer-aided design in synthetic biology. Alas, I have not had the opportunity to meaningfully pursue this goal as part of my doctoral work. Going forward, though, I hope to see the development of whole-cell computational methods that recapitulate the physiological effects of heterologous gene expression. My sense is that this work should start out with a limited number of well-characterized parts (e.g., plasmid origins of replication, promoter sequences, terminators, ribosome binding sites) whose effect on cell physiology (e.g., modulating effective rates of transcription, translation, degradation) is carefully quantified. Scaling this work to a larger library of parts

without overfitting to data may be challenging, but it would be incredibly powerful.

Spatial and structural considerations

Given our lab's close physical proximity to KC Huang's lab, we could leverage their expertise to incorporate models of cell structure in a whole-cell context. In the core *E. coli* model we currently lack an explicit model of cell shape and we assume a constant density to calculate cell volume. While this may be a valid assumption during exponential growth, it is likely a poor assumption in less favorable conditions such as osmotic shock. Cytokinesis is not currently modeled, either. Here we could adapt a spatial model of the Min system [168] to influence division site selection. Incorporating these spatial phenomena will be exciting, but I suspect—at least initially—they will make simulations considerably more expensive to run (and thus difficult to troubleshoot).

Collective behaviors

As part of the Allen Discovery Center for Cell Systems Modeling, we have the opportunity to collaborate with Shayne Peirce-Cottler's lab from the University of Virginia. They are experts in agent-based modeling, and, as such, bring expertise that would enable the modeling of collective behaviors from many individual whole-cell simulations. The two most obvious aspects of physiology to incorporate would be quorum sensing that coordinates gene expression across populations and, over longer time scales, horizontal gene transfer that spreads genes which confer a selective advantage (such as antibiotic resistance) through a population. Incorporating either or both of these aspects will likely require the acquisition of new measurements to parameterize these models in sufficient detail.

Pathogenesis and the treatment thereof

One application of whole-cell modeling that our lab explored briefly with the *M. genitalium* model was simulating the function of antibiotics. In *E. coli*, we could model this in considerably more detail because the organism is much easier to experimentally interrogate and thus validate. Still, this would be no small task as pathogenesis and antibiotic resistance are complex, multi-faceted issues. Before modeling a pathogenic strain of *E. coli* (or modeling a closely-related pathogen such as *Salmonella*), I think initial explorations should investigate modeling toxin-antitoxin systems and their link to persister formation (e.g., as put forth in [169]). This will necessitate the ability to model cells in stationary phase, which may be challenging with regards to reconciling parameters at a systems level. Modeling evolution of antibiotic resistance will also be challenging, but simulating the spread of resistance could leverage a horizontal gene transfer model discussed above in *Collective behaviors*. Thoroughly modeling all of these phenomena will be a massive undertaking, but their application to a pressing problem in human health is compelling.

8.1.3 Engineering

Data storage and accessibility

Whole-cell simulations have the ability to produce many terabytes of data. In our experience, querying this amount of data for analysis is non-trivial. For example, we tried storing data using the HDF5 file format, commonly used in scientific computing, but found that writing our raw data vectors straight to disk was significantly faster, even when retrieving them later for analysis. Unfortunately, these raw data vectors do not stand on their own and require knowledge of how the data was written to disk during a simulation. Jonathan Karr has developed WholeCellSimDB [8] to enable structured storage and querying of data from *M. genitalium* simulations. We would like a tool that, for instance, scales to *E. coli*-size data sets and more easily enables the querying of complex simulation attributes (such as the time-dependent position of individual ribosomes on an mRNA transcript—querying this post-simulation can be difficult). Because such a tool would ideally be part of the simulation test-debug cycle, run-time performance is critical and this may require significant amounts of (e.g., database) optimization.

Data visualization

There are a few ways to gain insight from a whole-cell model. The first way is to recognize "pain points" when constructing the model. What less-than-ideal assumptions do you have to make because of a lack of data? What do you do when data from different sources turns out to be internally inconsistent in your model? Often the answers to these questions suggest new experiments and opportunities for discovery. A second way is to explore simulation output. It is the only way to get a holistic view of everything going on simultaneously in (an approximation of) the cell—no experimental technique is capable of achieving that feat. Finding interdependencies and emergent phenomena in simulation output requires a massive amount of data analysis, and what better analytical engine to use for that purpose than the human visual cortex? Unfortunately, while existing tools such as WholeCellViz [9] do an excellent job communicating the scope of a whole-cell model, there is currently no good way to rapidly and easily explore whole-cell data for discovery purposes. As discussed in Chapter 3, new visual motifs are being developed for this purpose, and they will need to be coupled with advances in data storage and accessibility, outlined above, to enable rapid identification of previously-unidentified relationships.

Simulation run time

Many of the modeling endeavors described above will necessarily add complexity to the computations being performed in whole-cell simulations. While we improved simulation run time an order of magnitude over the *M. genitalium* simulations, more engineering can certainly be done. I would expect that with the rigorous application of software engineering principles and careful parallelization

efforts, simulations could be sped up another order of magnitude (to run on the order of a minute), and could be faster still on more exotic hardware platforms (e.g., GPUs, FPGAs). However, the hybrid nature of the model, and the accompanying communication overhead, makes these efforts non-trivial. Still, with the proper expertise and resource allocation, I believe that massive improvements in run time are possible. In addition to making the test-debug cycle more efficient, faster run times may be necessary to increase the accuracy and realism of whole-cell models, which I discuss below.

8.2 Major Challenges in Model Integration

While I've just outlined a number of exciting opportunities related to whole-cell modeling, I also want to give the reader a realistic outlook on the field. Early on in building the *E. coli* model, Nick Ruggero and I spent a lot of time trying to make very detailed sub-models that invoked many recently-reported single-cell biophysical measurements and models. By and large, we failed to implement those sub-models in the context of a whole-cell model. Why? The reason—as best I can articulate it—is that "bottom-up" models provide no guarantee of recapitulating system-level phenomena such as balanced cell growth at a desired growth rate. For example:

- Spatially-resolved whole-cell simulations used to study gene expression noise in the *lac* switch from the Luthey-Schulten group [73] do not show the cell growing over its life cycle (e.g., see their Video S1). Similarly, KC Huang's spatially-resolved simulations of Min oscillations, important in *E. coli* cell division, assume fixed-length cells [168]. While dismissing cell growth may have no bearing on the authors' respective conclusions, it is critical in the context of whole-cell modeling.

- Average off periods of RNA production reported by Ido Golding and Sunney Xie's group, respectively, are 37 min and 17 min [121,170]. This is long enough that, stochastically, a gene's transcript might not be produced for an entire generation (depending on the cell's doubling time). For an essential low-copy gene, or an essential gene whose protein product is rapidly turned over, this could create a problematic situation for the cell where it lacks a critical gene product. Probably due to technical difficulties (e.g., in obtaining a good signal-to-noise ratio), neither study focused on this class of genes.

- Models of transcription and transcriptional regulation derived from statistical mechanics considerations from Rob Phillip's group [171] implicitly assume the cell has a "healthy" number of RNA polymerases. To function in a whole-cell model, the (sub-)model of transcription would need to guarantee production of RNA polymerases (and other essential genes) at a "healthy" level, through carefully chosen parameters and/or regulatory motifs.

In a similar vein, we explored building sub-models that incorporated chromosome structure measured by HiC, building sub-models that incorporated RNA structural data and transcriptional

attenuation, and building models that explicitly incorporated ribosome stalling data from Jonathan Weissman's group [172] (we do include their concept of translational efficiency which makes use of ribosome stalling data). We were unable to get any of these novel sub-models to achieve balanced growth at the desired growth rate.

Based on these experiences, **in my opinion the biggest challenge in sub-model integration is parameter reconciliation**. In the Covert Lab, we colloquially refer to this problem as "parameter fitting" or just "fitting".

Due to the fact that simulations are still fairly computationally expensive (taking on the order of minutes rather than seconds to run), in order to produce a model that supports cell growth, we have a number of heuristic computational routines to reconcile parameters. For example, these routines will initialize "average" cells and ensure, assuming fixed elongation rates and sufficient amounts of monomeric building blocks, that cells have sufficient quantities of active RNA polymerases and ribosomes to achieve a desired growth rate. Unfortunately, if sub-models become too detailed/complicated and violate the admittedly naïve assumptions (e.g., of fixed elongation rates) used by the heuristic computational routines, the simulations fall apart. Figure 8.1 showcases the pathologies of some early un-fit simulations.

If it were possible to engineer simulations to run on the order of one second or less, we may be able to address the parameter reconciliation problem in a more principled manner that would enable the incorporation of detailed sub-models. We could compute the numerical gradient of a loss function that incentivizes balanced cell growth at a desired growth rate (as well as other system-level phenomena, such as recapitulating measured gene expression). Then, to compute (locally) optimal reconciled parameters, we could perform gradient descent, perhaps in a manner analogous to how deep neural networks are trained. Computing a numerical gradient would require individually perturbing thousands of model parameters and running simulations to assess their effects on the loss function. As a result, fast simulation run times would be necessary to see convergence on practical time scales of days to weeks.

Of course, it is entirely possible that such models may converge slowly or not at all. They may also be incredibly sensitive to the precise parameter values chosen. I speculate that there are a number of feedback loops—many of which haven't yet been discovered—within cells to maintain homeostasis in the face of stochasticity. These feedback loops would make the system robust to stochastic fluctuations and minor perturbations in parameter values. While whole-cell models can point to areas of cell physiology where such control loops may exist, careful experimental work, and perhaps even the development of new measurement technologies, will be needed to uncover, characterize, and quantify these circuits.

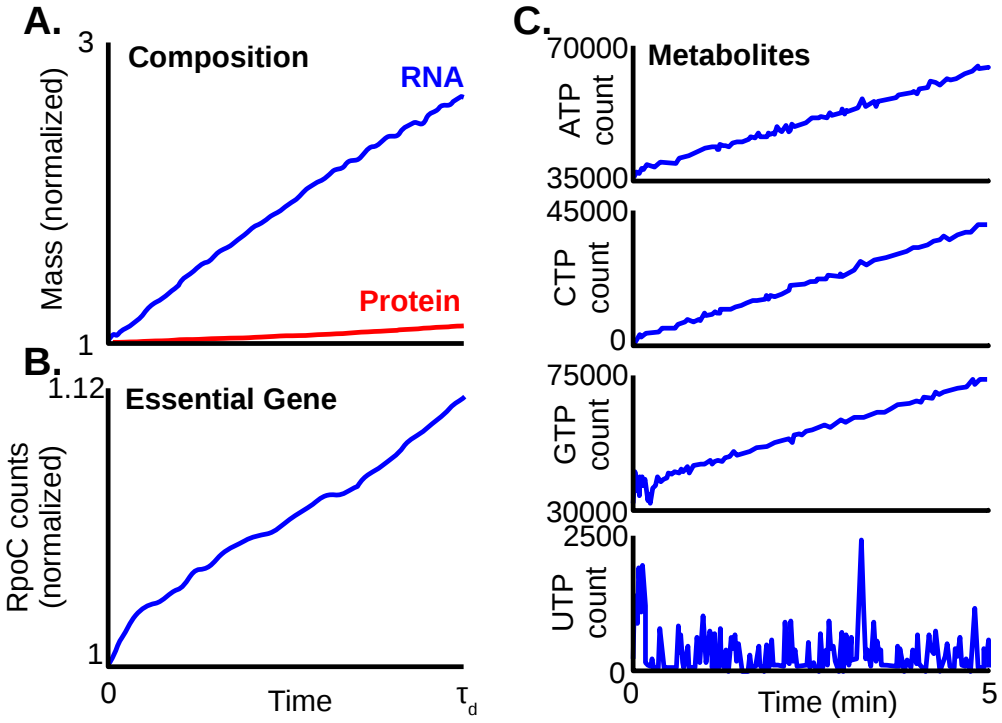


Figure 8.1: Pathologies of un-fit simulations.
(A) The RNA mass fraction nearly triples over the cell cycle while the protein fraction remains approximately constant, a result that is inconsistent with balanced growth.
(B) Following the behavior of the overall protein mass fraction in (A), the counts of RpoC, an essential subunit of RNA Polymerase, increase only 12% over the cell cycle. Over multiple generations, this essential enzyme would get diluted out.
(C) Results from an aborted simulation that show counts of ATP, CTP, and GTP accumulating excessively in only 5 minutes, while UTP remains limiting. This behavior is inconsistent with balanced cell growth.

Appendix A

Competing pathways control host resistance to virus via tRNA modification and programmed ribosomal frameshifting

Abstract

Viral infection depends on a complex interplay between host and viral factors. Here, we link host susceptibility to viral infection to a network encompassing sulfur metabolism, tRNA modification, competitive binding, and programmed ribosomal frameshifting (PRF). We first demonstrate that the iron-sulfur cluster biosynthesis pathway in *Escherichia coli* exerts a protective effect during lambda phage infection, while a tRNA thiolation pathway enhances viral infection. We show that $tRNA^{Lys}$ uridine 34 modification inhibits PRF to influence the ratio of lambda phage proteins gpG and gpGT. Computational modeling and experiments suggest that the role of the iron-sulfur cluster biosynthesis pathway in infection is indirect, via competitive binding of the shared sulfur donor IscS. Based on the universality of many key components of this network, in both the host and the virus, we anticipate that these findings may have broad relevance to understanding other infections, including viral infection of humans.

Chapter reproduced from: ND Maynard, DN Macklin, K Kirkegaard, MW Covert. "Competing pathways control host resistance to virus via tRNA modification and programmed ribosomal frameshifting". *Molecular Systems Biology.* 2012. 8: 567

A.1 Introduction

All viruses require host translational machinery in order to replicate. To make use of this machinery while avoiding detection, viruses have evolved a number of unconventional translational strategies [173], including internal ribosome entry sites [174], leaky scanning [175], ribosome shunting [176], reinitiation [177], and programmed ribosomal frameshifting (PRF) [178]. Many of these strategies are unique to viruses, and thus are potential antiviral targets [179]

During PRF, the translational machinery slips backward or forward by one nucleotide while decoding a so-called 'slippery sequence,' a heptanucleotide sequence of the form XXXYYYZ in the mRNA transcript (Figure A.1). Outside of PRF, frameshifting is a rare event, occurring fewer than 5×10^{-5} times per codon [180]. PRF can increase the occurrence of frameshifting by several orders of magnitude, depending on the slippery sequence itself as well as on the presence or absence of adjacent secondary structures that favor translational pausing and increase frameshift frequency [178]. In many instances of PRF, slippage causes the translational machinery to bypass a stop codon just downstream of the slippery sequence, resulting in the expression of an elongated protein [178]. Therefore, normal expression from these transcripts results in two products, a shorter product based on the initial translation frame plus a longer product, whose ratio depends on the frequency of slippage and is thought to have evolved to a value optimized for virus production [181, 182].

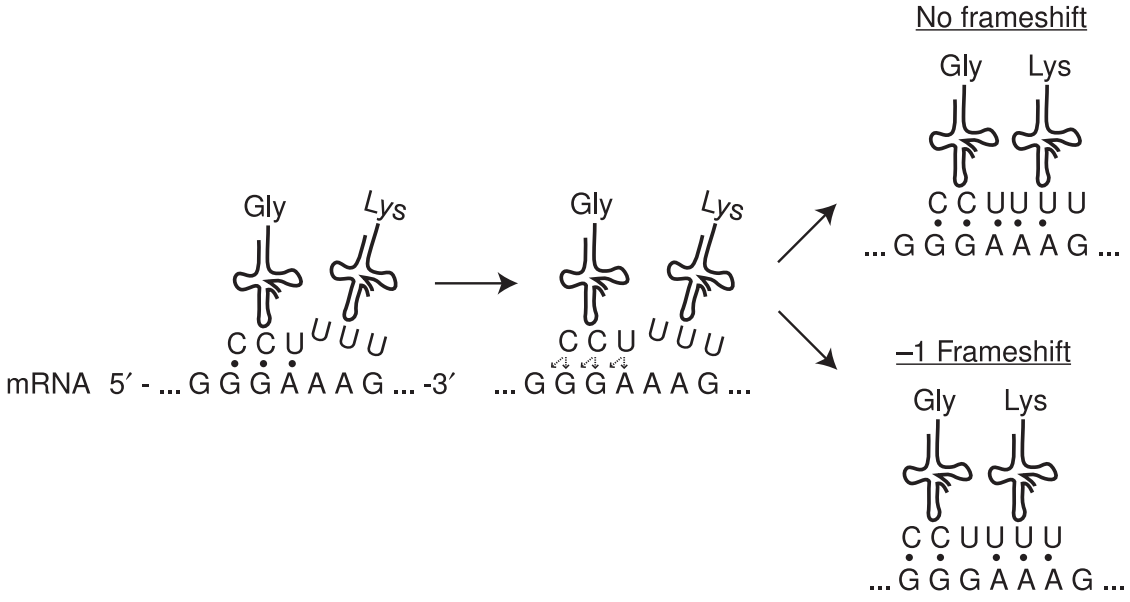


Figure A.1: Programmed ribosomal frameshifting. Schematic of a −1 programmed ribosomal frameshift. P-site tRNA slips in the −1 direction at the 'slippery sequence.'

PRF is an evolutionarily conserved phenomenon observed in a range of viruses, from bacteriophage to notable human pathogens (Table A.1) [183]. For example, in HIV, the Gag to Gag-Pol expression ratio is regulated by PRF and is necessary for the proper loading of reverse transcriptase into the new viral capsids; disturbing this ratio can inhibit viral replication [181, 182]. The role of PRF in the SARS life cycle is less clear, but is necessary for expression of its RNA-dependent RNA polymerase [184]. In *Escherichia coli* bacteriophage lambda, PRF regulates the expression of proteins gpG and gpGT [185]. As lambda phage genes G and T lie between its major tail gene (V) and its tape measure gene (H), they are thought to have a role in tail formation [186]. However, proteins gpG and gpGT do not make up any of the final tail structure [185], and thus may act as chaperones or scaffolds for gpH, assisting in the assembly of the tail shaft around the tape measure protein [186].

Reading-frame maintenance is a common function of tRNA modifications [187]. tRNAs contain upwards of 80 known modified nucleosides [188], and many of these are strongly conserved across species [189]. However, their loss is generally not lethal [190, 191]. In *E. coli*, four tRNA modifications require the addition of a thiol moiety and are made via two functionally distinct mechanisms [192]. The first mechanism requires proteins that contain iron-sulfur (Fe-S) clusters and includes 2-thiocytidine formation at position 32 and methylthio formation of ms2i6A at position 37 [193, 194]. The second mechanism is independent of Fe-S clusters and includes 4-thiouridine at position 8 and 2-thiouridine at position 34 [195]. For each of these modifications, sulfur is sequestered by *E. coli*'s primary cysteine desulfurase, IscS. Sulfur is then used for one of these four tRNA modifications or for Fe-S cluster biosynthesis. In addition to their roles in tRNA modification, Fe-S clusters are critical cofactors involved in many cellular processes including respiration, central metabolism, environmental sensing, RNA modification, DNA repair, and DNA replication [196].

We previously identified two pathways downstream of IscS in a screen to identify *E. coli* genes with a significant effect on lambda phage replication [3]. Specifically, lambda phage replication was inhibited following deletion of several members of the 2-thiouridine synthesis (TUS) pathway leading to 2-thiouridine modification of $tRNA^{Lys/Glu/Gln}$, a pathway independent of Fe-S cluster biosynthesis. Conversely, we found that knocking out members of the Fe-S cluster biosynthesis (ISC) pathway enhanced lambda replication.

The current investigation was therefore motivated by this overarching question: how do the host's TUS and ISC pathways control viral replication? We were particularly intrigued by the possibility

Virus	Slippery sequence	Region
HIV-1	UUUUUUA	*gag/gag-pol*
SARS-CoV	UUUAAAC	ORF*1a/1b*
Lambda phage	GGGAAAG	*G/GT*

Table A.1: Viruses dependent on programmed ribosomal frameshifting

that these pathways interact. Furthermore, the substantial evolutionary conservation of both host genes and viral PRF mechanism strongly suggested that our observations would have relevance beyond *E. coli* and lambda phage. Our current observations shed light on a complex network that extends from host metabolism and tRNA modification to viral translational regulation and finally to virion production. We find that 2-thiouridine hypomodification of $tRNA^{Lys/Glu/Gln}$ causes increased translational frameshifting, changing the ratio of the critical lambda phage proteins gpG and gpGT. We also show that IscU is linked to gpG and gpGT expression by competitive inhibition of the TUS pathway. Knocking out core members of the ISC pathway increases lambda phage replication by relieving competition for sulfur, allowing the TUS pathway to increase the rates of 2-thiouridine formation, thus reducing frameshifting. Targeting tRNA modifications to alter frameshifting rates, to which viral structural proteins are likely to be uniquely sensitive, presents a novel antiviral strategy.

A.2 Results

A.2.1 Viral replication can be slowed by deletion of TUS and accelerated by deletion of ISC genes

In a previous genome-wide screen [3], we found that plates growing TUS pathway deletion strains ($\Delta tusA$, $\Delta tusE$, and $\Delta mnmA$) infected with lambda phage produced plaques of an unusually small diameter compared with wild-type (WT) *E. coli*. In contrast, several ISC pathway deletion strains ($\Delta iscU$, $\Delta hscA$, and $\Delta hscB$) produced abnormally large diameter plaques compared with WT.

We investigated viral replication in these strains more thoroughly by culturing each strain in liquid media, in the presence and absence of virus. Without phage, WT *E. coli* grew in exponential phase for several hours, slowing and reaching stationary phase as available nutrients decreased (Figure A.2A, blue line). In the presence of lambda phage, the culture exhibited three phases (gray line). First, the culture underwent exponential growth similar to the uninfected culture. After ˜6 h in the WT strain, viral lysis overtook the culture and the culture began to clear (lytic phase). Lambda phage is a temperate phage with both lytic and lysogenic life cycles [197]; as a result, after a roughly 3-h lytic phase, lysogenized bacterial population growth took over the culture.

The time courses of lambda phage infection of the *tusA* and *iscU* knockout strains (the first members of the TUS and ISC pathways, respectively) differed significantly from that of WT. The $\Delta iscU$ culture entered the lytic phase very quickly (red curve, Figure A.2B), while the infected $\Delta tusA$ strain grew in exponential phase substantially longer than the WT strain, leading to a higher turbidity before the lytic phase occurred (green curve). We therefore decided to classify strains that entered the lytic phase earlier as more susceptible to viral infection (red) and those that entered the lytic phase later as less susceptible (green). This susceptibility classification was consistent with our earlier plaque assay experiments [3].

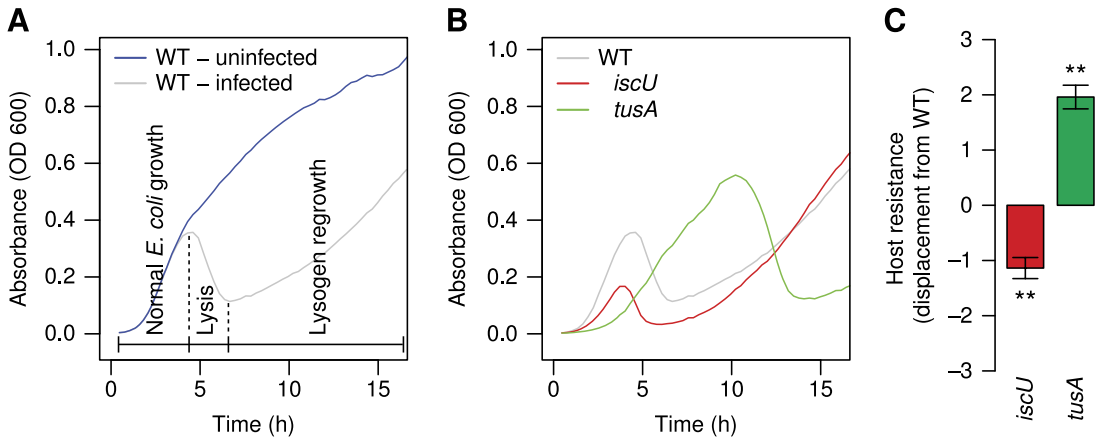


Figure A.2: Dynamics of lambda phage infection. (A) A typical trace of a WT *E. coli* culture infected with lambda phage. Several hours after infection, lysis begins to outpace *E. coli* growth and absorbance begins to decrease. Regrowth is due to the lambda lysogen population. (B) Infection dynamics of infected $\Delta iscU$ and $\Delta tusA$ cultures were compared. (C) The displacement from the WT growth curve. In (C), the bars indicate the 95% confidence interval (CI). For (A) and (B), absorbance was recorded over the course of 16 h for three biological replicates with four technical replicates for each biological replicate ($^{**}P < 0.01$, P-values were calculated using an independent two-sample t-test).

We then defined a metric to compare the resistance of different strains with infection with phage lambda. One useful way to summarize the information contained in a time course is to consider each experiment as a vector in n-dimensional space, where n is the number of time points taken. In earlier work, we found that it was useful to normalize these infection time course experiment vectors by the growth rate [3]. However, we have since found that bacterial growth and viral growth are tightly linked, and our attempts to numerically correct for or eliminate the effect of bacterial growth rate often distorted the raw data. Bacterial growth is therefore an important (but not sufficient) consideration in classifying our strains.

As a result, we determined our new metric simply by calculating the Euclidean distance between the knockout and WT strain time course vectors under infection conditions. Combining this distance with our susceptibility classification, we obtain a displacement vector that points in a negative direction for strains that are less resistant to virus and a positive direction for more resistant strains. The displacement vector, which we call the 'host resistance', enables us to compare the magnitude and direction of each strain relative with the WT strain and with each other. For example, the $\Delta iscU$ strain is significantly less resistant to viral infection than the WT strain, while the $\Delta tusA$ strain is more resistant (Figure A.2C).

A.2.2 Deletion mutants in Fe-S biosynthesis and tRNA thiolation exhibit altered viral infection dynamics

As both the TUS and ISC pathways are linked to sulfur metabolism through IscS, we wondered about the interplay between these two pathways and what might be the mechanistic basis for their effect on lambda replication. We began our investigation by examining the impact of other known ISC and TUS pathway members on lambda phage replication (Figure A.3A; see Figure B.1 for line plots of individual replicates). First, we investigated the effect on lambda phage replication of several genes associated with Fe-S cluster biosynthesis. We performed infection time courses of Δ*iscU* and other strains deficient in members of the *iscRSUA-hscBA-fdx-iscX* operon (*hscA*, *hscB*, *iscA*, *iscR*, *fdx*, and *iscX*). The slow growth of the *iscS* deletion strain made it impossible to assess infection dynamics and was not included in the analysis. The Δ*iscU*, Δ*hscA*, and Δ*hscB* strains all exhibited very similar infection dynamics, clearing the culture significantly before WT (Figure A.3A, top row; Figure A.3B). IscU is the primary scaffolding protein used for Fe-S cluster biosynthesis [198], and HscA and HscB are accessory proteins that help transfer FeS clusters to apoproteins [199] The Δ*iscA* strain exhibited increased replication efficiency, but less than the *hscA*, *hscB*, and *iscU* gene deletions (Figure A.3A and B). IscA assists in Fe-S cluster biosynthesis by shuttling iron to IscU [200].

Deletion strains for the remaining three genes in the operon, *iscR*, *fdx*, and *iscX*, displayed little if any change in viral infection dynamics from WT (Figure A.3A and B). The roles of Fdx and IscX in Fe-S cluster biosynthesis are not known [201], but IscR is the transcriptional repressor of the ISC operon and its active form includes an Fe-S cluster [202]. Interestingly, lambda phage replication is slightly but significantly hindered in the *iscR* knockout strain (Figure A.3A and B). Taken together, these observations suggest that ISC pathway members IscU, HscA, HscB, IscA, and possibly IscR have a more significant role in lambda phage infection than Fdx or IscX, and that their WT function helps to restrict phage growth.

We next determined how deletion of tRNA thiolation enzymes affected viral susceptibility of the host. We began with the Fe-S cluster-containing enzymes encoded by *miaB* and *ttcA*. MiaB is responsible for methylthio formation of ms2i6A at position 37 in $tRNA^{Phe}$ [194]; a sulfur donor for MiaB has not been identified. TtcA catalyzes the 2-thiocytidine modification at position 32 of $tRNA^{Arg}$(ICG), $tRNA^{Arg}$(CCG), $tRNA^{Arg}$(mnm5UCU), and $tRNA^{Ser}$(CGU) [193]. Δ*ttcA* strains displayed doubling times equivalent to WT *E. coli* (Figure A.3A, bottom), and little is known about the functional consequences of modification. Strains lacking either of these genes showed no effect on lambda infection dynamics (Figure A.3A, bottom; Figure A.3C). In contrast, deletion strains for the two Fe-S cluster-independent tRNA thiolation enzymes ThiI and MnmA displayed a significant increase in viral resistance versus the WT strain (Figure A.3A, bottom; Figure A.3C). Sulfur is transferred to ThiI through direct binding of IscS [203], and ThiI then transfers the sulfur to s4U8 of $tRNA^{Phe}$. Lambda replication in the Δ*thiI* strain was mildly but significantly inhibited (Figure A.3A, bottom; Figure A.3C).

The Δ*mnmA* strain exhibited the most dramatic effect on viral infection of all the strains associated with tRNA thiolation. MnmA catalyzes 2-thiouridination at position 34 of several tRNAs [204]. Sulfur is supplied to MnmA through the TUS pathway, beginning with the direct binding of IscS by TusA [195]. The Δ*mnmA* strain underwent infection dynamics (Figure A.3A, bottom; Figure A.3C) very similar to the upstream TUS pathway gene deletion strains we characterized in a previous study (Maynard et al, 2010), including Δ*tusA*, Δ*tusBCD*, and Δ*tusE*. All members of the TUS pathway had a strong, significant, and similar effect on lambda replication (Figure A.3A and C), leading us to conclude that Fe-S cluster-independent tRNA thiolation in *E. coli* exerts a significant effect on lambda phage replication through the TUS pathway and, to a lesser degree, ThiI.

A.2.3 Changes in tRNA modification and frameshifting propensity inhibit viral replication

Based on the preceding data and a previous observation of increased frameshifting in Δ*mnmA* strains [187], we hypothesized that reduced lambda replication in the TUS knockouts was due to increased frameshifting during lambda protein synthesis as a result of the decreased tRNA modification. To test this hypothesis, we generated infection curves for several deletion mutants that have been shown to affect frameshifting by loss of tRNA modification, including *tgt*, *truA*, *miaA*, *thiI*, and mnmE [187]. We infected deletion strains for these genes with lambda phage and determined their susceptibility to infection (Figure A.4; see Figure B.2 for line plots of individual replicates). Δ*tgt*, Δ*miaA*, and Δ*truA* exhibited no significant affect on lambda phage replication, suggesting that the tRNA modifications from these enzymes have no role in lambda phage replication. The remaining knockout strains displayed significant changes in lambda phage's ability to infect the host, including Δ*thiI*, Δ*mnmA* (Figures A.3 and A.4), and Δ*mnmE* (Figure A.4). Intriguingly, MnmE performs a methylaminomethyl modification on carbon 5 of the same tRNA species and nucleoside targeted by MnmA [205], suggesting that the antiviral properties of the TUS pathway knockouts may be due to specific tRNA modifications instead of to general translational fidelity issues.

A.2.4 Codon usage bias does not underlie differential infection dynamics

Host-preferred codon usage is common among many viruses and their hosts [206,207]. We wondered whether the lambda genome used the codons whose tRNAs are modified by the *mnmA*, and *mnmE* gene products preferentially with respect to *E. coli*. The tRNA species modified by MnmA and MnmE decode the lysine, glutamic acid, and glutamine codons [204]; we examined the codon usage for lysine, glutamic acid, and glutamine in lambda phage and in E. coli for synonymous codon usage bias. Lambda phage encodes lysine 748 times (65% AAA and 35% AAG), glutamic acid 900 times (57% GAA and 43% GAG), and glutamine 576 times (20% CAA and 80% CAG), with similar biases in *E. coli* (Table A.2). This similarity in codon usage does not support the hypothesis that lambda

phage protein synthesis is any more impaired in strains harboring hypomodified 2-thiouridine tRNAs than its host.

Amino acid	Codon	*E. coli*: fraction (number)	Lambda phage: fraction (number)
Lysine	AAA	0.76 (46116)	0.65 (486)
	AAG	0.24 (14174)	0.35 (262)
Glutamic acid	GAA	0.69 (54431)	0.57 (515)
	GAG	0.31 (24629)	0.43 (385)
Glutamine	CAA	0.35 (21121)	0.20 (117)
	CAG	0.65 (39836)	0.80 (459)

Table A.2: Codon usage in *E. coli* and lambda phage

A.2.5 Thiolation of $tRNA^{Lys}$ (UUU) and frameshifting via PRF are linked through a genetic network

Since we thought it unlikely that lambda phage replication was being significantly affected by general reading frame maintenance in the TUS pathway knockout mutants, we turned instead to specific tRNA modifications that could explain our observations of an increase in viral resistance. In particular, our attention was drawn to PRF as a specific instance of frameshifting that could be particularly sensitive to changes in frameshifting frequency.

In lambda phage, PRF occurs during translation of the lambda phage *GT* region, which encodes two possible protein products, either gpG or gpGT; expression of gpGT depends on ribosomal frameshifting at the slippery sequence GGGTTTG [185]. This sequence encodes the dipeptide Gly-Lys in both the 0 open reading frame (ORF; GGT-TTG) and the −1 ORF (GGG-TTT). In the 0 ORF, which is maintained 96% of the time, gpG is expressed (Figure A.5A); and in the remaining cases, the ribosomal machinery reaches the slippery sequence and slips back one base pair. As a result, the stop codon for gpG, immediately downstream of the slippery sequence, is bypassed and the larger gpGT product is produced. Interestingly, both the lysine codons TTG (0 ORF) and TTT (−1 ORF) are decoded by the same $tRNA^{Lys}$ (UUU), a target of MnmA and MnmE. We therefore wondered whether TUS pathway gene deletion strains would exhibit an altered rate of frameshifting and consequently a change in the ratio of gpG to gpGT.

To monitor gpG and gpGT expression in *E. coli*, we created a plasmid for the inducible synthesis of these proteins (pBAD-λGT; Figure A.5B). We PCR amplified the *GT* region of the lambda genome and introduced this region behind an arabinose-inducible promoter, to control expression of the transcript, and the Xpress Epitope C-terminal tag, to facilitate detection by immunoblotting. We then transformed WT, $\Delta tusA$, $\Delta iscU$, and $\Delta tusA\Delta iscU$ strains with pBAD-λGT. Deletion of the iscU gene led to abrogation of frameshifting, whereas frameshifting frequency increased in both the $\Delta tusA$ and the $\Delta tusA\Delta iscU$ strains relative to WT (Figure A.5C). We also tested frameshifting

frequency for additional strains ($\Delta mnmA$, $\Delta thiI$, $\Delta miaB$, $\Delta ttcA$, $\Delta hscA$, $\Delta hscB$, Δfdx, $\Delta iscA$, and $\Delta iscR$) and found frameshifting levels to be consistent with infection dynamics (see Figure B.3). We interpret these results to signify that the decreased thiolation of $tRNA^{Lys}$ (UUU) in the TUS pathway knockout strains leads to an increase in frameshifting at the slippery sequence.

A.2.6 Competitive binding of IscU and TusA for IscS binding integrates sulfur metabolism and infection dynamics

Surprisingly, the $\Delta iscU$ strain exhibited markedly reduced gpGT levels relative to the WT strain (Figure A.5C), strongly indicating that both IscU and TusA act on lambda phage replication by altering the gpG:gpGT ratio. Furthermore, our observation that the *miaB* and *ttcA* deletion strains exhibited normal lambda phage replication (Figure A.3) suggested that IscU does not act directly through thiolation of specific tRNA modifications. How, then, does IscU influence viral replication independent of direct tRNA modification?

One possible answer is that IscU competes with TusA to bind IscS. Recent X-ray crystallography studies showed that IscU and TusA bind IscS at distinct but adjacent locations [208]. Superposition of the IscU and TusA structures indicated a spatial overlap between the volume occupied by TusA and IscU when bound to IscS. Furthermore, three-way pull-down experiments demonstrated that IscS may bind either TusA or IscU individually, but not both at the same time [208]. To further explore the possibility that thiolation of $tRNA^{Lys}$ (UUU) is affected by competition between TusA and IscU for sulfur transfer from IscS, we constructed a mathematical model of IscU and TusA binding to IscS (Figure A.6A). Our model consisted of a set of equilibrium and mass conservation relationships. In this case, the model parameters had either been previously measured or were not difficult to estimate based on existing data [41,198]. Our model predicts the relative amount of U34 thiolation from the TUS pathway for a specified amount of IscU.

We combined this protein binding model with our previous mathematical model of *E. coli* infection by lambda phage [3]. Our earlier model predicts the *E. coli* time course dynamics of lambda phage infection based on three parameters: the fraction of infections that are lytic instead of lysogenic, the rate of infection, and the burst size (how many functional progeny are released upon lysis). To integrate these two models, we related U34 thiolation (the output of the protein binding model) to the burst size (input for our infection model). We were thus able to explore the effect of changing the IscU or TusA concentration on the production of functional phage and the resulting infection time course. We first tested the integrated model's ability to predict the outcome of deleting *iscU* or *tusA*, and in both cases the model was able to capture the infection time course (Figure A.6B-E; see Figure B.4 for line plots of individual replicates). We also considered titration of IscU experimentally as well as computationally by introducing an IPTG-inducible *iscU* gene construct into a $\Delta iscU$ strain and determining the infection time courses under various IPTG concentrations. The effect of IscU titration was predicted well by our competitive inhibition model (Figure A.6F and G;

see Figure B.4 for line plots of individual replicates).

Finally, we considered the case of a double deletion of *tusA* and *iscU*. The competitive inhibition model suggested that the double mutant would behave in a manner similar to a Δ*tusA* strain under infection conditions (Figure A.6H). Again, this prediction agreed with the experimental time course (Figure A.6I and J; see Figure B.4 for line plots of individual replicates), as well as our experimental observation that the ratio of gpGT to gpG in the double mutant was similar to that observed in the Δ*tusA* strain, but not in the Δ*iscU* strain (Figure A.5C).

As a control, we compared the predictions of the competitive inhibition model with those of the alternative hypothesis that IscU and TusA influence viral infection independently. This hypothesis required some relatively simple modifications in the equation*s and parameters of our competitive binding model (Materials and methods). Our independent effect model was able to capture the results of *tusA* deletion, *iscU* deletion, and IscU titration as accurately as the competitive inhibition model (data not shown). However, the independent effect model was not able to predict the behavior of the double knockout strain (Figure A.6K).

Our competitive inhibition model predicts that the removal of IscU, one of TusA's primary competitors for sulfur, will cause an increase in the fraction of 2-thiolation modified U34 $tRNA^{Lys}$ (UUU) relative to BW25113. To directly measure the fraction of thiolated $tRNA^{Lys}$ (UUU), we performed an [(N-Acryloylamino)phenyl] mercuric chloride (APM) northern blot (Figure A.7), which specifically retards the migration of the thiolated tRNA fraction [209]. We found a small fraction of hypomodified $tRNA^{Lys}$ in BW25113, and as predicted by the competitive inhibition model, loss of IscU in the Δ*iscU* strain decreased this fraction even further. We therefore conclude that our competitive inhibition model, but not our independent effect model, is sufficient to explain the effect of TUS and ISC pathway knockouts on lambda phage replication.

A.3 Discussion

Taken together, our observations suggest the existence of a complex network that links sulfur metabolism, tRNA modification, competitive binding, and PRF-based regulation of viral protein expression with host susceptibility to viral infection (Figure A.8). TusA obtains sulfur from the cysteine desulfurase IscS, and passes it along the TUS pathway for eventual modification of $tRNA^{Lys}$ (UUU) at U34. The fraction of $tRNA^{Lys}$ (UUU) that undergoes this modification influences the frequency of PRF in lambda phage due to its role in decoding part of the slippery sequence within its *GT* region. Although lambda phage requires both gpG and gpGT for normal replication, the gpG:gpGT ratio must be kept high. Deletion of any member of the TUS pathway prevents modification of $tRNA^{Lys}$ (UUU), and subsequently increases frameshifting, both decreasing the gpG:gpGT ratio and lambda phage production.

In the presence of HscA, HscB, and (less critically) IscA, IscU binds IscS more strongly than

TusA [208] and is therefore able to outcompete TusA for sulfur. When IscU, HscA, HscB, or IscA is removed from the system, TusA obtains more sulfur than normal, leading to hypermodification of $tRNA^{Lys}$ (UUU), a decrease in frameshifting, and increases in the gpG:gpGT ratio and lambda phage production. Previous investigations in *B. subtilis* reported that disrupting the function of IscU affects several levels of tRNA modifications, including an increase in 2-thiolation of U34 [210]; our observations suggest that in the case of $tRNA^{Lys}$ (UUU), this effect is a result of an increase of sulfur flux through the TUS pathway (Figures A.5C, A.6K, and A.7). Removal of the transcriptional repressor IscR leads to an increase in protein production from the ISC operon [202], reducing the amount of sulfur available to TusA. This competition effect depends on an intact TUS pathway, which is why the $\Delta tusA\Delta iscU$ deletion mutant expresses a phenotype similar to the $\Delta tusA$ strain.

This model fits our observations well, but unresolved issues remain, most notably the role of ThiI in viral infection. ThiI would seem to be another possible competitor for sulfur from IscS (Figure A.2A). However, the $\Delta thiI$ strain exhibited decreased lambda replication, albeit somewhat less strongly than the TUS pathway deletion strains. One clue to this mystery may reside with the CyaY protein, which is thought to be an iron donor for Fe-S cluster assembly, forming a ternary complex with IscU and IscS [208]. The binding site for ThiI on IscS overlaps substantially with CyaY [208], and it is conceivable that in the absence of ThiI, CyaY is able to interact more efficiently with IscU, either stabilizing the IscS-IscU-CyaY complex or assisting sulfur transfer to IscU. In either case, deleting *thiI* may increase IscU's ability to compete with TusA for sulfur.

Our findings point to several novel antiviral strategies and targets. Although this work was performed using *E. coli* and phage lambda, many of the key elements of the network we describe are conserved in other systems, in both in host and the virus. Both Fe-S clusters and tRNA modifications are ubiquitous in cells [188, 196] and PRF is a common viral translational strategy [178]. Indeed, frameshifting has already been identified as a potential target in HIV, but efforts to target the *gag-pol* mRNA secondary structure have been challenging [211]. Our results point to upstream (e.g., tRNA modifying enzymes) and even indirect targets (e.g., Fe-S cluster biosynthesis pathways), which can just as potently alter infection and in some cases are tunable (Figure A.6G). We therefore anticipate that our observations and inferences may have broad relevance to understanding other infections, including viral infection of humans.

A.4 Materials and Methods

A.4.1 Strains

The *E. coli* strains used in this study were all taken or derived from the Keio Collection, a comprehensive library of single gene deletion strains [4]. BW25113 is the parental strain used for the construction of the Keio Collection. Lambda phage was obtained from the ATCC (23724-B2). The $\Delta tusBCD$ strain was generated previously [3], and the $\Delta tusA\Delta iscU$ double knockout strain

was constructed according to the Datsenko-Wanner method [212]. Primer sequences are available upon request. The lambda *GT* reporter strain was constructed using a pBAD/His A plasmid (Invitrogen, Cat# V430-01). The lambda phage *GT* region was amplified via PCR (primers: 5'-cacattctcgagatgttcctgaaaaccgaatca-3' and 5'-gtgtaaaagcttgccataccggactcctcct-3'). The amplified *GT* region and pBAD/His A plasmid were cut with *Xho*I and *Hind*III. BW25113, Δ*tusA*, Δ*iscU*, and Δ*tusA*Δ*iscU* strains were transformed with the arabinose-inducible lambda phage *GT* reporter construct (Figure A.5B). A chemically inducible Δ*iscU* strain was created by transforming Δ*iscU* with pCA24N-*iscU* isolated from the ASKA collection [213].

A.4.2 Cell culture quantification

We monitored infection dynamics using an incubated plate reader (Perkin-Elmer Victor3, 2030-0030) under growth conditions described previously [3]. Exponentially growing cells were normalized to 0.1 OD 600 nm. Fifteen microliters of 0.1 OD *E. coli* and 15 μl of ˜10^4 plaque forming units (p.f.u.)/ml lambda stock were added to 170 μl of LB medium in 96-well plates, representing a multiplicity of infection of ˜$2 \cdot 10^{-4}$ p.f.u./bacteria. Each plate contained four technical replicates of infected and uninfected samples. The plate reader measured the absorbance of the culture approximately every 17 min for 16 h. Each plate contained a sample of BW25113 for between-plate normalization. Three independent biological replicates were performed for each strain.

A.4.3 Comparative metrics for time courses

Time courses were compared based on a useful displacement metric that includes a direction in addition to a distance. The distance was determined by calculating the Euclidean distance between the two time course vectors; the Euclidean distance between the recorded absorbances for each culture were calculated at each time point, and these distances were summed for all time points. For our purposes, the direction could only hold two values, ± 1. The positive value was associated with strains exhibiting increased resistance to viral infection, and the negative value was associated with increased susceptibility to viral infection. The direction was determined by the difference in the time it took a culture to reach its peak absorbance. For example, time courses with longer times to peak compared with BW25113 were considered to be displaced in a positive direction from BW25113. The displacement was then calculated as (direction) $\cdot$ (distance).

The Euclidean distance from BW25113 was measured for each biological replicate. Statistical significance was determined using a two-sample t-test comparing the Euclidian distances calculated for each biological replicate and its distance from BW25113 for each plate containing the strain being compared.

A.4.4 *E. coli* and lambda phage codon usage

Lambda phage ORFs were extracted from NCBI Reference Sequence NC_001416.1 (Enterobacteria phage lambda, complete genome) and codon usage was tallied.

Codon usage for WT E. coli (W3110) was determined from the Codon Usage Database (`http://www.kazusa.or.jp/codon/`).

A.4.5 Assaying lambda PRF

pBAD-λGT-transformed cells were cultured overnight at 37°C in LB and the appropriate antiobiotics. The cultures were diluted 1:50 into fresh LB the following morning and placed back in the shaker for 1 h. The culture was then split into 1 ml aliquots of arabinose. Arabinose induction was performed for 2 h at 0.2, 0.02, and 0.002% arabinose, after which the samples were spun down in a table-top centrifuge at ˜20000 g for 30 s. The media was aspirated and the pellets were frozen at -80°C. The samples were run on a 12% SDS-PAGE gel for 1.5 h at 80 V in Tris/glycine/SDS buffer. The protein was transferred onto a polyvinylidene fluoride membrane for 1 h at 100 V at room temperature. The membrane was washed for 30 s in TBST, then blocked with 5% milk blocking buffer for 1 h. The primary antibody (Anti-express antibody, Invitrogen) was diluted 1:5000 and incubated with the membrane blot overnight at 4°C. The membrane was washed five times for 10 min with TBST, followed by a 1-h incubation with a 1:5000 dilution of the secondary antibody (goat anti-mouse IgG horseradish peroxidase). The membrane was washed five times for 5 min with TBST. Luminol reagent was used for chemiluminescence imaging (Santa Cruz Biotechnology, sc2048).

A.4.6 tRNA enrichment

BW25113 and $\Delta iscU$ were cultured overnight and diluted 1:100 to 20 ml in the morning. When the cultures reached ˜0.5 OD 600 nm, the samples were placed on ice. The cells were pelleted at 5000 r.p.m. for 5 min at 4°C and the supernatant was decanted. tRNA enrichment was performed using standard techniques. Briefly, the cell pellets were resuspended in 300 μl cold sodium acetate (0.3 M, pH4.5, 10 mM EDTA) and transferred to eppendorf tubes. The samples were snap frozen in liquid nitrogen. After thawing, 300 μl phenol:chloroform (pH 4.7) was added to the samples and vortexed four times for 30 s bursts with 1 min incubation on ice in between. The samples were centrifuged for 15 min at 4°C in a table-top centrifuge at maximum speed. The top layer (aqueous phase) was transferred to a new tube and the acid phenol extraction procedure was repeated. After the aqueous phase was transferred to a new tube, the RNA was ethanol precipitated by adding three volumes of cold 100% ethanol and spinning at maximum speed for 25 min at 4°C. The RNA pellet was resuspended in 60 μl cold sodium acetate (0.3 M, pH 4.5) and precipitated again by adding 400 μl 100% ethanol and spinning at maximum speed for 25 min at 4°C. The supernatant was decanted

and all traces of ethanol were removed. The RNA pellet was air dried on ice then resuspended in 100 μl cold sodium acetate (10 mM, pH 4.5).

A.4.7 APM northern blot

The APM northern blot was performed according to the protocol specified by [209]. Briefly, two 8 M Urea, 5% acrylamide gels were cast in 0.5 × TBE buffer. One gel contained 200 μl of 1 mg/ml APM (kindly supplied by Gabor Igloi) for 10 ml of gel. The gels were prerun for 30 min at 180 V in 0.5 × TBE. Approximately 2.5 μg of RNA (for each well) was then mixed with 2 × loading buffer and heated to 65°C for 3 min before being cooled on ice for 1 min. The gels were run for ˜45 min at 180 V. The RNA was transferred onto a nylon membrane using a semi-dry system for 1 h at 250 mA. The transfer buffer contained 0.5 × TBE supplemented with 10 mM β-mercaptoethanol. The RNA was then UV crosslinked to the nylon membrane. T4 Polynucleotide Kinase (PNK) was used to radiolabel 25 pmol of 32P ATP to 25 pmol of $tRNA^{Lys}$(UUU) oligonucleotide probe for each blot. The oligonucleotide sequence used to probe $tRNA^{Lys}$(UUU) was 5'-TGGGTCGTGCAGGATTCGAA-3'. Qiagen's Qiaquick Nucleotide Exchange Kit was used to remove free 32P ATP. The blots were incubated in 25 ml of prehybridization buffer (10 × Denhardt′s, 6 × SSC, 0.1% SDS) at 42°C for 1 h. The labeled probe was heated to 95°C for 2-3 min before being added to the prehybridaztion buffer. The blots were hybridized at 42°C overnight. The next day, the blots were washed with high salt buffer (5 × SSC, 0.1% SDS) at 37°C for 20 min twice, followed by two washes with low salt buffer (1 × SSC, 0.1% SDS) at 42°C for 20 min. After the final wash, the blots were exposed to a phosphorscreen for 10 min.

A.4.8 Mathematical modeling

We constructed a competitive inhibition model based on the following assumed mechanisms: (1) the cysteine desulfurase IscS binds sulfur, converting cysteine to alanine in the process; (2) thiolated IscS presents sulfur to the apo forms of TusA and IscU; (3) TusA and IscU bind thiolated IscS with their respective affinities and obtain the sulfur; (4) TusA and IscU transfer the sulfur to their downstream pathways, represented as sTUS and sISC, respectively. The source code for our model can be downloaded from `https://simtk.org/home/lambda-tus`.

The reactions are as follows:

$$\mathrm{Cys} + \mathrm{IscS} \underset{k_{-1}}{\overset{k_1}{\rightleftharpoons}} \mathrm{Cys}\cdot\mathrm{IscS} \underset{0}{\overset{k_2}{\rightleftharpoons}} \mathrm{IscS}\cdot\mathrm{S} + \mathrm{Ala}$$

$$\mathrm{IscS}\cdot\mathrm{S} + \mathrm{IscU} \underset{k_{-3}}{\overset{k_3}{\rightleftharpoons}} \mathrm{IscS}\cdot\mathrm{S}\cdot\mathrm{IscU} \underset{0}{\overset{k_4}{\rightleftharpoons}} \mathrm{IscS} + \mathrm{IscU}\cdot\mathrm{S}$$

$$\text{IscS} \cdot \text{S} + \text{TusA} \underset{k_{-5}}{\overset{k_5}{\rightleftharpoons}} \text{IscS} \cdot \text{S} \cdot \text{TusA} \underset{0}{\overset{k_6}{\rightleftharpoons}} \text{IscS} + \text{TusA} \cdot \text{S}$$

$$\text{IscU} \cdot \text{S} \underset{0}{\overset{k_7}{\rightleftharpoons}} \text{IscU} + \text{sISC}$$

$$\text{TusA} \cdot \text{S} \underset{0}{\overset{k_8}{\rightleftharpoons}} \text{TusA} + \text{sTUS}$$

From these reactions we derived the following set of ordinary differential equations and mass conservation relationships:

$$\begin{aligned}
\frac{d[\text{Cys}]}{dt} &= k_{-1}[\text{Cys} \cdot \text{IscS}] - k_1[\text{Cys}][\text{IscS}] \\
\frac{d[\text{Cys} \cdot \text{IscS}]}{dt} &= k_1[\text{Cys}][\text{IscS}] - (k_{-1} + k_2)[\text{Cys} \cdot \text{IscS}] \\
\frac{d[\text{Ala}]}{dt} &= -\left(\frac{d[\text{Cys}]}{dt} + \frac{d[\text{Cys} \cdot \text{IscS}]}{dt}\right) \\
\frac{d[\text{IscS} \cdot \text{S}]}{dt} &= k_2[\text{Cys} \cdot \text{IscS}] + k_{-3}[\text{IscS} \cdot \text{S} \cdot \text{IscU}] + k_{-5}[\text{IscS} \cdot \text{S} \cdot \text{TusA}] \\
&\quad - k_3[\text{IscS} \cdot \text{S}][\text{IscU}] - k_5[\text{IscS} \cdot \text{S}][\text{TusA}] \\
\frac{d[\text{IscU}]}{dt} &= k_{-3}[\text{IscS} \cdot \text{S} \cdot \text{IscU}] + k_7[\text{IscU} \cdot \text{S}] - k_3[\text{IscS} \cdot \text{S}][\text{IscU}] \\
\frac{d[\text{IscS} \cdot \text{S} \cdot \text{IscU}]}{dt} &= k_3[\text{IscS} \cdot \text{S}][\text{IscU}] - (k_{-3} + k_4)[\text{IscS} \cdot \text{S} \cdot \text{IscU}] \\
\frac{d[\text{IscU} \cdot \text{S}]}{dt} &= -\left(\frac{d[\text{IscU}]}{dt} + \frac{d[\text{IscS} \cdot \text{S} \cdot \text{IscU}]}{dt}\right) \\
\frac{d[\text{TusA}]}{dt} &= k_{-5}[\text{IscS} \cdot \text{S} \cdot \text{TusA}] + k_8[\text{TusA} \cdot \text{S}] - k_5[\text{IscS} \cdot \text{S}][\text{TusA}] \\
\frac{d[\text{IscS} \cdot \text{S} \cdot \text{TusA}]}{dt} &= k_5[\text{IscS} \cdot \text{S}][\text{TusA}] - (k_{-5} + k_6)[\text{IscS} \cdot \text{S} \cdot \text{TusA}] \\
\frac{d[\text{TusA} \cdot \text{S}]}{dt} &= -\left(\frac{d[\text{TusA}]}{dt} + \frac{d[\text{IscS} \cdot \text{S} \cdot \text{TusA}]}{dt}\right) \\
\frac{d[\text{IscS}]}{dt} &= -\left(\frac{d[\text{Cys} \cdot \text{IscS}]}{dt} + \frac{d[\text{IscS} \cdot \text{S}]}{dt} + \frac{d[\text{IscS} \cdot \text{S} \cdot \text{IscU}]}{dt} + \frac{d[\text{IscS} \cdot \text{S} \cdot \text{TusA}]}{dt}\right) \\
\frac{d[\text{sISC}]}{dt} &= k_7[\text{IscU} \cdot \text{S}] \\
\frac{d[\text{sTUS}]}{dt} &= k_8[\text{TusA} \cdot \text{S}]
\end{aligned}$$

$$[\text{IscS}] + [\text{Cys} \cdot \text{IscS}] + [\text{IscS} \cdot \text{S}] + [\text{IscS} \cdot \text{S} \cdot \text{IscU}] + [\text{IscS} \cdot \text{S} \cdot \text{TusA}] = [\text{IscS}]_{\text{TOTAL}}$$

$$[\text{IscU}] + [\text{IscU} \cdot \text{S}] + [\text{IscS} \cdot \text{S} \cdot \text{IscU}] = [\text{IscU}]_{\text{TOTAL}}$$

$$[\text{TusA}] + [\text{TusA} \cdot \text{S}] + [\text{IscS} \cdot \text{S} \cdot \text{TusA}] = [\text{TusA}]_{\text{TOTAL}}$$

$$[\text{Cys}] + [\text{IscS} \cdot \text{Cys}] + [\text{Ala}] = [\text{Cys}]_{\text{TOTAL}}$$

Many of the parameter values were taken from the literature. [41] reported values of ˜800 IscS proteins per cell and 450 IscU proteins per cell. Approximating the *E. coli* volume as 10^{-15} L, we obtained concentrations of 1.37 and 0.77 μM for IscS and IscU, respectively. Since TusA amounts were not reported, we estimated its concentration to be similar to but less than IscU 0.5 μM. An in-vitro study reported the K_d values of IscS and IscU to be 2 μM, and the k_{cat} (k_2 in the above equations) to be 8.5 per minute [198]. IscU displaced TusA in a three-way pull-down assay [208], from which we hypothesized that the K_d of TusA-IscU was 20% more than that of IscU-IscS. The cysteine concentration was chosen to be equal to the IscS concentration.

The forward rates of reaction were assigned as follows: $k1 = k3 = k5 = k7 = k8 = 10^5$, $k4 = k6 = k^{-3}$. The reverse reaction rates were obtained by multiplying the corresponding forward rate by the appropriate dissociation constant. Gene deletion conditions were simulated by setting the pertinent concentration values to zero. IscU titrations were modeled by setting its concentration to 50, 100, and 150% of the WT amount.

Steady-state values of the sTUS parameter relative to WT were used to scale the amplification factor b in our previous non-dimensionalized predator-prey model [3]. We simulated infection dynamics with the following parameters: $k_i = 0.25$, $f = 0.93$, $b = 10 \cdot \frac{\text{sTUS+sTUS}_{\text{WT}}}{\text{sTUS}_{\text{WT}}}$, $\mu^* = 0.3$, $K^* = 0.4$, $k_s = 0$

The independent effect model hypothesizes that there is no competition between IscU and TusA for sulfur. Thus, they each receive their own (equally sized) pool of thiolated IscS ($\text{IscS}_U \cdot \text{S}$ belongs to IscU and $\text{IscS}_A \cdot \text{S}$ belongs to TusA). The dissociation constants of both IscU and TusA with their respective pools of IscS were set to 2.7 μM. The concentrations of IscU and TusA were set to the same values as in the competitive model. The amount of thiolated IscS, however, was increased 100-fold to prevent it from being limiting. The reactions are as follows:

$$\text{IscU} + \text{IscS}_U \cdot \text{S} \underset{k_{1r}}{\overset{k_{1f}}{\rightleftharpoons}} \text{IscU} \cdot \text{S} + \text{IscS}_U$$

$$\text{TusA} + \text{IscS}_A \cdot \text{S} \underset{k_{2r}}{\overset{k_{2f}}{\rightleftharpoons}} \text{TusA} \cdot \text{S} + \text{IscS}_A$$

$$\text{IscU} \cdot \text{S} \underset{0}{\overset{k_{3f}}{\rightleftharpoons}} \text{IscU} + \text{sISC}$$

$$\text{TusA} \cdot \text{S} \underset{0}{\overset{k_{4f}}{\rightleftharpoons}} \text{TusA} + \text{sTUS}$$

From these, we derived the following set of ordinary differential equations and constraints:

$$\begin{aligned}
\frac{d[\text{IscU}]}{dt} &= k_{1r}[\text{IscU} \cdot \text{S}][\text{IscS}_U] + k_3 f[\text{IscU} \cdot \text{S}] - k_{1f}[\text{IscU}][\text{IscS}_U \cdot \text{S}] \\
\frac{d[\text{TusA}]}{dt} &= k_{2r}[\text{TusA} \cdot \text{S}][\text{IscS}_A] + k_4 f[\text{TusA} \cdot \text{S}] - k_{2f}[\text{TusA}][\text{IscS}_A \cdot \text{S}] \\
\frac{d[\text{IscU} \cdot \text{S}]}{dt} &= -\left(\frac{d[\text{IscU}]}{dt}\right) \\
\frac{d[\text{TusA} \cdot \text{S}]}{dt} &= -\left(\frac{d[\text{TusA}]}{dt}\right) \\
\frac{d[\text{IscS}_U \cdot \text{S}]}{dt} &= k_{1r}[\text{IscU} \cdot \text{S}][\text{IscS}_U] - k_{1f}[\text{IscU}][\text{IscS}_U \cdot \text{S}] \\
\frac{d[\text{IscS}_A \cdot \text{S}]}{dt} &= k_{2r}[\text{TusA} \cdot \text{S}][\text{IscS}_A] - k_{2f}[\text{TusA}][\text{IscS}_A \cdot \text{S}] \\
\frac{d[\text{IscS}_U]}{dt} &= -\left(\frac{d[\text{IscS}_U \cdot \text{S}]}{dt}\right) \\
\frac{d[\text{IscS}_A]}{dt} &= -\left(\frac{d[\text{IscS}_A \cdot \text{S}]}{dt}\right) \\
\frac{d[\text{sISC}]}{dt} &= k_{3f}[\text{IscU} \cdot \text{S}] \\
\frac{d[\text{sTUS}]}{dt} &= k_{4f}[\text{TusA} \cdot \text{S}]
\end{aligned}$$

$$[\text{IscU}] + [\text{IscU} \cdot \text{S}] = [\text{IscU}]_{\text{TOTAL}}$$

$$[\text{TusA}] + [\text{TusA} \cdot \text{S}] = [\text{TusA}]_{\text{TOTAL}}$$

$$[\text{IscS}_U \cdot \text{S}] + [\text{IscS}_U] = [\text{IscS}_U \cdot \text{S}]_{\text{TOTAL}}$$

$$[\text{IscS}_A \cdot \text{S}] + [\text{IscS}_A] = [\text{IscS}_A \cdot \text{S}]_{\text{TOTAL}}$$

$$[\text{IscS}_U \cdot \text{S}]_{\text{TOTAL}} = [\text{IscS}_A \cdot \text{S}]_{\text{TOTAL}}$$

The kinetic parameters were specified as follows: $k_{1f} = k_{2f} = k_{3f} = k_{4f} = 10^5$. Non-zero reverse rates were set to the product of the forward rate and the appropriate dissociation constant.

Steady-state values of the sTUS and sISC parameters relative to wild-type were used to scale the amplification factor b. Parameters for the predator-prey equation were: $k_i = 0.25$, $f = 0.93$, $b = 10 \cdot \frac{\text{sTUS}+\text{sTUS}_{\text{WT}}}{\text{sTUS}_{\text{WT}}} - 50 \cdot \frac{\text{sISC}-\text{sISC}_{\text{WT}}}{\text{sISC}_{\text{WT}}}$, $\mu^* = 0.3$, $K^* = 0.4$, $k_s = 0$. We see that in both models, the WT burst rate is 20 and the *tusA* knockout burst rate is 10. Table B.1 describes the parameters used in the two models.

The standard ode23s solver in Matlab was used for solving the sulfur transfer equations. The ode45 solver was used to solve the predator-prey equations.

A.5 Acknowledgments

We gratefully acknowledge R Young as well as members of the Covert and Kirkegaard laboratories for helpful discussions; T Vora for editing the text; G Igloi for kindly supplying the APM; O'Reilly Science Art for assistance with the figures; and the funding this work through an NIH Director's Pioneer Award (1DP1OD006413) to MWC, a postdoctoral fellowship (1F32GM090545) to NDM, and a Benchmark Stanford Graduate Fellowship to DNM.

Author contributions: NDM, KK, and MWC conceived and designed the experiments. NDM and DNM performed the experiments. NDM, DNM, and MWC analyzed the data. NDM and DNM contributed reagents/materials/analysis tools. NDM, DNM, KK, and MWC wrote the paper.

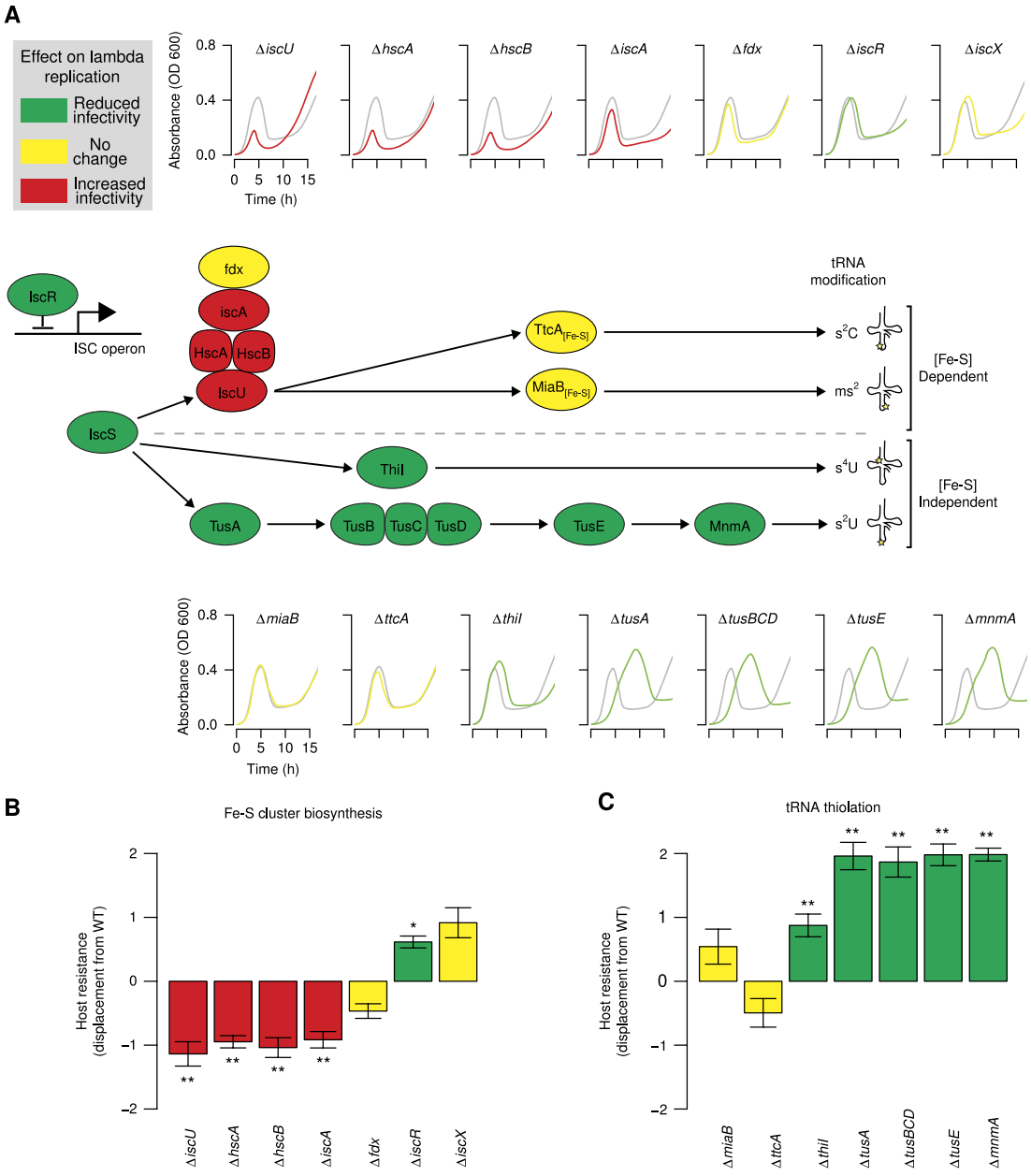


Figure A.3: The effects of Fe-S cluster and tRNA thiolation deletions on lambda phage infection dynamics. (A) Cell-culture dynamics following infection overlay the uninfected control data (WT alone, gray). Small arrows in the cartoon indicate sulfur transfer and the large bent arrow indicates expression of the ISC operon. (B) Displacement from the WT strain for ISC operon members ($^*P < 0.05$, $^{**}P < 0.01$, bars indicate 95% CI, P-values were calculated using independent two-sample t-test). (C) Comparison of the displacement from the WT strain for tRNA thiolation pathway members. Source data is available for this figure in Appendix B.

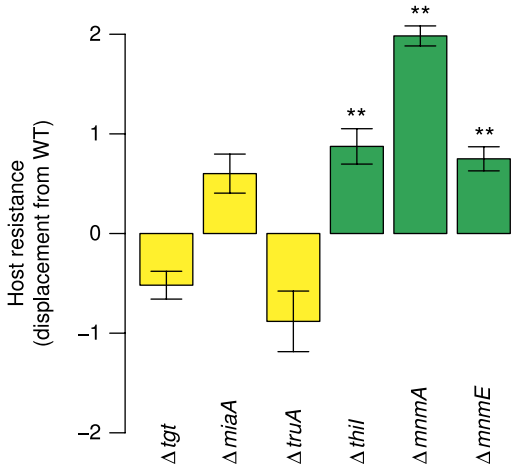


Figure A.4: Displacement vector comparisons for mutants in *E. coli* genes known to affect frameshifting. Comparison of the displacement from the WT for deletion mutants in genes known to affect frameshifting (**$P < 0.01$, bars indicate 95% CI, P-values were calculated using independent two-sample t-test). Green, reduced infectivity; yellow, no effect on infectivity. Source data is available for this figure in Appendix B.

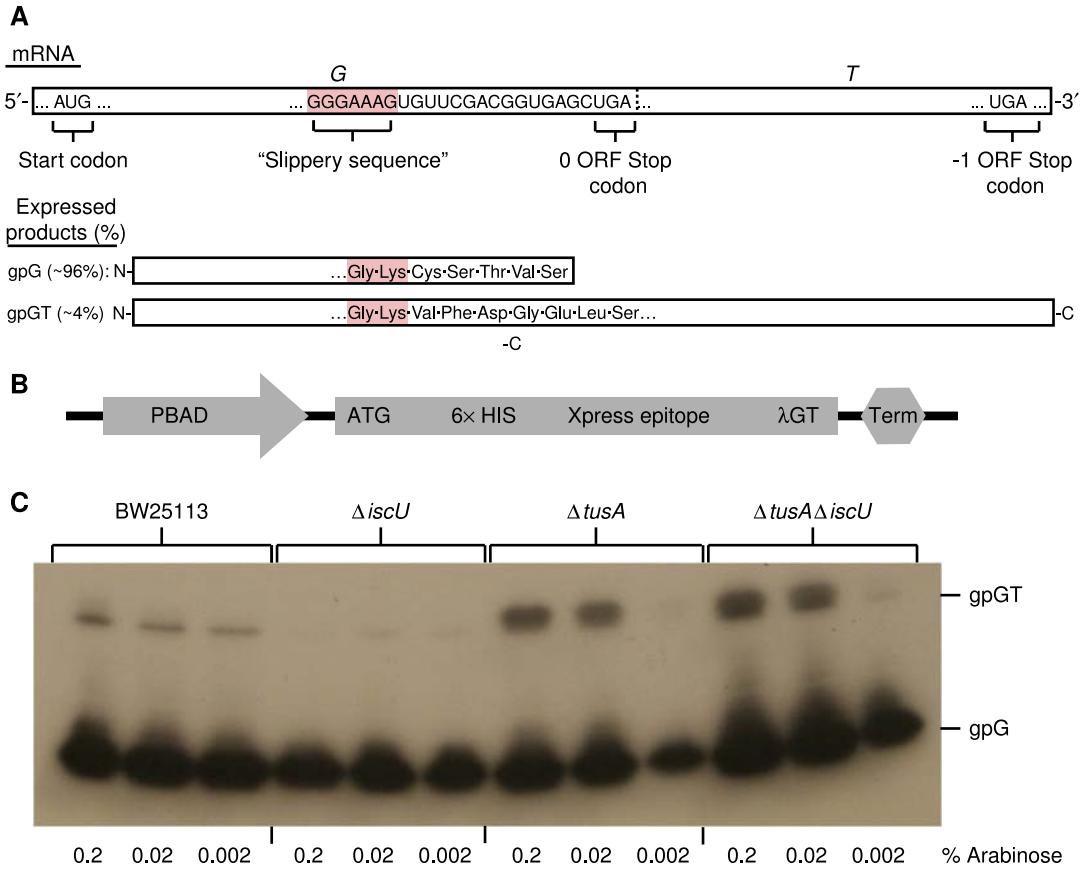


Figure A.5: Lambda phage proteins gpG and frameshift product gpGT. (A) Schematic of lambda phage's frameshifting region in G and T. (B) Schematic of the arabinose-inducible region of the pBAD-λGT vector. (C) Immunoblotting of pBAD-λGT in BW25113, $\Delta iscU$, $\Delta tusA$, and $\Delta tusA\Delta iscU$ strains. We induced expression of the pBAD-λGT transcript with 0.2, 0.02, or 0.002% l-arabinose for 2 h and assessed the gpG and gpGT protein levels by immunoblotting against the Xpress Epitope tag. We found a decrease in gpGT levels in $\Delta iscU$ and an increase in $\Delta iscU$ and $\Delta tusA\Delta iscU$.

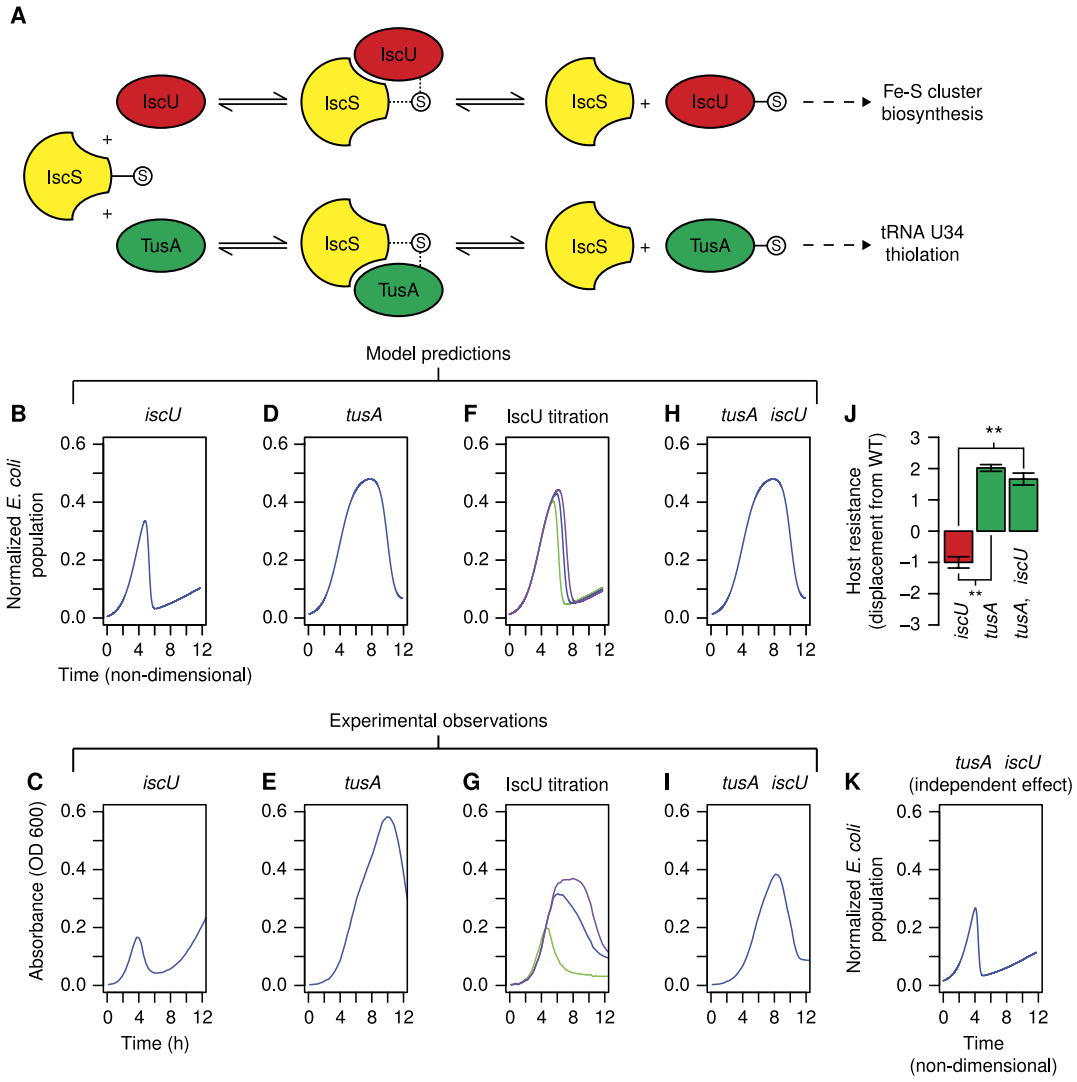


Figure A.6: Competitive binding of IscU and TusA to IscS: theory and experiment. (A) A schematic of our competitive binding model. IscS may bind and transfer sulfur to either IscU or TusA, leading to production of Fe-S cluster biosynthesis or thiolated tRNA, respectively. (B-I) Comparison of model predictions (top row) and experimental observations (bottom row) of infection time courses. The genetic perturbations include the deletion mutant strains $\Delta iscU$ (B, C) and $\Delta tusA$ (D, E), titration of expression from $iscU$ (F, G) and the double deletion mutant strain $\Delta tusA\Delta iscU$ (H, I). (J) The host resistance metric calculated for the $\Delta tusA$, $\Delta iscU$, and $\Delta tusA\Delta iscU$ deletion mutant strains ($^{**}P < 0.01$, bars indicate 95% CI, P-values were calculated using independent two-sample t-test). (K) The prediction of an alternate model-independent effect of IscU and TusA on viral biosynthesis—for an infection time course in the $\Delta tusA\Delta iscU$ background. Comparison of this plot with (H) and (I) indicates that the competitive binding model is better able to account for the experimental observations. Source data is available for this figure in Appendix B.

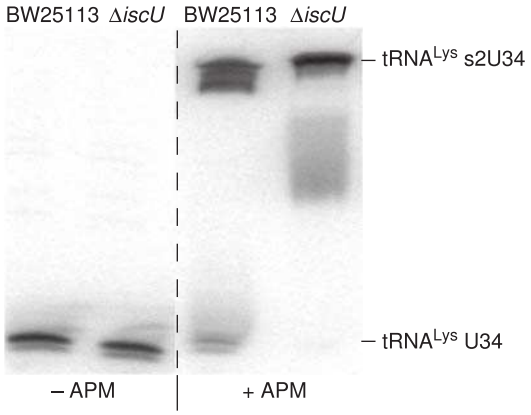


Figure A.7: APM northern blot of $tRNA^{Lys}$ (UUU) in BW25113 and $\Delta iscU$. tRNA northern blots for $tRNA^{Lys}$ (UUU) in urea denaturing gels with and without APM. tRNA extracts were probed to determine the thiolated fraction of U34 in BW25113 and $\Delta iscU$. A relative shift in the APM+ gel reflects the fraction of 2-thiolated U34 $tRNA^{Lys}$ ($tRNA^{Lys}$ s2U34). A small but detectable level of hypomodified $tRNA^{Lys}$ ($tRNA^{Lys}$ U34) was seen in BW25113 but was essentially undetectable in $\Delta iscU$.

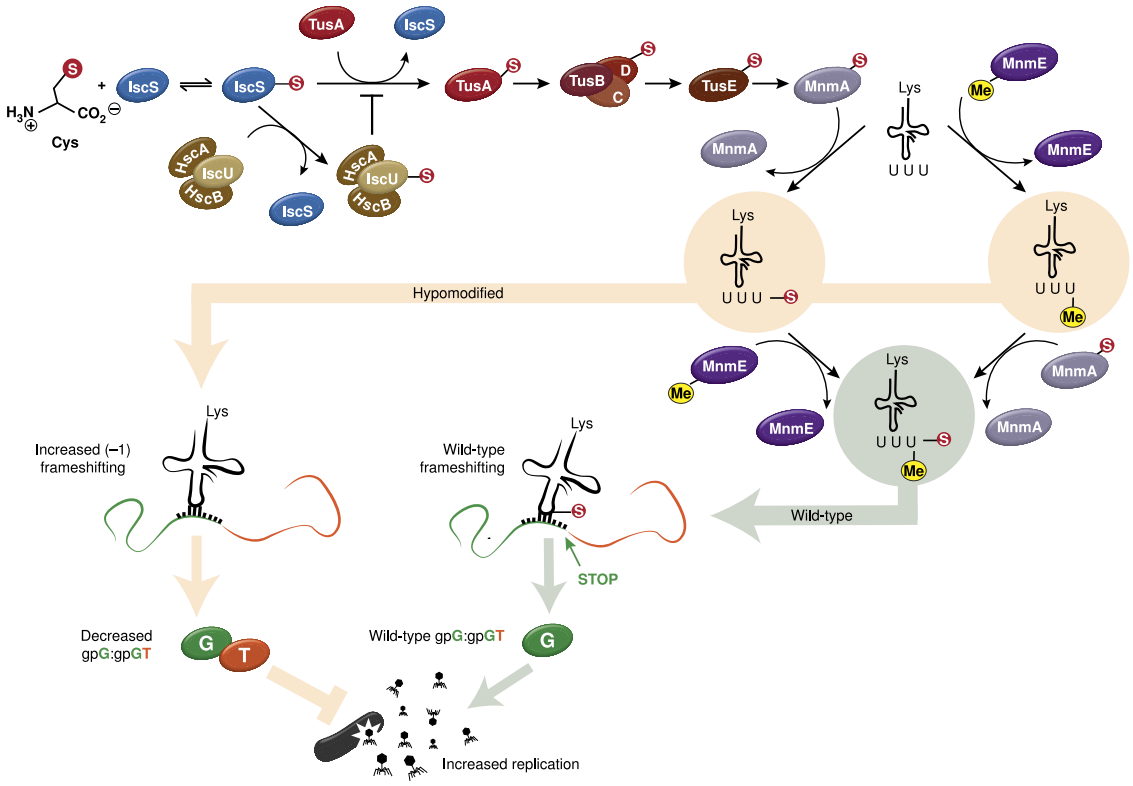


Figure A.8: A network linking host resistance to viral infection to sulfur metabolism, tRNA modification, PRF, and competitive protein binding. IscS obtains sulfur from L-cysteine and passes it to either TusA or IscU. From TusA, the TUS pathway leads to thiolation by MnmA of $tRNA^{Lys}$ (UUU) U34, which is also methylated by MnmE. During viral infection, the WT amount of modified tRNA leads to normal translation (and frameshifting) for the viral *G* and *T* genes, and consequently a favorable ratio of gpG:gpGT and virion production. Hypomodification of $tRNA^{Lys}$ (UUU), whether by deletion of *mnmE* or the TUS pathway genes, or by overexpression of IscU, leads to increased frameshifting during translation of *G* and *T*. This increase in frameshifting leads to more gpGT expression and subsequently, a lower gpG:gpGT ratio and decreased virion production.

Appendix B

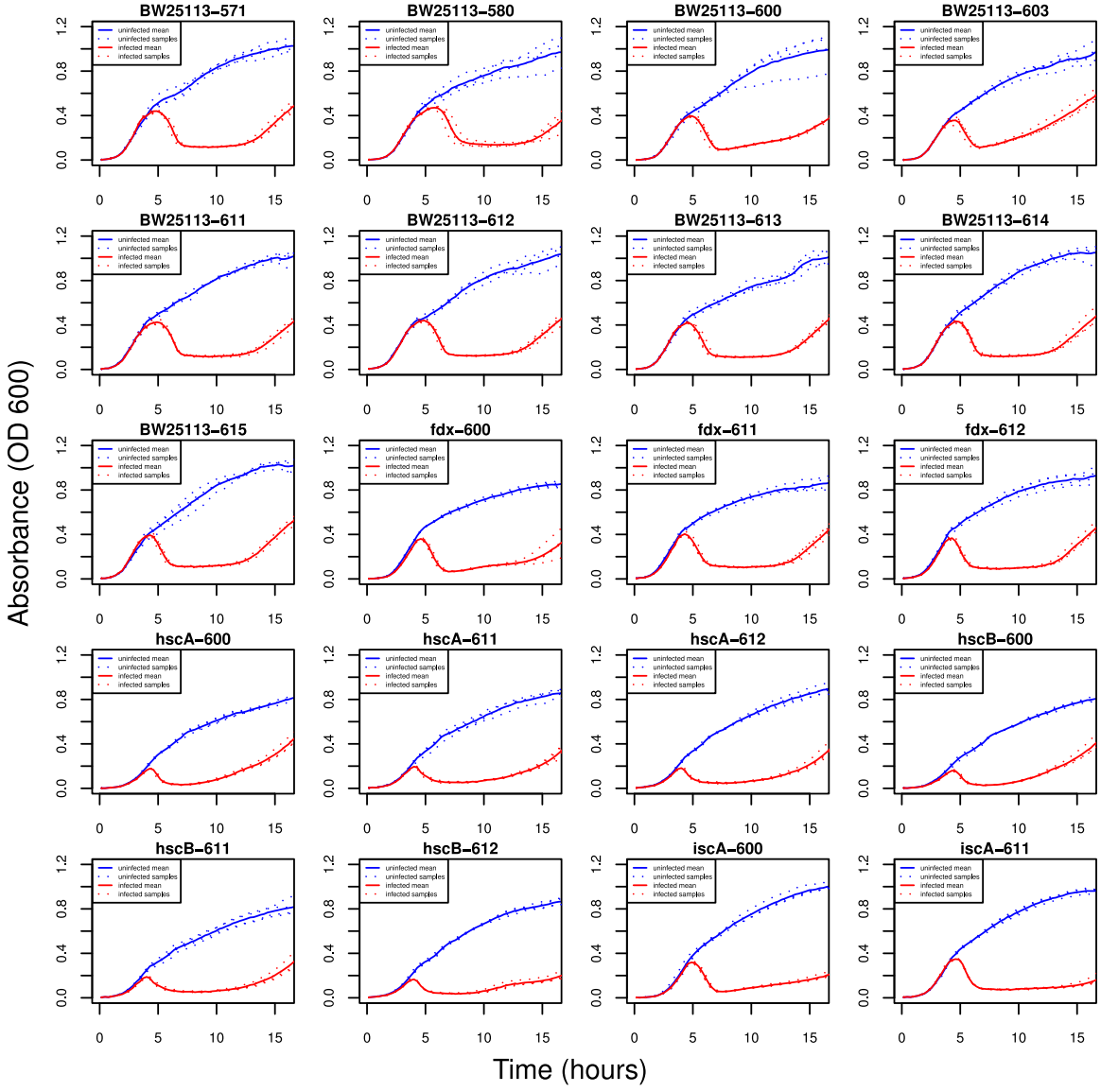
Supplement to "Competing pathways control host resistance to virus via tRNA modification and programmed ribosomal frameshifting"

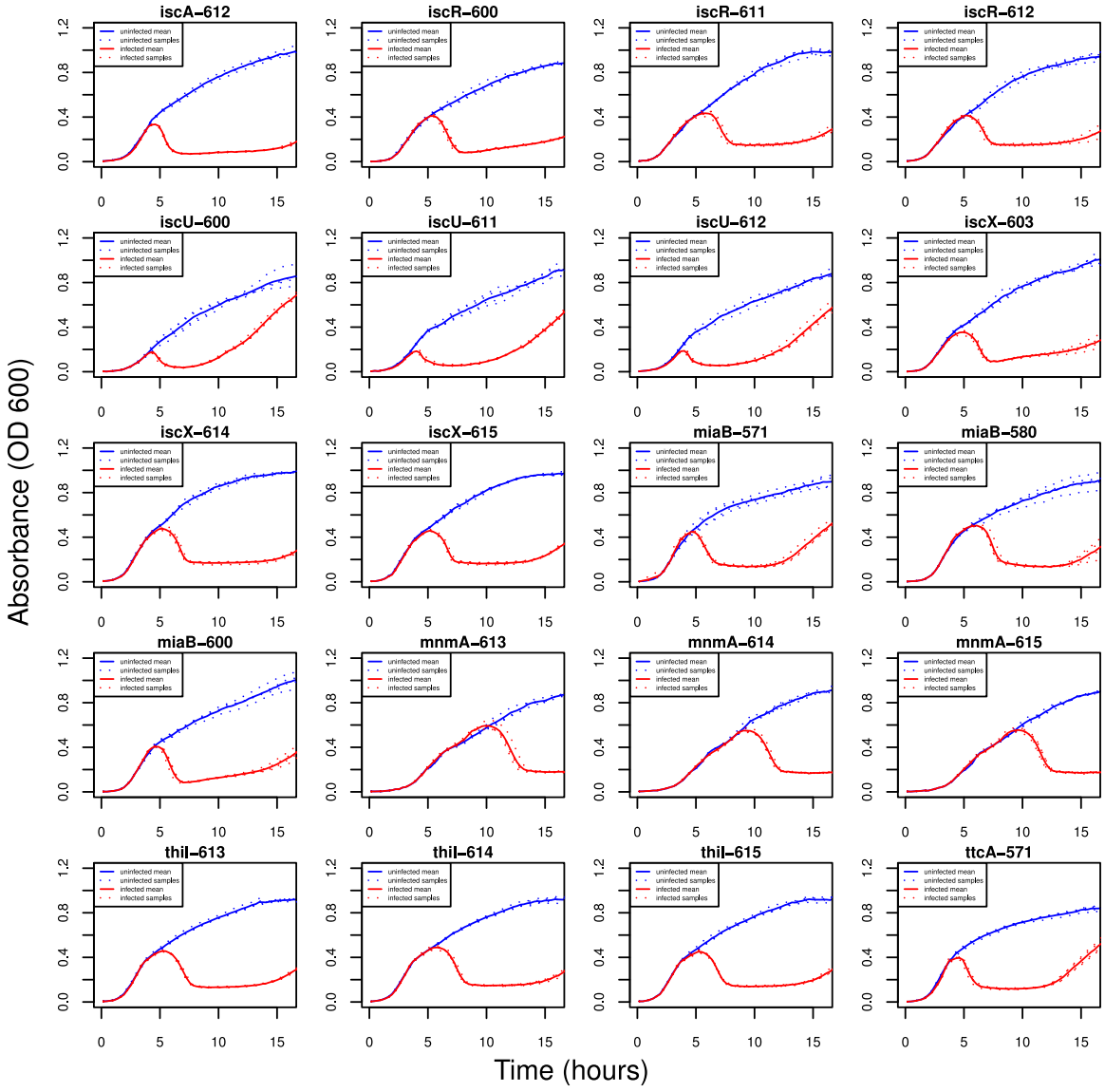
	Competitive Inhibition Model	Independent Effect Model	Value
Dissociation constant for IscS and cysteine	$k_{\mathrm{d,IscS-Cys}}$	N/A	$2.7 \cdot 10^{-6}$
Dissociation constant for IscS and IscU	$k_{\mathrm{d,IscS-IscU}}$	$k_{\mathrm{d,IscS-IscU}}$	$2 \cdot 10^{-6}$
Dissociation constant for IscS and TusA	$k_{\mathrm{d,IscS-TusA}}$	$k_{\mathrm{d,IscS-TusA}}$	$1.2 \cdot k_{\mathrm{d,IscS-IscU}}$
Forward rate of IscS and cysteine interaction	k_1	N/A	10^5
Reverse rate of IscS and cysteine interaction	k_{-1}	N/A	$k_1 \cdot k_{\mathrm{d,IscS-Cys}}$
Rate of formation of thiolated IscS	k_2	N/A	0.1417
Forward rate of thiolated IscS and IscU interaction	k_3	k_{1f}	10^5
Reverse rate of thiolated IscS and IscU interaction	k_{-3}	k_{1r}	$k_3 \cdot k_{\mathrm{d,IscS-IscU}}$
Irreversible rate of formation of thiolated IscU	k_4	N/A	k_{-3}
Forward rate of thiolated IscS and TusA interaction	k_5	k_{2f}	10^5
Reverse rate of thiolated IscS and TusA interaction	k_{-5}	k_{2r}	$k_5 \cdot k_{\mathrm{d,IscS-TusA}}$
Irreversible rate of formation of thiolated TusA	k_6	N/A	k_4
Irreversible rate of sISC modification	k_7	k_{3f}	10^5
Irreversible rate of sTUS modification	k_8	k_{4f}	k_7

Rate of lambda infection (normalized)	k_i	k_i	0.25
Frequency of lambda lytic decision	f	f	0.93
Lysogen growth rate (normalized)	μ^*	μ^*	0.3
Lysogen carrying capacity (normalized)	K^*	K^*	0.4
Rate of lysogen induction (normalized	k_s	k_s	0
Burst rate for competitive inhibition model	b	N/A	$10 \cdot \frac{\mathrm{sTUS}+\mathrm{sTUS}_{\mathrm{WT}}}{\mathrm{sTUS}_{\mathrm{WT}}}$
Burst rate for independent effect model	N/A	b	$10 \cdot \frac{\mathrm{sTUS}+\mathrm{sTUS}_{\mathrm{WT}}}{\mathrm{sTUS}_{\mathrm{WT}}} - 50 \cdot \frac{\mathrm{sISC}+\mathrm{sISC}_{\mathrm{WT}}}{\mathrm{sISC}_{\mathrm{WT}}}$

Table B.1: Description of parameters in Competitive Inhibition Model vs. Independent Effect Model.

Supplementary Information




iscA–612
iscR–600
iscR–611
iscR–612
iscU–600
iscU–611
iscU–612
iscX–603
iscX–614
iscX–615
miaB–571
miaB–580
miaB–600
mnmA–613
mnmA–614
mnmA–615
thiI–613
thiI–614
thiI–615
ttcA–571
uninfected mean
uninfected samples
infected mean
infected samples
Absorbance (OD 600)
Time (hours)

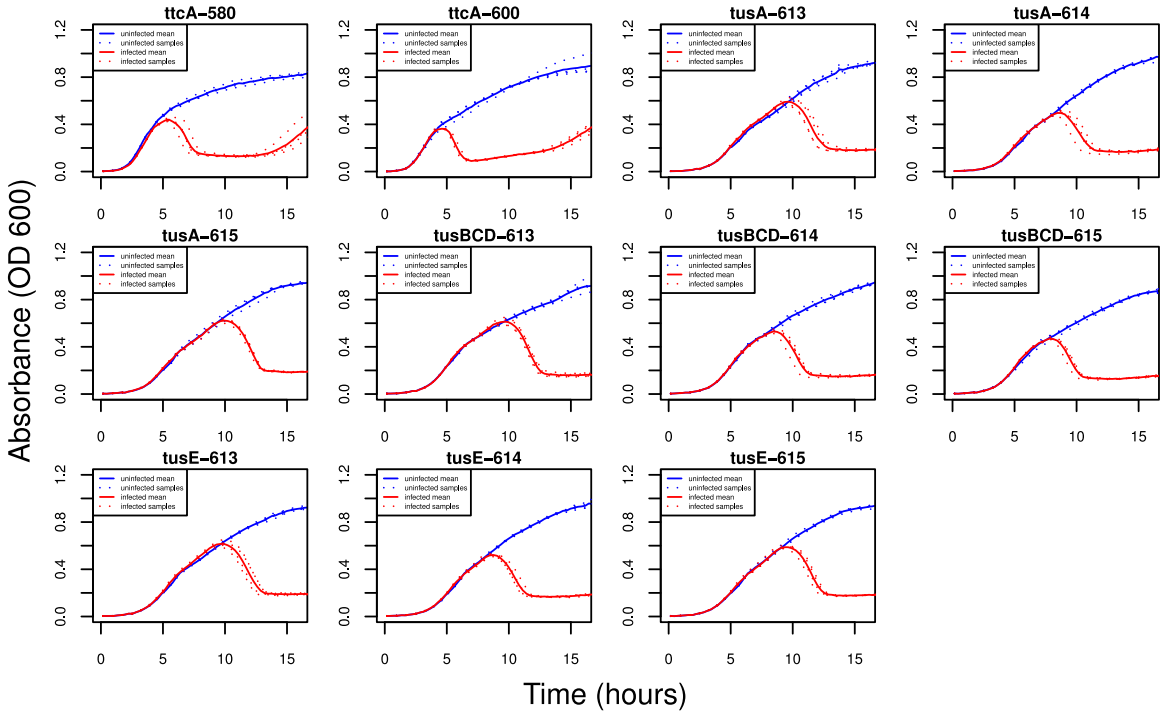
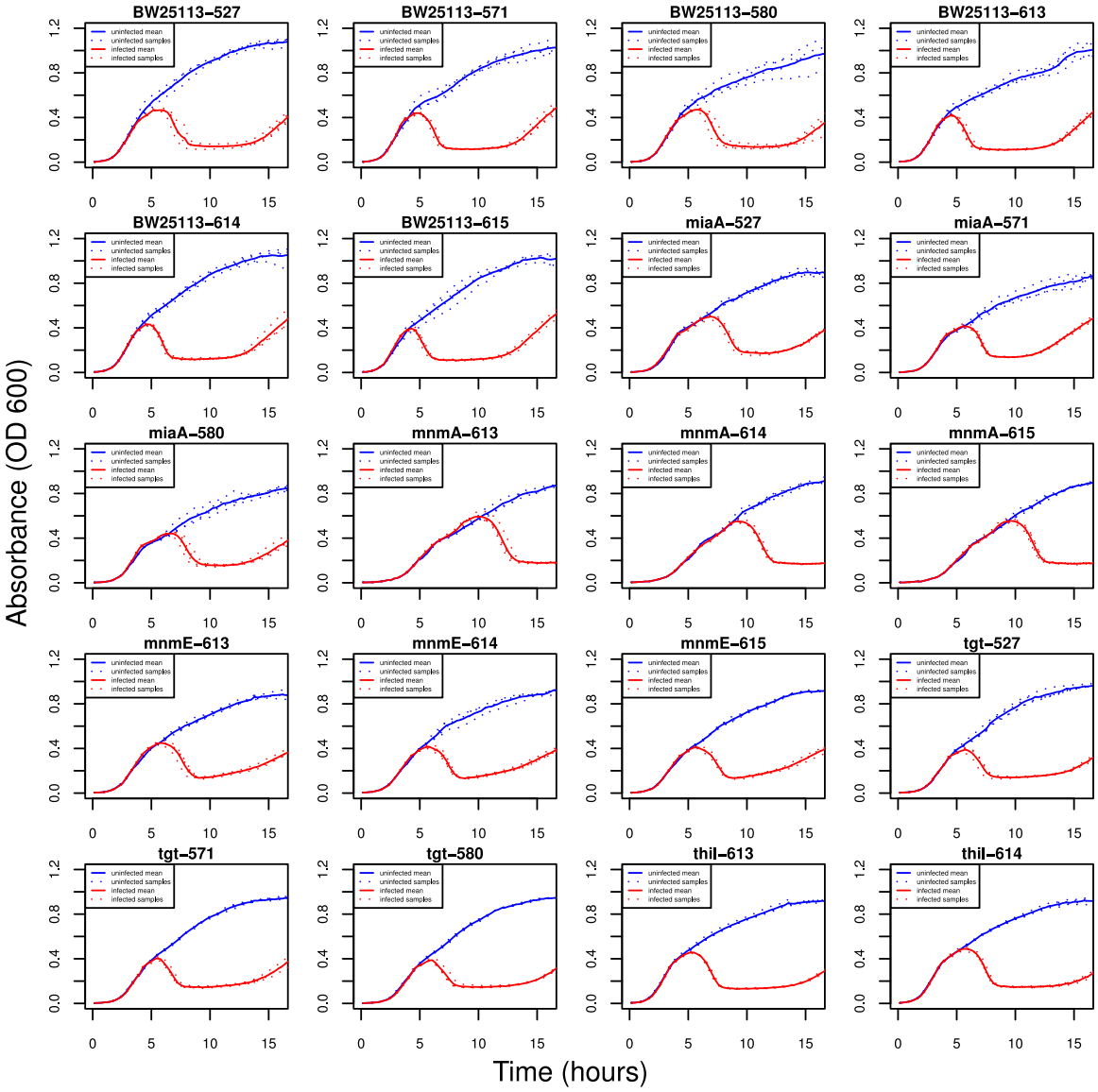


Figure B.1: (Multiple pages) Cell-culture infection dynamics for TUS and ISC pathway knockout strains infected with lambda phage.


BW25113−527
BW25113−571
BW25113−580
BW25113−613
BW25113−614
BW25113−615
miaA−527
miaA−571
miaA−580
mnmA−613
mnmA−614
mnmA−615
mnmE−613
mnmE−614
mnmE−615
tgt−527
tgt−571
tgt−580
thil−613
thil−614
uninfected mean
uninfected samples
infected mean
infected samples
Absorbance (OD 600)
Time (hours)

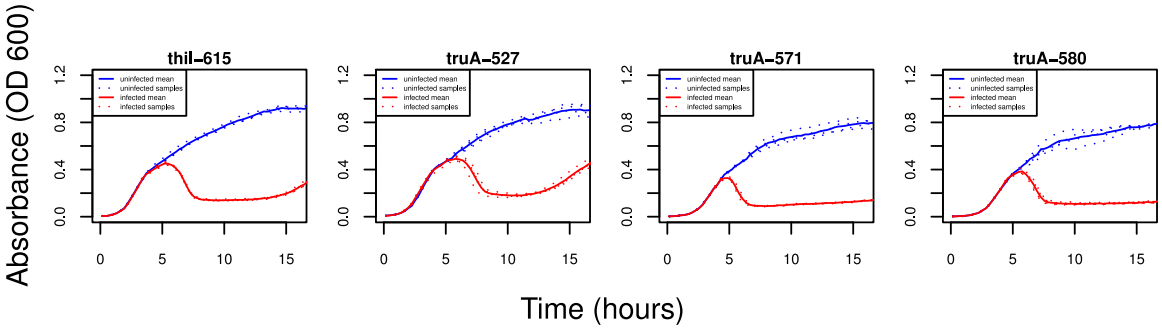


Figure B.2: (Multiple pages) Cell-culture infection dynamics for *E. coli* knockouts of genes known to affect frameshifting.

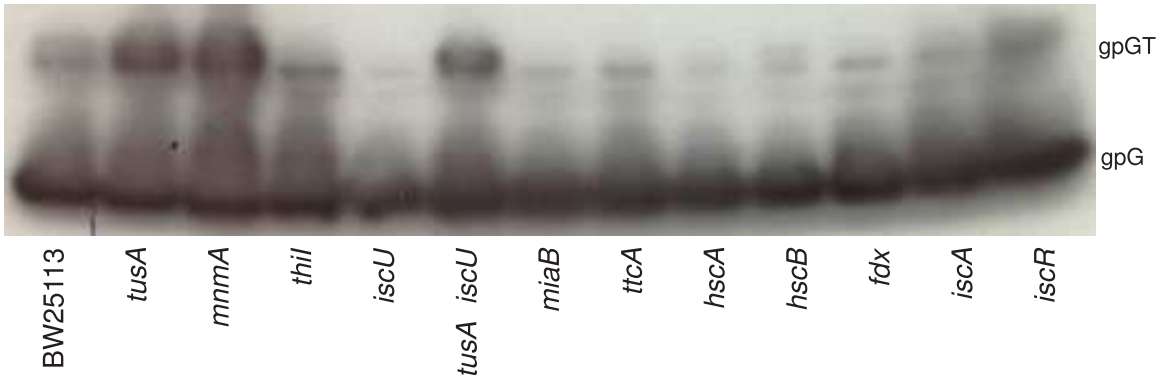


Figure B.3: Immunoblotting of pBAD-λGT in BW25113 and several strains from both the TUS and ISC pathways. We induced expression of the pBAD-λGT transcript with 0.02% L-arabinose for 2 hours and assessed the gpG and gpGT protein levels by immunoblotting against the Xpress Epitope tag.

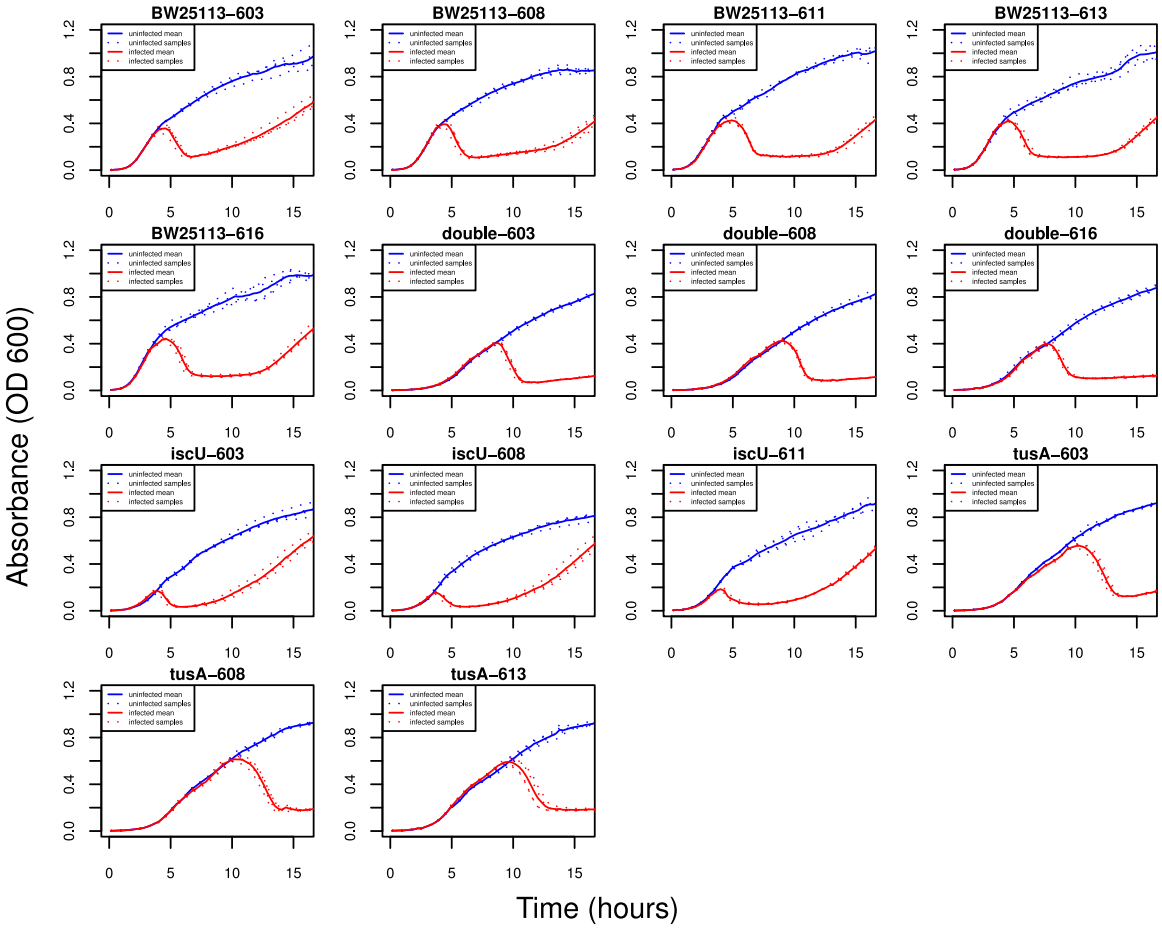


Figure B.4: Cell-culture infection dynamics for *E. coli* knockouts for *tusA*, *iscU*, and *tusAiscU* double knockout.

Appendix C

Supplement to Crick's "complete solution of *E. coli*," 40 years later

C.1 Introduction

This document serves as a companion to our source code and simulation archive[1] in support of the main text. Similar to the structure of the supplement to the *M. genitalium* model [7], we first present an overview of our computational methods and then delve into the implementation details. We then provide a description of our experimental methods used to collect (1) expression data to initially parameterize the model and (2) protein decay rates used to refine the model.

Constructing a gene-complete whole-cell model of *E. coli* is a major undertaking. While the overarching engineering goal for the *M. genitalium* model was to include the function of every annotated gene, *E. coli* contains roughly ten times as many genes as *M. genitalium* and 50-100 times as many molecules that can interact, presenting us significant challenges in both modeling and computation. Although a gene-complete model of *E. coli* is our long-term goal, before focusing on increasing the number of genes, we decided to focus on improving and expanding our utilization of data. We chose to work towards a whole-cell model that integrated as much organism-specific data as possible, preferably across multiple environments and growth conditions, in which results and predictions could be experimentally verified. Building larger, gene-complete models would be impossible without the innovation in parameter estimation, data integration, modeling framework extensibility, and feedback regulation this goal required.

Perhaps the largest difference between *M. genitalium* and *E. coli* physiology is the extensive amount

[1] `https://simtk.org/projects/ecoli`. Updates and errata will be available at this location as well.

of regulation and control present in the latter. Whereas *M. genitalium* can only be cultured—and thus simulated—in a rich medium, *E. coli* can grow in a number of different environments, at a number of different growth rates (see Section C.2.6). This behavior is mediated by extensive regulatory mechanisms at the transcriptional and post-transcriptional levels that we can now simulate and that we describe in the following sections.

As discussed in the main text, while the model presented here is not gene-complete, it incorporates the biological processes for which the majority of high-throughput data is available. We spent considerable effort evaluating data sets and merging them into our framework. Ultimately, this enabled us to make the quantitative comparisons presented in the main text—comparisons that could not be made when modeling *M. genitalium*. However, this version of the *E. coli* model lacks several of the sub-models implemented in the *M. genitalium* model. Going forward, we plan to continue working toward a gene-complete model of *E. coli*.

For readers familiar with our *M. genitalium* work, we summarize our improvements over that model in terms of *Modeling* and *Computation*:

Modeling:

- We have a quantitative model of transcriptional regulation that incorporates the function of 22 transcription factors regulating 355 genes. This includes one- and two-component signaling processes to modulate transcription factor activity, as well as the modulation of RNA polymerase recruitment via TF-DNA binding interactions.
- The metabolic model is much more robust and includes detailed quantitative (Michaelis-Menten) parameters for 340 reactions. The metabolic model now maintains concentrations of metabolite pools subject to resource availability rather than producing metabolites in a fixed ratio at every time step. This enables the metabolic model to adjust to time-dependent/cell cycle-dependent behavior from other simulated processes while maintaining homeostasis.
- We now have an implementation of growth-rate control that enables our simulated *E. coli* cells to grow at different doubling times as a function of the environment. This improvement, supported by our model of DNA replication which can track multiple rounds of replication, is showcased in the main text and was not possible in *M. genitalium* simulations.
- Our model of translation uses translational efficiency data to inform ribosome binding to mRNA transcripts.
- We have a more detailed model of RNA decay that incorporates both the rates of degradation due to endonuclease-mediated cleavage and the rates of transcript digestion by exoRNases.

Computation:

- We have decreased simulation run-time by nearly two orders of magnitude. Whereas *M. genitalium* simulations took roughly 10 hours to run, *E. coli* simulations—which account for 50 times more molecules—take approximately 15 minutes to simulate the life cycle of an *E. coli* cell. We achieved this by (1) improving file I/O, (2) writing inner loops in Cython or C, and (3) warm-starting the linear solver in our metabolic model.
- This improvement in run-time enables us to reliably simulate multiple generations of cells (e.g., as shown in the main text), which was not possible with the *M. genitalium* simulations.
- Additionally, we have improved the whole-cell application programming interface (API). Code is much more readable, and on-boarding new researchers takes roughly 2 weeks rather than 6 months.

With this overview in mind, we now present our computational methods.

C.2 Computational methods

Figure C.1 summarizes the overall workflow of running and analyzing simulations. We begin with data sets from the primary literature, our own experiments, and databases (e.g., EcoCyc) which we unify into a `KnowledgeBase` (KB). We then reconcile parameters using a heuristic fitting procedure. Using these reconciled parameters, we run simulations and save their output for downstream analysis. In principle, as highlighted in Figure C.1, the first three procedures can be performed just once and the remaining procedures can be performed multiple times to explore the effects of different perturbations.

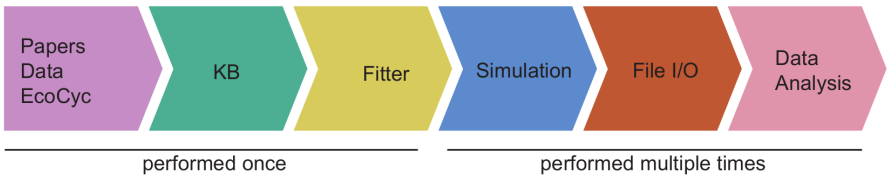


Figure C.1: **Overall workflow**
Starting with curated datasets, the `KnowledgeBase` is created. Using the `KnowledgeBase`, parameters are reconciled in the Fitter and used as initial conditions for the Simulation. For each simulation run, these preparatory steps are performed once. After all simulations are performed, visualizations are produced with analysis scripts.

C.2.1 Reconstruction and fitting

The *E. coli* model is based on a reconstruction of *E. coli* physiology that considers a multitude of data including both single-cell and population measurements. Some examples include average cell masses, macromolecular composition (e.g., RNA, protein, and DNA mass fractions), gene expression, protein and RNA half-lives, enzymatic rates, cell cycle parameters, and gene annotation. The bulk of the experimental data we used comes from three lab strains of *E. coli*: K-12 MG1655, B/r, and BW25113. The model can therefore be thought of as a composite strain which uses all of these data. In most cases, this assumption likely holds. However, in some cases these strains have different physiology with the most notable example being a difference in growth rate, under similar environmental conditions, between MG1655 (where most high-throughput data comes from) and B/r (where detailed composition data have been obtained). Our model was optimized using the B/r growth rates.

Due to the diverse nature of the data sources considered for the model, it was necessary to perform data reconciliation to create a single consistent parameter set. For example, cell mass must, on average, grow exponentially and double within the expected doubling time given all of the parameters. More generally, the reconciled parameter set, or `KnowledgeBase`, should produce balanced exponential growth while satisfying constraints set by cell theory and physiology.

Testing whether a set of parameters produced balanced exponential growth would ideally take place in the full large-scale cell simulation framework. However, even with >1 order of magnitude improvements in runtime over the *M. genitalium* model, this proves too computationally intensive. Therefore, to enable computationally tractable parameter fitting we created surrogate models that approximated the average behavior of the full-scale simulation using heuristic routines. For example, we approximate the output of a dynamic, stochastic simulation that produces a spectrum of proteins with a statistical distribution of protein counts. These surrogate models were executed in order to numerically fit parameters to our constraints (e.g., of cell theory).

In cases where parameters were inconsistent with the constraints imposed, or no value was known for a parameter, one of two operations was performed: (1) iteratively changing the parameter value (such as the expression of RNA polymerases, ribosomes, and metabolic enzymes) until constraints are satisfied or (2) calculating the parameter value from other known and reconciled data (such as using the dissociation constants of ligands binding their binding partners to calculate the anticipated intracellular concentration of ligands).

C.2.2 Estimating the number of parameters

We are often asked how many parameters are contained our models. Estimating this as a simple number belies the complex and heterogeneous nature of the model, where each sub-model is specified using a different mathematical formalism. At one extreme, every piece of data could be considered a parameter (e.g., we could count as a parameter every single nucleotide in the chromosome sequence), but we feel this isn't a helpful estimate. Likewise, the stoichiometric coefficients in the metabolic network, molecular masses of each mRNA and protein species, as well as all of the data in our RNA expression databases, could all be treated as parameters—but we don't take them into account when answering this question. While one could therefore count parameters in many different ways, Table C.1 provides the breakdown for the over nineteen thousand parameters that we state in the main text.

Estimating the number of fit parameters can also be ambiguous, as everything is linked, and in some cases it can be impossible to change a parameter in isolation. For example, given the fact that we account for (1) all molecules in a cell and (2) the total mass of the cell, changing counts of one molecule (by changing one parameter) means that we either have to update counts of other molecules (to maintain the total mass of the cell), or change the total mass of the cell (to account for the change in molecule counts). Not withstanding this difficulty, Table C.2 provides our breakdown of the major sets of parameters that were modified from their initial values.

Description	Parameter count
RNA Polymerase recruitment strengths (basal and TF-modulated)	4996
EndoRNase-RNA affinities (govern decay rate of each RNA)	4558
Translation efficiencies	4353
Protein half-lives	4353
Metabolic reaction constraints	616
Metabolite pools (basal condition)	140
External exchange flux bounds (basal condition)	54
Dissociation constants (e.g., for ligand-TF binding)	28
Reaction rates for two-component systems	21
Total	19119

Table C.1: Estimate of number of parameters in the model.

Description	Parameter count
RNA expression for the ribosome (58 genes)	58
TF-ligand affinities for TrpR, ArgP, Lrp, PutA, MetJ, CytR	6
RNA expression for *pabC*, *menH*, *cdsA*, *yibQ**, *atoB**	5
Translational efficiencies for *pabC*, *menH*, *yibQ**, *atoB**	4
RNA expression for RNA Polymerase (*rpoA*, *rpoB*, *rpoC*)	3
RNA degradation rate for RNA Polymerase (*rpoA*, *rpoB*, *rpoC*)	3
RNA degradation rate for *pabC*, *cdsA*	2
Protein decay rate of YibQ*	1
Total	82

Table C.2: Estimate of number of fit parameters in the model. *Denotes that it was necessary to modify this parameter for the anaerobic simulations to be viable.

C.2.3 Initial conditions

The state of the *E. coli* simulation is initialized immediately after cell division. Using the unified parameter set created during `Reconstruction` and stored in the `KnowledgeBase` the counts and properties of every species are set, using a statistical model to give each simulation a uniquely determined random initial state that, on average, fits experimental data.

Initializing RNA and protein counts The counts of RNA and protein molecules are initialized as follows. First, the total counts of RNA and protein molecules of each species are computed using Equation C.1.

$$M_{total} = \sum_i c_i \cdot MW_i / N_a \tag{C.1}$$

Where M_{total} is the total mass of RNA or protein, c_i, f_i, and MW_i are the counts, mass fraction, and molecular weight of RNA or protein i, and N_a is Avogadro's number. Substituting $c_i = c_{total} \cdot f_i$, where c_{total} is the total counts of RNA or protein species, and rearranging gives Equation C.2.

$$c_{total} = \frac{M_{total}}{\vec{f} \cdot \vec{MW} / N_a} \tag{C.2}$$

The total masses of RNA and protein per cell (M_{total}) as well as their expected distributions ($\vec{f}$) are all known from reconciled datasets in `KnowledgeBase`.

Total counts and the expected distribution are then used to sample a `multinomial` distribution to statistically compute the counts of each individual RNA or protein species using Equation C.3.

$$\vec{c} = \texttt{multinomial}(\vec{f}, c_{total}) \tag{C.3}$$

Details of this algorithm can be found in Algorithm 1.

Algorithm 1: Algorithm for initializing counts of RNA and protein in *E. coli* model

Input : $M_{total}^{RNA}, M_{total}^{Protein}$ Total mass per cell of RNA and protein

Input : MW_i, MW_j Molar molecular weights of RNA $i = 1$ **to** n_{RNA} and protein $j = 1$ **to** $n_{protein}$

Input : f_i^{RNA} mass fraction based on RNA expression of RNA $i = 1$ **to** n_{RNA}

Input : $k_{d,j}$ degradation rate of protein $j = 1$ **to** $n_{protein}$

Input : N_a Avogadro's number

Input : ψ_j translational efficiencies of each mRNA $j = 1$ **to** $n_{protein}$

1. Calculate total counts of RNAs (C_{total}^{RNA}) based on total RNA mass (M_{total}^{RNA}) and distribution of expression ($\vec{f}^{RNA}$) from `KnowledgeBase`

$$c_{total}^{RNA} = \frac{M_{total}^{RNA}}{\vec{f}^{RNA} \cdot \vec{MW}/N_a}$$

2. Calculate counts of each RNA ($\vec{c}_{RNA}$) by sampling a `multinomial` distribution c_{total}^{RNA} times weighted by the expected distribution of expression ($\vec{f}^{RNA}$).

$$\vec{c}_{RNA} = \texttt{multinomial}\ (\vec{f}^{RNA}, c_{total}^{RNA})$$

3. Calculate expected distribution of protein counts ($\vec{f}^{protein}$) based on expected distribution of RNA counts ($\vec{f}^{RNA}$), translational efficiencies ($\vec{\psi}$), protein degradation rates ($\vec{k_d}$), and dilution using a steady state assumption.

$$\vec{f}^{protein} = \frac{\vec{f}^{RNA} \cdot \vec{\psi}}{\frac{ln(2)}{\tau} + \vec{k_d}}$$

4. Calculate total counts of proteins ($c_{total}^{protein}$) based on total protein mass ($M_{total}^{protein}$) and distribution of counts ($\vec{f}^{protein}$).

$$c_{total}^{protein} = \frac{M_{total}^{protein}}{\vec{f}^{protein} \cdot \vec{MW}/N_a}$$

5. Calculate counts of each protein ($\vec{c}_{protein}$) by sampling a `multinomial` distribution $c_{total}^{protein}$ times weighted by the expected distribution of expression ($\vec{f}^{protein}$).

$$\vec{c}_{protein} = \texttt{multinomial}\ (\vec{f}^{protein}, c_{total}^{protein})$$

Result: Counts of RNA are set at the beginning of the first generation of simulated cells

Initializing small molecule counts The counts of small molecules such as cytoplasmic and membrane constituents are initialized as follows. Expected concentrations of small molecules are either known experimentally, or computed from an FBA biomass reaction and stored as a reconciled

dataset in the `KnowledgeBase`. The volume of the cell is computed using its mass divided by its density. Therefore adding counts of small molecules to the cell in order to match a concentration will necessarily change the volume of the cell. Using a system of linear equations, the counts of each small molecule and the new adjusted mass of the cell (adding the mass of the new small molecule counts) is calculated. The details of this calculation can be found in Algorithm 2.

Algorithm 2: Algorithm for initializing counts of small molecules in *E. coli* model

Input : C_k^{SM} concentration of each small molecule $k = 1$ **to** n_{SM}

Input : ρ density of cell

Input : MW_k molecular weight of small molecule $k = 1$ **to** n_{SM}

Input : m_{init} Initial mass of the cell only considering RNA, protein, and DNA

1. Calculate masses of each metabolite to add (m_k) in order to achieve known metabolite concentration (C_k^{SM}) from `KnowledgeBase` assuming cell volume is calculated by dividing the cell mass by its density.

$$\begin{bmatrix} \frac{\rho}{C_1^{SM} \cdot MW_1} - 1 & -1 & \dots & -1 \\ -1 & \frac{\rho}{C_2^{SM} \cdot MW_2} - 1 & & \vdots \\ \vdots & & \ddots & -1 \\ -1 & \dots & -1 & \frac{\rho}{C_k^{SM} \cdot MW_k} - 1 \end{bmatrix} \cdot \begin{bmatrix} m_1 \\ m_2 \\ \vdots \\ m_k \end{bmatrix} = \begin{bmatrix} m_{init} \\ m_{init} \\ \vdots \\ m_{init} \end{bmatrix}$$

2. Calculate expected counts of each small molecule (c_k^{SM}).

$c_k^{SM} = \frac{m_k}{MW_k \cdot N_a}$

Result: Counts of each small molecule are calculated and set in state

Initializing chromosome state In *E. coli* there are potentially multiple rounds of replication proceeding simultaneously at any point in the cell cycle. The simulation begins immediately after cell division and the number and position of any replication forks that are inherited from previous generations must be determined to correctly initialize the simulated cell. The number of origins of replication, replication forks, and their positions are initialized as follows.

First, the number of rounds of replication that on average need to proceed simultaneously can be estimated in an average cell in a population using the length of time required to replicate the chromosome (C period) and the length of time for cytokinesis (D period) as well as the expected doubling time given the environment (τ). The number of simultaneous rounds (n_{limit}) can be calculated with Equation C.4 as the ratio of C+D period over the doubling time [214]. Because we are considering a specific cell and not an average of a population of cells, the number of rounds of replication needs to be an integer, and we take the floor because a fractional round of chromosome

initiation has not yet occurred.

$$n_{limit} = floor(\frac{C+D}{\tau}) \tag{C.4}$$

For every round of replication proceeding there are a pair of replication forks and a pair of origins of replication. We are assuming that on average a cell after division has inherited one chromosome molecule (i.e. no more than one terC), and that it may have more than one round of replication proceeding on it (i.e. number of oriC ≥ 1). Therefore the number of origins of replication (n_{origin}) is defined by Equation C.5.

$$n_{origin} = 2^{n_{limit}} \tag{C.5}$$

Finally, the position between the oriC and the terC of each replication fork needs to be determined on average. This can be calculated with Equation C.6 where f is the fraction of length between the origin and terminus of replication that the replication fork has proceeded for the nth round of replication (where n can be any integer value between 1 and n_{limit}).

$$f = 1 - \frac{n \cdot \tau - D}{C} \tag{C.6}$$

Where n is every integer value 1,2,...n_{limit}. The position in nucleotides (l) can then be calculated from Equation C.7 where L is the total length of the chromosome in *E. coli*.

$$l = f \cdot \frac{L}{2} \tag{C.7}$$

Proper initialization of the cell ensures the simulation begins close to the steady state of the system, and in practice the simulation is relatively stable. Perturbations in the ratio of cell mass to number of origins of replication quickly re-converge to steady state for a given environment. A detailed algorithm for chromosome initialization can be found in Algorithm 3.

Algorithm 3: Algorithm for initializing chromosome state in *E. coli* model

Input : C length of C period
Input : D length of D period
Input : τ expected doubling time
Input : L length of chromosome in nucleotides
$n_{limit} = \texttt{floor}(\frac{C+D}{\tau})$
$n = 1$ **while** $n \leq n_{limit}$ **do**
  1 Determine initial number of forward and reverse replication forks ($n_{fork,f,init}$ and $n_{fork,r,init}$) for the given round of replication
  $n_{fork,f,init} = 2^{n-1}$
  $n_{fork,r,init} = 2^{n-1}$
  2 Determine position of each fork on forward and reverse strand as a fraction of total chromosome length (f)
  $f = 1 - \frac{n \cdot \tau - D}{C}$
  3 Calculate position of each fork on forward and reverse strand (l) in nucleotides and initialize $n_{fork,f,init}$ and $n_{fork,r,init}$ DNA polymerases at the calculated positions
  $l_{fork,f,init} = f \cdot \frac{L}{2}$
  $l_{fork,r,init} = f \cdot \frac{L}{2}$
  4 Increment round of replication that is being initialized
  $n = n + 1$
end
$n_{origin,init} = 2^{n_{limit}}$
Result: State of chromosome in cell is correctly initialized around the average of a population

C.2.4 Simulation algorithm

A whole-cell model may be thought of as a system of ordinary differential equations (ODEs) where the cellular states are analogous to the ODEs' state variables and the cellular processes are analogous to the differential equations. Extending this analogy, the *E. coli* model is simulated using an algorithm that is comparable to those used to numerically integrate ODEs. The only significant difference from ODE numerical integration is that shared resources stored in cellular states must be partitioned to each cellular process in order to ensure mass conservation. Algorithm 4 and Figure C.2 summarize the simulation algorithm to execute a time step. The temporal evolution of the cell state is calculated on a short time scale (typically <1 second) by allocating cell state variables among processes (described in Algorithm 4 under "Allocate shared resources"), and executing the process code that updates counts in the state variables until the cell divides. Critically, we make the assumption that over a short time scale, each process acts independently.

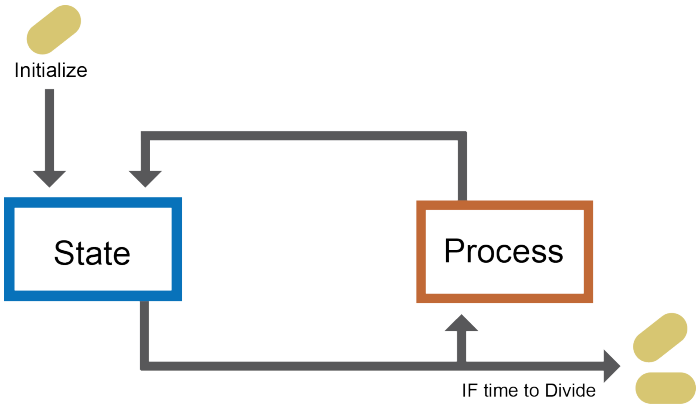


Figure C.2: **Schematic of whole-cell simulation algorithm**
The model takes in a set of initial conditions about a single cell and encodes this information as States, which contains information about each molecule. At the start of a time step these molecules are fed into Processes, while at the end of a time step the molecule information within States is updated. This sequence is iterated over the entire life cycle of the cell until it divides, which constitutes a single generation. Each of the daughter cells could then serve as the initial conditions for a new generation.

Algorithm 4: Algorithm for whole-cell dynamic simulation

Initialize simulation states (described in Algorithm 1, 2, 3)
repeat
  `/* Allocate shared resources */`
  for *each molecule* i **do**
    for *each process* j **do**
      1. Calculate demand $d_{i,j}$ of process j for molecule i.
      2. Divide total count c_i of molecule i into partition $p_{i,j}$, for each process proportional to the demand such that $p_{i,j} = c_i \frac{d_{i,j}}{\sum_j d_{i,j}}$.
    end
  end
  `/* Calculate temporal evolution */`
  for *each process* j **do**
    1. Retrieve partitioned molecules $p_{i,j}$.
    2. Compute the contribution of process j to the temporal evolution of the partitioned molecules $\Delta p_{i,j}$.
    3. Update partitioned molecule counts $p_{i,j} = p_{i,j} + \Delta p_{i,j}$.
  end
  `/* Merge partitioned molecules */`
  for *each molecule* i **do**
    Update counts c_i based on updated partitions computed in each process, $c_i = \sum_j p_{i,j}$
  end
  Increment simulation step by 1
until *cell division*;
Result: Whole-cell model is executed for one cell cycle

C.2.5 States and Processes

States

The simulation States are defined as the counts, locations, and attributes of every species in the model at a given time step, which are then operated on by Processes. There are two classes of States within the model - `BulkMolecules` and `UniqueMolecules`.

The `BulkMolecules` state tracks species in the simulation where individuals are not further distinguished from each other. For example, two ATP molecules in the cytoplasm are considered identical and tracked in `BulkMolecules`.

The `UniqueMolecules` state tracks species in the simulation where individuals are distinguishable from each other by an attribute and cannot be interchanged without effect. For example, two ribosomes on different mRNA transcripts are uniquely identified by the transcript they are translating and their location on the transcript.

Processes

The simulation Processes update the simulation States from one time step to the next. Each `Process` represents an aspect of physiology of an *E. coli* cell. We discuss the implementation of each Process in detail in Section C.3.

C.2.6 Environments

We simulate different environments by adding or removing exchange flux bounds for the metabolic network (see Section C.3.2). Based on these bounds, the `Metabolism` process updates cellular concentrations of small molecules, and the `Transcription` and `Translation` processes modulate the activities and expression of RNA polymerases and ribosomes to globally shift cell composition. The transcriptional regulatory network (see Section C.3.1) responds to the new small molecule concentrations and adjusts gene expression appropriately.

In the main text we simulate 3 different environments (minimal media with glucose, minimal media with glucose supplemented with amino acids, and anaerobic - minimal media with no oxygen) which map to 3 different growth rates (doubling times of: 44 minutes, 25 minutes, and 100 minutes, respectively) to demonstrate the ability to shift cell composition. In benchmarking our simulations, we also simulated environments that would activate and inactivate each of the transcription factors (not shown).

C.2.7 Computational implementation and workflow

Programming language

The model is primarily implemented in Python, with Cython used for computationally intensive inner loops.

Workflow management

Workflows are defined, managed, and executed using FireWorks - a free open-source code for automating workflow execution which can be defined in Python (https://pythonhosted.org/FireWorks/) [215].

Model API

The *E. coli* model uses an organism independent whole-cell modeling application program interface (API) developed to facilitate model development, human readability, and consistent coding style. The API classes include I/O tools such as `Listeners` for recording data, `Views` for managing interactions between Processes and States, `Containers` for States, and other functions that are useful for writing Processes.

Listeners `Listeners` is a class that facilitates writing data to disk during simulation runtime. They create a human readable interface and reduce both the file size of simulation output and post-hoc computation by saving user-specified quantities computed during a simulation.

Views `Views` is a class that abstracts away potential issues with indexing into large matrices and provides a programmatic interface that allow States and Processes to interact cleanly during simulation.

C.3 Processes

The Processes of the *E. coli* model span several major areas of cellular physiology. We have clustered the Processes into groups that correspond to figures in the main text (Central Dogma, Metabolism, and Balanced Growth) and present these groups in that same order below. We modeled Processes using the most appropriate mathematics for their individual network topology and degree of experimental characterization. Each process is a computational representation of chemical reactions or transformations grouped by a physiological function. The actual division of reactions across processes is a modeling decision made during model construction, and the number of Processes does not reflect their complexity or scope. The inputs and outputs of each `Process` are the counts of metabolites or macromolecules and the catalytic capacity or configuration of the enzymes that catalyze the reactions in each `Process`. This section details the model implementation, computational algorithm, associated data, and relevant code for each `Process`.

C.3.1 Central dogma

Transcription

Model implementation

Transcription occurs through the action of two processes in the model: `TranscriptInitiation` and `TrancriptElongation`. `TranscriptInitiation` models the binding of RNA polymerase to each gene. The number of initiation events per gene is proportional to the number of free RNA polymerases weighted by each gene's synthesis probability. Details are in Algorithm 5.

`TranscriptElongation` models nucleotide polymerization into RNA molecules by RNA polymerases. Polymerization occurs across all polymerases simultaneously and resources are allocated to maximize the progress of all polymerases up to the limit of the expected polymerase elongation rate and available nucleotides. The termination of RNA elongation occurs once a RNA polymerase has reached the end of the annotated gene. Details are in Algorithm 6.

Difference from *M. genitalium* model

The *M. genitalium* model modeled RNA polymerase as existing in 4 states: free, non-specifically bound on a chromosome, bound to a promoter, and actively transcribing a gene. The *E. coli* model simplifies this by assuming RNA polymerase exists in two states: free and actively transcribing. Every time step, free RNA polymerase transitions to the actively transcribing state to maintain an experimentally-observed active fraction of RNA polymerase. The *E. coli* model does not yet include sigma, elongation or termination factors. The *E. coli* model also currently treats each gene as its own transcription unit.

Algorithm 5: Algorithm for RNA polymerase initiation on DNA

Input : f_{act} fraction of RNA polymerases that are active

Input : r expected termination rate for active RNA polymerases

Input : $v_{\text{synth,i}}$ RNA synthesis probability for each gene where $i = 1$ **to** n_{gene}

Input : $c_{RNAP,f}$ count of free RNA polymerase

Input : `multinomial()` function that draws samples from a multinomial distribution

1. Calculate probability (p_{act}) of a free RNA polymerase binding to a gene.

$$p_{act} = \frac{f_{act} \cdot r}{1 - f_{act}}$$

2. Calculate the number of RNA polymerases that will bind and activate ($c_{RNAP,b}$).

$$c_{RNAP,b} = p_{act} \cdot c_{RNAP,f}$$

3 Sample multinomial distribution $c_{RNAP,b}$ times weighted by $v_{synth,i}$ to determine which genes receive a RNA polymerase and initiate ($n_{init,i}$).

$$n_{init,i} = \texttt{multinomial}(c_{RNAP,b}, v_{synth,i})$$

4 Assign $n_{init,i}$ RNA polymerases to gene i. Decrement free RNA polymerase counts.

Result: RNA polymerases bind to genes based on the number of free RNA polymerases and the synthesis probability for each gene.

Algorithm 6: Algorithm for mRNA elongation and termination

Input : e expected RNA polymerase elongation rate in given environment

Input : L_i length of each gene $i = 1$ **to** n_{gene} for each coding gene.

Input : p_j gene position of RNA polymerase $j = 1$ **to** n_{RNAP}

Input : $c_{nuc,k}$ counts of nucleotide $k = 1$ **to** 4

Input : δt length of current time step

`/* Elongate RNA transcripts up to limits of sequence or nucleotides */`

for *each RNA polymerase j on gene i* **do**

1. Based on RNA polymerase position p_j on a gene i and maximal elongation rate e determine "stop condition" (s_j) for RNA polymerase j assuming no nucleotide limitation.

$$s_j = \texttt{min}(p_j + e \cdot \delta t,\ L_i)$$

Stop condition is either maximal elongation rate scaled by the time step or the full length of sequence (i.e. the RNA polymerase will terminate in this time step).

2. Derive sequence between RNA polymerase position (p_j) and stop condition (s_j).

3. Based on derived sequence calculate the number of nucleotides required to polymerize sequence $c^{req}_{nuc,k}$.

4. Elongate up to limits:

if $\texttt{all}(c^{req}_{nuc,k} < c_{nuc,k})$ **then**

Update the position of each ribosome to stop position

$$p_j = s_j$$

else

4a. Attempt to elongate all RNA fragments.

4b. Update position of each polymerase to maximal position given the limitation of $c_{nuc,k}$.

end

5. Update counts of $c_{nuc,k}$ to reflect polymerization usage.

end

`/* Terminate RNA polymerases that have reached the end of their gene */`

for *each RNA polymerase j on gene i* **do**

if $p_j == L_i$ **then**

1. Increment count of RNA that corresponds to elongating RNA transcript that has terminated.

2. Increment free RNA polymerase counts.

end

end

Result: Each RNA transcript is elongated up to the limit of available gene sequence, expected elongation rate, or nucleotide limitation. RNA polymerases that reach the end of their genes are terminated and released.

Associated data

Parameter	Symbol	Units	Value	Reference
Active fraction of RNAP	f_{act}	-	0.20 (growth-dependent)	[216]
RNA synthesis probability[(1)]	p_{synth}	-	[0, 0.015]	See Table C.4
RNAP elongation rate	e	nt/s	50 (growth-dependent)	[216]

Table C.3: Table of parameters for Transcript Initiation and Elongation processes.
[(1)]RNA synthesis probabilities were calculated as the relative fraction of RNA production (which is equal to the RNA degradation) for a given gene.

Associated files

`wcEcoli` Path	File	Type
`wcEcoli/models/ecoli/processes`	`transcript_initiation.py`	process
`wcEcoli/models/ecoli/processes`	`transcript_elongation.py`	process
`wcEcoli/reconstruction/ecoli/dataclasses/process`	`transcription.py`	data

Table C.4: Table of files for transcription.

Transcription regulation

Model implementation

There are two aspects to modeling transcriptional regulation: (1) modeling the activation or inhibition of a transcription factor (e.g., by a ligand), and (2) given an active transcription factor, modeling its effect on RNA polymerase recruitment to a promoter site. We address these topics sequentially below.

Modeling transcription factor activation

We consider three classes of transcription factors based on their mechanism of activation:

1. **One-component systems**: transcription factors that are directly activated or inhibited by a small molecule ligand. Examples of this class include the repressor TrpR which binds tryptophan, and the inducer AraC which binds arabinose.

2. **Two-component systems**: transcription factors that are paired with a separate sensing protein that responds to an environmental stimulus (these are simple analogs to the vast, complicated signaling networks that exist in eukaryotic cells). The sensing protein phosphorylates the cognate transcription factor in a condition-dependent fashion. Examples include

ArcA which is phosphorylated by its cognate ArcB in anaerobic conditions, and NarL which responds to the presence of nitrate when phosphorylated by its cognate sensor NarX.

3. **Zero-component systems**: transcription factors that are considered to be active whenever they are expressed. Examples include the Fis and Hns proteins. These two proteins, for instance, are important in maintaining higher-order DNA structure and likely have complex feedback loops modulating their activity. Because this complexity is not yet fully understood, we make the simplifying assumption that these proteins are always active unless they are knocked out.

One-component systems

For a transcription factor with concentration T whose activity is directly modulated by a ligand with concentration L that binds with stoichiometry n, we assume that the two species achieve equilibrium on a short time scale and that the affinity of the two molecules can be described by a dissociation constant K_d:

$$nL + T \rightleftharpoons T^* \tag{C.8}$$

where T^* represents the concentration of the ligand-bound transcription factor.

With the dissociation constant K_d defined as:

$$K_d = \frac{L^n \cdot T}{T^*} \tag{C.9}$$

we have:

$$\frac{T^*}{T_T} = \frac{L^n}{L^n + K_d} \tag{C.10}$$

where T_T is the total concentration of the transcription factor, both ligand-bound and unbound. As we can see, the fraction of bound transcription factor is a function of ligand concentration and the dissociation constant. Importantly, if the ligand concentration is (approximately) constant over time, the fraction of bound transcription factor is (approximately) constant over time.

To computationally simulate this model we start with total counts of free transcription factor and ligand, completely dissociated from one another. We then form one molecule of the ligand-TF complex at a time and evaluate how close the ratio of $L^n \cdot T/T^*$ is to the actual K_d. We select the values of L, T and T^* that minimize the absolute difference between K_d and $L^n \cdot T/T^*$ (see Algorithm 7).

Two-component systems

For a transcription factor with concentration T; a cognate sensing protein with concentration S; a ligand with concentration L; subscripts f denoting a free (unbound) form of a molecule, b denoting a ligand-bound form of a molecule, and p denoting a phosphorylated form of a molecule; and ATP, ADP, H^+, and H_2O denoting concentrations of these molecules, we propose a system with the following:

Free (unbound) cognate sensing protein at equilibrium with ligand-bound cognate sensing protein, described by dissociation constant K_d:

$$L + S_f \rightleftharpoons S_b \tag{C.11}$$

The autophosphorylation of a free (unbound) cognate sensing protein at a rate k_A:

$$S_f + ATP \overset{k_A}{\rightarrow} S_{fp} + ADP + H^+ \tag{C.12}$$

The autophosphorylation of a ligand-bound cognate sensing protein at a rate k_B:

$$S_b + ATP \overset{k_B}{\rightarrow} S_{bp} + ADP + H^+ \tag{C.13}$$

The phosphorylation of a transcription factor by its free, phosphorylated cognate sensing protein at a rate k_C:

$$S_{fp} + T \overset{k_C}{\rightarrow} S_f + T_p \tag{C.14}$$

The phosphorylation of a transcription factor by its bound, phosphorylated cognate sensing protein at a rate k_D:

$$S_{bp} + T \overset{k_D}{\rightarrow} S_b + T_p \tag{C.15}$$

The auto-phosphatase activity of a transcription factor at a rate k_E:

$$T_p + H_2O \overset{k_E}{\rightarrow} T + P_i \tag{C.16}$$

By assuming mass-action kinetics, we can represent this system mathematically using ordinary differential equations. Ligand binding is simulated in a fashion identical to the one-component systems and the rest of the sub-model is simulated using a numerical ODE integrator (see Algorithm 8).

Zero-component systems

We assume all transcription factors of this class will bind to available promoter sites.

Modeling the modulation of RNA polymerase recruitment

After modeling transcription factor activation, we need to model the probability that the transcription factor is bound to DNA, P_T, and, when the transcription factor is DNA-bound, its effect on RNA polymerase recruitment to the promoter site, Δr (see Algorithm 9). Recalling the notation used in the *Transcription* section (Algorithm 5), we want to modulate the j^{th} entry in the v_{synth} vector of RNA polymerase initiation probabilities such that:

$$v_{\text{synth},j} = \alpha_j + \sum_i P_{T,i} \Delta r_{ij} \tag{C.17}$$

where α_j represents basal recruitment of RNA polymerase and the second term is dependent on transcription factor activity: the probability that the i^{th} transcription factor is DNA-bound is $P_{T,i}$, and the recruitment effect of the i^{th} transcription factor on the j^{th} gene is Δr_{ij}. The α and Δr values are computed prior to simulation based on gene expression values from conditions that modulate transcription factor activity. Values for P_T are calculated as described in Table C.5.

Transcription factor type	Promoter-bound probability
Zero-component system	$P_T = 1$ if TF is present, 0 otherwise
One-component system	$P_T = (T^*)/(T^* + T)$
Two-component system	$P_T = (T_p)/(T_p + T)$

Table C.5: Formulas used to compute the probability that a transcription factor is promoter-bound. T^* is the active form of a one-component system transcription factor, while T_p is the phosphorylated form of a two-component system transcription factor, and T is the inactive or unphosphorylated form of a transcription factor.

Algorithm 7: Algorithm for equilibrium binding

Input : c_m counts of molecules where $m = 1$ **to** $n_{molecules}$

Input : S matrix describing reaction stoichiometries where $S[i, j]$ describes the coefficient for the i^{th} molecule in the j^{th} reaction

Input : K_d^r dissociation constant where $r = 1$ **to** $n_{reactions}$

for *each ligand-binding reaction* j **do**

  1. Dissociate all complexes in c formed by reaction j into constituent molecules

  while *True* **do**

    1. Form complex described by $S[:, j]$

    if $\left| \frac{c_{reactant1}^{S[reactant1,j]} \cdot c_{reactant2}^{S[reactant2,j]} \cdot \ldots \cdot c_{reactantm}^{S[reactantm,j]}}{c_{complex}} - K_d \right|$ *has reached a minimum (i.e., the ratio of reactants to products is as close as possible to the dissociation constant)* **then**

      1. Set reactant and product values in c to these levels

      2. Break out of while loop

    end

  end

end

Result: Ligands are bound to or unbound from their binding partners in a fashion that maintains equilibrium.

Algorithm 8: Algorithm for two-component systems

Input : Δt length of current time step

Input : c_m counts of molecules where $m = 1$ **to** $n_{molecules}$

Input : k_A rate of phosphorylation of free histidine kinase

Input : k_B rate of phosphorylation of ligand-bound histidine kinase

Input : k_C rate of phosphotransfer from phosphorylated free histidine kinase to response regulator

Input : k_D rate of phosphotransfer from phosphorylated ligand-bound histidine kinase to response regulator

Input : k_E rate of dephosphorylation of phosphorylated response regulator

Input : `solveToNextTimeStep()` function that solves two-component system ordinary differential equations to the next time step and returns the change in molecule counts (Δc_m)

1. Solve the ordinary differential equations describing phosphotransfer reactions to perform reactions to the next time step (Δt) using c_m, k_A, k_B, k_C, k_D and k_E.

Δc_m = `solveToNextTimeStep(`c_m, k_A, k_B, k_C, k_D, k_E, Δt`)`

2. Update molecule counts.

$c_m = c_m + \Delta c_m$

Result: Phosphate groups are transferred from histidine kinases to response regulators and back in response to counts of ligand stimulants.

Algorithm 9: Algorithm for transcription factor binding

Input : c_a^i counts of active transcription factors where $i = 1$ **to** $n_{\text{transcription factors}}$
Input : c_i^i counts of inactive transcription factors where $i = 1$ **to** $n_{\text{transcription factors}}$
Input : P_i list of promoter sites for each transcription factor where $i = 1$ **to** $n_{\text{transcription factors}}$
Input : t_i type of transcription factor (either one of two-component, one-component, or zero-component) where $i = 1$ **to** $n_{\text{transcription factors}}$
Input : `randomChoice()` function that randomly samples elements from an array without replacement
for *each transcription factor* **do**
  if *active transcription factors are present* **then**
    1. Compute probability p of binding the target promoter.
    if t_i *is zero-component transcription factor* **then**
      transcription factor present $\rightarrow p = 1$
      transcription factor not present $\rightarrow p = 0$
    else
      $p = \frac{c_a^i}{c_a^i + c_i^i}$
    end
    2. Distribute transcription factors to gene targets.
      $P_i^{bound} =$ `randomChoice(`*from* P_i *sample* $p \cdot len(P_i)$ *elements*`)`
    3. Decrement counts of free transcription factors.
  else
    move on to next transcription factor
  end
end
Result: Activated transcription factors are bound to their gene targets.

Environment dependence: Constitutive and induced transcriptional frequency groups

In the main text, we categorize genes as expressed (1) at least once per cell cycle, (2) less than once per cell cycle, or (3) never expressed over a 32-generation simulation. We also note that for some genes, the categorization is environment-dependent, while for others, the categorization is constitutive (see Figure 5H). Careful readers will note that the model is only optimized for three conditions; this text explains how we could estimate gene expression for many conditions using the transcriptional regulatory data we compiled.

First, we observed that the synthesis probabilities (P_{synth}) and transcriptional frequencies (T_{freq}) of

the genes that are expressed less than once per cell cycle are linearly correlated (Figure 5B). Outside of this linear range, the frequency of observing at least one transcript is bounded at zero (for very low synthesis probabilities) or one (for high synthesis probabilities). Using a piece-wise function interpolated from this linear behavior, we can infer the transcriptional frequency for a given gene with a particular transcript synthesis probability:

$$T_{freq} = \begin{cases} 0 & P_{synth} \leq P_{synth,min} \\ \alpha + \beta \cdot P_{synth,g} & P_{synth,min} < P_{synth} < P_{synth,max} \\ 1 & P_{synth} \geq P_{synth,max} \end{cases} \tag{C.18}$$

Note that α and β are fit with the baseline data in Figure 5B.

The estimation of transcriptional frequencies (P^*_{synth}) under different environmental conditions and genetic perturbations was done as follows:

$$P^*_{synth,g} = FC_g \cdot P_{synth,g} \tag{C.19}$$

where $P_{synth,g}$ is the synthesis probability of a particular gene in our baseline condition, and FC_g is the gene expression fold change measured in the perturbed condition relative to the baseline.

The estimated synthesis probability for a given gene could then be used to compute the estimated transcription frequency (using Equation C.18) corresponding to a different environmental condition.

Using the above model allowed us to (1) use our collection of gene expression profile shifts (described in Associated data) to determine the maximal and minimal fold changes observed for each gene across all conditions, (2) calculate the fold change in expression with respect to our baseline condition (note that we restricted the experiments considered for this analysis to be exclusively those which were directly comparable to our baseline condition), (3) evaluate Equation C.19 to obtain $P^*_{synth,g}$, (4) and subsequently $T^*_{freq,g}$ using Equation C.18, and finally (5) compare the newly-computed $T^*_{freq,g}$ to the original value.

If the new $T^*_{freq,g}$ value is in a different category than the old value, then the gene's expression categorization is environment-dependent. As an example, if a gene is transcribed less than once per cell cycle in our baseline condition, but in another environment a $T^*_{freq,g}$ is calculated that is one or higher, this means that the gene can be induced to the category of more highly-expressed genes (i.e., moving from the blue group in Figure 5B to the red group).

While we recognize that this model might not be accurate for inferring transcriptional frequencies under conditions in which there are significant variations in cell physiology, the results of this analysis investigates how far genes can be induced into or out of the transcription frequency groups identified in Figure 5B. Genes that remained within the same transcription frequency group after being subjected to the linear transformations described are considered to be expressed constitutively at their minimal media frequency (represented by the red, blue, and yellow regions in Figure 5H). On the other hand, genes that were able to be induced into different transcription frequency groups are represented by the purple, orange, and green regions (according to which frequency groups they were able to sample as a result of this analysis). A final note: the total sum of genes shown in Figure 5H (4416 counts) is greater than the total number of genes represented in Figure 5B (4353 genes). This is because 63 genes originated from the blue transcription frequency zone and were found to explore both the "always transcribed" and "never transcribed" groups under different environment conditions, and thus are represented twice: once in the purple region and once in the green region.

Associated files

`wcEcoli` Path	File	Type
`wcEcoli/models/ecoli/processes`	`equilibrium.py`	process
`wcEcoli/models/ecoli/processes`	`tf_binding.py`	process
`wcEcoli/models/ecoli/processes`	`two_component_system.py`	process
`wcEcoli/reconstruction/ecoli/dataclasses/process`	`equilibrium.py`	data
`wcEcoli/reconstruction/ecoli/dataclasses/process`	`transcription_regulation.py`	data
`wcEcoli/reconstruction/ecoli/dataclasses/process`	`two_component_system.py`	data

Table C.6: Table of files for transcription regulation.

Difference from *M. genitalium* model

The most significant difference from the *M. genitalium* model is the enhanced coverage of the regulatory network; 438 regulatory interactions are described by 22 transcription factors that regulate 355 genes. Accordingly, regulation is represented by three different classes of transcription regulators: two-component system, one-component system and zero-component systems. While the phosphotransfer reactions of two-component signaling pathways are modeled in `TwoComponentSystems`, one-component systems (which bind directly to the transcription factor) and zero-component systems (whose presence or absence determines activity) are modeled by the `EquilibriumBinding` and `TranscriptionFactorBinding` Processes.

Associated data

Parameter	Symbol	Units	Value	Reference
Ligand::TF dissociation constant	$k_d = k_r/k_f$	μM	[2e-15, 5e3]	Supp. Materials
Free HK phosphorylation rate	k_A	μM/s	[1e-4, 5e2]	Supp. Materials
Ligand::HK phosphorylation rate	k_B	μM/s	1.7e5	Supp. Materials
Phosphotransfer rate from free HK-P to TF	k_C	μM/s	1e8	Supp. Materials
Phosphotransfer rate from ligand::HK-P to TF	k_D	μM/s	1e8	Supp. Materials
Dephosphorylation rate of TF-P	k_E	μM/s	1e-2	Supp. Materials
DNA::TF dissociation constant	K_d	pM	[2e-4, 1.1e5]	Supp. Materials
Promoter sites	n	targets/chromosome	[1, 108]	Supp. Materials
Fold-change gene expression	FC	$log_2(a.u.)$	[-10.48, 9.73]	Supp. Materials
Gene expression profile shifts	-	shifts	294*	Supp. Materials

Table C.7: Table of parameters for equilibrium binding, two-component systems, and transcription factor binding Processes. HK: histidine kinase, TF: transcription factor, HK-P: phosphorylated histidine kinase, TF-P: phosphorylated transcription factor. *We found 144 pairs of comparable shifts (see Figure 2C).

RNA degradation

Model Implementation

The RNA decay sub-model encodes a molecular simulation of RNA degradation and occurs via two steps that represent RNase-mediated mechanisms. It is implemented in the `RNAdegradation` process (detailed in Algorithm 10).

Endo-nucleolytic Cleavage First, the total counts of RNA degraded during a time step are computed as a fraction of the total capacity for endo-cleavage. Then, the total amount of RNA degraded is divided into different species (mRNA, tRNA, and rRNA) using known endoRNase::RNA affinities. Finally, non-functional RNA fragments are represented as an additional pseudo-metabolite in the `BulkMolecules` state.

Exo-nucleolytic Digestion The exoRNase enzymatic capacity is used to determine the fraction of RNA fragments that can be digested and converted to individual nucleotides that can be recycled by the `Metabolism` process.

Difference from *M. genitalium* model

The *E. coli* model provides a more detailed, mechanistic representation in the `RNADegradation` process compared to the *M. genitalium* model. Unlike the previous model, the gene functionality of endoRNase and exoRNase is mechanistically integrated to evaluate: (1) rates of RNA degradation due to endo-nucleolytic cleavage, and (2) rates of nucleotides digested by exoRNases.

Associated files

`wcEcoli` Path	File	Type
`wcEcoli/models/ecoli/processes`	`rna_degradation.py`	process
`wcEcoli/reconstruction/ecoli/dataclasses/process`	`rna_decay.py`	data

Table C.8: Table of files for RNA degradation.

Associated data

Parameter	Symbol	Units	Value	Reference
EndoRNase catalytic rate	$K_{cat,endo}$	non-functional RNA counts/s	0.10	Supp. Materials
ExoRNase catalytic rate	$K_{cat,exo}$	nt digested/s	50	Supp. Materials
mRNA half-lives[(1)]	τ_{mRNA}	min	[1.30, 31.40]	[217]
tRNA, rRNA half-lives	τ_{tRNA}, τ_{rRNA}	hour	48	[217]
Michaelis constant[(2)]	K_m	RNA counts	-	See Table C.8
RNAse mechanism of action	-	endo-/exo-RNAse	-	Supp. Materials
EndoRNase specificity[(3)]	-	(mRNA, tRNA, rRNA)/RNase	Boolean	Supp. Materials

Table C.9: Table of parameters for RNA degradation process.
[(1)]Non-measured mRNA half-lives were estimated as the average mRNA half-life (5.75 min).
[(2)]Michaelis constants were calculated by fitting the `RNADegradation` model to be equal to the first-order `RNADegradation` model, as follows:
$K_{cat,endo} \cdot c_{endo} \frac{c_{RNA,i}/K_{m,i}}{\sum_j c_{RNA,j}/K_{m,j}} = \frac{ln(2)}{\tau_{RNA,i}} \cdot c_{RNA,i}$

[(3)]Types of RNA that can be targeted by a given RNase.

Algorithm 10: Algorithm for RNA degradation: endo-cleavage for transcripts, and exo-nucleolytic digestion

Input : $K_{m,i}$ Michaelis constants of each mRNA transcript where $i = 1$ **to** n_{RNA}

Input : $K_{cat,endo}$, $K_{cat,exo}$ catalytic rate of endoRNase and exoRNase

Input : c_{endo}, c_{exo} count of endoRNase and exoRNase

Input : $c_{frag,i}$ count of non-functional RNA fragments where $i = 1$ **to** 4 for AMP, CMP, GMP, UMP

Input : c_{mRNA}, c_{tRNA}, c_{rRNA} count of each mRNA, tRNA and rRNA

Input : c_{molec} count of small molecules where $molec \rightarrow$ [H_2O, PPI, Proton, NMPs]

Input : `multinomial()` function that draws samples from a multinomial distribution

Input : `countNTs()` function that returns counts of AMP, CMP, GMP, and UMP for a given non-functional RNA fragment

Input : `lengthFragments()` function that returns the total number of bases of all RNA fragments

`/* Endo-nucleolytic cleveage */`

1. Calculate fraction of active endoRNases (f_i) that target each RNA where $i = 1$ **to** n_{gene}

$$f_i = \frac{\frac{c_{RNA,i}}{K_{m,i}}}{1+\sum \frac{c_{RNA}}{K_m}}$$

2. Calculate total counts of RNAs degraded (R)

$R_{mRNA} = \sum K_{cat,endo} \cdot c_{endo,mRNA} \cdot f_i$ where $i = 1$ **to** n_{mRNAs}

$R_{tRNA} = \sum K_{cat,endo} \cdot c_{endo,tRNA} \cdot f_i$ where $i = 1$ **to** n_{tRNAs}

$R_{rRNA} = \sum K_{cat,endo} \cdot c_{endo,rRNA} \cdot f_i$ where $i = 1$ **to** n_{rRNAs}

where $c_{endo,j}$: number of endoRNases targeting specific species considering endoRNase specificities, $j = 1$ **to** [mRNA, tRNA, rRNA]

3. Sample multinomial distribution D times weighted by endoRNase::RNA affinities to determine which RNAs are converted into non-functional RNAs (d_i)

$d_i =$ `multinomial(`$R, \frac{f_i}{\sum f}$`)`

4. Increase number of RNA fragments. Decrease RNA counts and amount of water required for RNA hydrolysis by endoRNases ($c_{H_2O,endo}$)

$c_{frag} = c_{frag} +$ `countNTs(`d_i`)`

$c_{RNA} = c_{RNA} - d_{RNA}$

$c_{H_2O} = c_{H_2O} - c_{H_2O,endo}$

$c_{PPi} = c_{PPi} + D$

(continued on next page)

`/* Exo-nucleolytic digestion */`

5. Compute exoRNase capacity (E)

$$E = K_{cat,exo} \cdot c_{exo}$$

if $E > \sum c_{frag,i}$ **then**

Update NMPs, water and proton counts

$$c_{NMP} = c_{NMP} + c_{frag}$$

$$c_{H_2O} = c_{H_2O} - \texttt{lengthFragments}(c_{frag})$$

$$c_{proton} = c_{proton} + \texttt{lengthFragments}(c_{frag})$$

Set counts of RNA fragments equal to zero ($c_{frag,i} = 0$)

else

Sample multinomial distribution c_{frag} with equal probability to determine which fragments are exo-digested ($c_{fragDig}$) and recycled

$$c_{fragDig,i} = \texttt{multinomial}(E, \tfrac{c_{frag,i}}{\sum c_{frag}})$$

Update NMPs, water, proton counts, and RNA fragments

$$c_{NMP} = c_{NMP} + c_{fragDig}$$

$$c_{H_2O} = c_{H_2O} - \texttt{lengthFragments}(c_{fragDig})$$

$$c_{proton} = c_{proton} + \texttt{lengthFragments}(c_{fragDig})$$

$$c_{frag} = c_{frag} - c_{fragDig}$$

end

Result: RNAs are selected and degraded by endoRNases, and non-functional RNA fragments are digested through exoRNases. During the process water is consumed, and amino acids are released.

Translation

Model implementation

Translation is the process by which the coding sequences of mRNA transcripts are translated by 70S ribosomes into polypeptides that then fold into proteins. This process accounts for more than two thirds of an *E. coli* cell's ATP consumption during rapid growth [218] and the majority of macromolecular mass accumulation. In the *E. coli* model translation occurs through the action of two processes in the model: `PolypeptideInitiation` and `PolypeptideElongation`.

`PolypeptideInitiation` models the complementation of 30S and 50S ribosomal subunits into 70S ribosomes on mRNA transcripts. Full 70S ribosomes are formed on mRNA transcripts by sampling a multinomial distribution with probabilistic weights calculated from the abundance of mRNA transcripts, and each transcript's translational efficiency (See Algorithm 11). Translational efficiencies were calculated from ribosomal profiling data [219].

`PolypeptideElongation` models the polymerization of amino acids into polypeptides by ribosomes using an mRNA transcript as a template, and the termination of elongation once a ribosome has reached the end of an mRNA transcript. This process is implemented assuming that tRNA charging by synthetases, ternary complex formation (GTP : EF-Tu : charged-tRNA), and ternary complex diffusion to elongating ribosomes are not rate limiting for polypeptide polymerization. Given this assumption this process directly polymerizes amino acids based on the codon sequence of the mRNA transcript. Polymerization occurs across all ribosomes simultaneously and resources are allocated to maximize the progress of all ribosomes up to the limit of the expected ribosome elongation rate in a medium, available amino acids, and available transcripts (see Algorithm 12).

Algorithm 11: Algorithm for ribosome initiation on mRNA transcripts

Input : t_i translational efficiency of each mRNA transcript where $i = 1$ **to** n_{gene}

Input : $c_{mRNA,i}$ count of each mRNA transcript where $i = 1$ **to** n_{gene}

Input : c_{30S} count of free 30S ribosomal subunit

Input : c_{50S} count of free 50S ribosome subunit

Input : `multinomial()` function that draws samples from a multinomial distribution

1. Calculate probability (p_i) of forming a ribosome on each mRNA transcript weighted by the count and translational efficiency of the transcript.

$$p_i = \frac{c_{mRNA,i} \cdot t_i}{\sum_{i=1}^{n_{gene}} c_{mRNA,i} \cdot t_i}$$

2. Calculate maximal number of ribosomes that could be formed.

$$r_{max} = \texttt{min}(c_{30S}, c_{50S})$$

3 Sample multinomial distribution r_{max} times weighted by p_i to determine which transcripts receive a ribosome and initiate ($n_{init,i}$).

$$n_{init,i} = \texttt{multinomial}(r_{max}, p_i)$$

4 Assign $n_{init,i}$ ribosomes to mRNA transcript i. Decrement 30S and 50S counts.

$$c_{30S} = c_{30S} - \sum_{i=1}^{n_{gene}} n_{init,i}$$

$$c_{50S} = c_{50S} - \sum_{i=1}^{n_{gene}} n_{init,i}$$

Result: 70S ribosomes are formed from free 30S and 50S subunits on mRNA transcripts scaled by the count of the mRNA transcript and the transcript's translational efficiency.

Algorithm 12: Algorithm for peptide chain elongation and termination

Input : $e_{expected}$ expected elongation rate of ribosome ($e_{expected} < e_{max}$)
Input : p_i position of ribosome on mRNA transcript $i = 1$ **to** $n_{ribosome}$
Input : δt length of current time step
Input : c_{GTP} counts of GTP molecules
Input : L_j length of each mRNA $j = 1$ **to** n_{gene} for each coding gene.
`/* Elongate polypeptides up to limits of sequence, amino acids, or energy */`
for *each ribosome i on mRNA transcript j* **do**
  1. Based on ribosome position p_i on mRNA transcript and expected elongation rate $e_{expected}$ determine "stop condition" position (t_i) for ribosome assuming no amino acid limitation. Stop condition is either maximal elongation rate scaled by the time step or the full length of sequence (i.e. the ribosome will terminate in this time step).
    $t_i = \texttt{min}(p_i + e_{expected} \cdot \delta t,\ L_j)$
  2. Derive sequence between ribosome position (p_i) and stop condition (t_i).
  3. Based on derived sequence calculate the number of amino acids required to polymerize sequence $c^{req}_{aa,i}$ and number of GTP molecules required c^{req}_{GTP}.
  4. Elongate up to limits:
  if $\texttt{all}(c^{req}_{aa,k} < c_{aa,k})$ ***and*** $c^{req}_{GTP} < c_{GTP}$ **then**
    Update the position of each ribosome to stop position
      $p_i = t_i$
  else
    4a. Attempt to elongate all polypeptide fragments.
    4b. Update position of each ribosome to maximal position given the limitation of $c_{aa,k}$ and c_{GTP}.
  end
  5. Update counts of $c_{aa,k}$ and c_{GTP} to reflect polymerization usage.
end
`/* Terminate ribosomes that have reached the end of their mRNA transcript */`
for *each ribosome i on transcript j* **do**
  if $p_i == L_j$ **then**
    1. Increment count of protein that corresponds to elongating polypeptide that has terminated.
    2. Dissociate ribosome and increment 30S and 50S counts.
  end
end
Result: Each ribosome is elongated up to the limit of available mRNA sequence, expected elongation rate, amino acid, or GTP limitation. Ribosomes that reach the end of their transcripts are terminated and released.

Associated files

wcEcoli Path	File	Type
wcEcoli/models/ecoli/processes	polypeptide_initiation.py	process
wcEcoli/models/ecoli/processes	polypeptide_elongation.py	process
wcEcoli/reconstruction/ecoli/dataclasses/process	translation.py	data

Table C.10: Table of files for translation.

Difference from *M. genitalium* model

The PolypeptideInitiation process is implemented similarly in the *M. genitalium* with a few key differences. As the model of *E. coli* is not yet gene complete, the checks for initiation factors are not present. A major advance over the *M. genitalium* model is that the probability of ribosome initiation on a transcript is now proportional to the product of the mRNA count and its translational efficiency. In the *M. genitalium* model translational efficiency was not taken into account.

The PolypeptideElongation algorithm is implemented similarly to the *M. genitalium* model but again because the *E. coli* model is not yet gene complete, elongation factors are not accounted for. Additionally, tRNAs and their synthetases are not accounted for explicitly. Instead, the model directly polymerizes amino acids. This avoids computational issues with the simulation time step, tRNA pool size, and tRNA over expression that were present in the *M. genitalium* model. There is no implementation of ribosome stalling or tmRNAs. The polymerization resource allocation algorithm is the same as in *M. genitalium*.

Associated data

Parameter	Symbol	Units	Value	Reference
Translational efficiency[(1)]	t_i	RIB/mRNA	[0, 5.11]	[219]
Ribosome elongation rate	e	aa/s	18 (growth-dependent)	[216]
Protein counts (validation data)	$c_{protein}$	protein counts	[0, 250000]	[220]

Table C.11: Table of parameters for translation process.
[(1)]Non-measured translational efficiencies were estimated by the average translational efficiency (1.11 RIB/mRNA).

Protein degradation

Model Implementation

The `ProteinDegradation` process accounts for the degradation of protein monomers. It uses the N-end rule [221] to assign degradation rates for each protein, and selects proteins to be degraded as a Poisson process.

Algorithm 13: Algorithm for Protein Degradation

Input : $t_{1/2,i}$ Protein half-lives for each monomer where $i = 1$ **to** $n_{protein}$

Input : L_i length of each protein monomer where $i = 1$ **to** $n_{protein}$

Input : $c_{aa,i,j}$ count of each amino acid present in the protein monomer where $i = 1$ **to** $n_{protein}$ and $j = 1$ **to** 21 for each amino acid

Input : $c_{protein,i}$ the number of each protein present in the cell

1. Determine how many proteins to degrade based on the degradation rates and counts of each protein.

$$n_{protein,i} = \texttt{poisson}(\frac{ln(2)}{t_{1/2,i}} \cdot c_{protein,i} \cdot \Delta t)$$

2. Determine the number of hydrolysis reactions (n_{rxns}) that will need to occur.

$$n_{rxns} = \sum_i (L_i - 1) \cdot n_{proteins,i}$$

3. Determine the number of amino acids ($n_{aa,j}$) that will be released.

$$n_{aa,j} = \sum_i c_{aa,i,j} \cdot n_{proteins,i}$$

4. Degrade selected proteins, release amino acids from those proteins back into the cell, and consume H_2O that was required for hydrolysis reactions.

Result: Proteins are selected and degraded. During the process water is consumed, and amino acids are released.

Difference from *M. genitalium* model.

The *E. coli* model is not yet gene complete, hence this process does not take into account the activities of specific proteases and does not specifically target prematurely aborted polypeptides. In addition, protein unfolding and refolding by chaperones is not accounted for by this process.

Associated files

`wcEcoli` Path	File	Type
`wcEcoli/models/ecoli/processes`	`protein_degradation.py`	process
`wcEcoli/reconstruction/ecoli/dataclasses/process`	`translation.py`	data

Table C.12: Table of files for protein degradation.

Associated data

Parameter	Symbol	Units	Value	Reference
Protein half-lives	$t_{1/2}$	min	[2, 600]	[221]

Table C.13: Table of parameters for protein degradation process.

Complexation

Model implementation

This process models the formation of all macromolecular complexes except for 70S ribosome formation, which is handled by `Translation`. Macromolecular complexation is done by identifying complexation reactions that are possible (which are reactions that have sufficient counts of all subcomponents), performing one randomly chosen possible reaction, and re-identifying all possible complexation reactions. This process assumes that macromolecular complexes form spontaneously, and that complexation reactions are fast and complete within the time step of the simulation.

Algorithm 14: Algorithm for macromolecular complexation

Input : c_i counts of molecules where $i = 1$ **to** $n_{molecules}$

Input : S matrix describing reaction stoichiometries where $S_{i,j}$ describes the coefficient for the i^{th} molecule in the j^{th} reaction

Input : `getPossibleReactions` function that takes c_i and S and returns all reactions that are possible

Input : `chooseRandomReaction` function that takes all possible reactions and returns one randomly chosen reaction

while *possible reactions remaining* **do**

1. Get all possible reactions (r)

$r =$ `getPossibleReactions`(S, c_i)

2. Choose a random possible reaction (r_{choice}) to perform

$r_{choice} =$ `chooseRandomReaction`(r)

3. Perform r_{choice} by incrementing product counts and decrementing reactant counts

end

Result: Macromolecule complexes are formed from their subunits.

Associated files

wcEcoli Path	File	Type
wcEcoli/models/ecoli/processes	complexation.py	process
wcEcoli/reconstruction/ecoli/dataclasses/process	complexation.py	data

Table C.14: Table of files for complexation.

Associated data

Stoichiometric coefficients that define 1,023 reactions to form protein complexes from EcoCyc [222].

Difference from *M. genitalium* model

This sub-model is implemented very similarly to the *M. genitalium* model of complexation. In the *M. genitalium* simulations, however, the selection of a complexation reaction was weighted by a multinomial distribution parameterized by substrate availability rather than a uniform distribution. We found that the choice of distribution had no major effect on behavior of the process. Additionally, the *M. genitalium* simulations describe 201 macromolecular complexes, whereas over 5 times as many are implemented in the *E. coli* model.

C.3.2 Metabolism

Metabolism

Model implementation

Flux balance analysis (FBA) is a common way to model large-scale metabolic network behavior with a low parameter requirement. However, traditional implementations of FBA are inappropriate for whole-cell modeling due to the dynamic nature of whole-cell simulation and fixed nature of the classic FBA objective function.

To alleviate this we use an alternative objective function that involves a multi-objective minimization for homeostatic metabolite composition and reaction kinetics that extends previous work by Birch *et al.* [223]. The effect of this multi-objective function is twofold: (1) to maintain cellular concentrations of small molecule metabolites and (2) to enforce constraints on metabolic fluxes calculated from Michaelis-Menten kinetics based on metabolite concentrations and curated kinetic parameters. A weighting factor is used to balance the contribution from the two objectives.

We used the metabolic network reconstruction from Karp *et al.* [224] because it was well-connected

to the rest of EcoCyc's resources and data which we relied on. This network reconstruction was based on the Orth model [225]. Different nutrient conditions (minimal M9, +amino acids, -oxygen, etc.) can be specified by changing bounds on metabolite import reactions, and shifts between these nutrient conditions can be programmatically varied.

Homeostatic objective The homeostatic objective attempts to maintain small molecule metabolite concentrations at a constant value. For example, if during a time-step the net effect of other `Process` execution transforms ATP to ADP, the concentration of ATP will be lower and ADP higher. The homeostatic objective ensures that the metabolic network will attempt to increase the ATP concentration and decrease the ADP concentration using chemical transformations available in the network.

A total of 140 metabolite set-point concentrations are specified in the objective ($C_{o,i}$ in Equation C.20). The homeostatic objective minimizes the deviation from these measured concentrations and can be specified as:

$$\text{minimize} \sum_i \left| 1 - \frac{C_i}{C_{o,i}} \right| \tag{C.20}$$

where C_i is the concentration of metabolite i and $C_{o,i}$ is the measured set-point concentration for metabolite i. Cytoplasmic concentrations were chosen based on data from Bennett *et al.* [226], and other components of biomass have set-point concentrations specified based on the overall composition of the cell (lipids, metal ions, etc.) [227] and can be dependent on the media environment of the simulation.

Kinetics objective The *E. coli* model simulates both metabolic enzyme expression via transcription and translation and dynamically maintains 140 metabolite concentrations. This enables the use of Michaelis-Menten kinetic equality constraints on metabolic fluxes using Equation C.21:

$$v_{o,j} = k_{cat} \cdot E \cdot \left(\frac{C_1}{C_1 + K_{m,1}} \right) \cdot \left(\frac{C_2}{C_2 + K_{m,2}} \right) \ldots \left(\frac{C_n}{C_n + K_{m,n}} \right) \tag{C.21}$$

where $v_{o,j}$ is the kinetic target for the flux through reaction j that has n substrates, k_{cat} is the catalytic turnover rate for enzyme E, $K_{m,n}$ is the saturation constant for substrate n, E is the concentration of metabolic enzyme, and C_n is the concentration of substrate n in reaction j.

Kinetics data was reviewed from over 12,000 papers identified from BRENDA [228]. We filtered out papers that did not have a k_{cat}, which did not use a lab strain, or which did not involve enzymes in our metabolic network. The result was roughly 1200 papers which we manually curated due

to our and others' observation that about 20% of the values in the BRENDA database are copied incorrectly from their primary source papers [128]. From this set, 181 constraints with a K_m and k_{cat} and 219 constraints with only a k_{cat} are used to constrain a total of 340 reactions (with some reactions having multiple constriants). Although some additional constraints were identified, they are not currently being used in the model. In particular, constraints were found for tRNA charging (18 reactions) but not used since tRNA charging is not explicitly included in the model. Additionally, contraints for two reactions involved in the Citric Acid cycle (succinate dehydrogenase and fumurate reductductase) were identified that, when enabled, caused a much higher glucose uptake rate than observed without kinetic constraints and higher than what has been experimentally measured. Based on this, we excluded these constraints from the model.

In cases where the enzyme parameters were recorded at non-physiological temperatures, we used the following scaling factor to adjust the k_{cat}:

$$2^{\frac{37-T}{10}} \tag{C.22}$$

where T is the reported temperature (in °C) for the experimental conditions—this increases the kinetic rate by a factor of 2 for every 10°C away from 37°C.

Similar to the homeostatic objective, the kinetics objective minimizes the deviation from a kinetic target that is calculated at each time step based on the enzyme and metabolite concentrations. Formally:

$$\text{minimize} \sum_j \left| 1 - \frac{v_j}{v_{o,j}} \right| \tag{C.23}$$

where v_j is the flux through reaction j and $v_{o,j}$ is the target flux for reaction j calculated from Equation C.21.

Difference from *M. genitalium* model

We have made a number of improvements to the metabolic sub-model compared to the *M. genitalium* implementation. In the *M. genitalium* simulations, metabolites were produced in a fixed ratio at every time step regardless of the behavior of the rest of the simulated cell—this could lead to pooling or depletion of metabolites. Furthermore, if one metabolite could not be produced, none of the metabolites could be produced. Our homeostatic objective fixes both of these shortcomings. In addition, we have quantitative data and a method to softly constrain 340 reactions.

Associated data

Parameter	Symbol	Units	Value	Reference
Metabolic network	S	-	Stoichiometric coefficients	[224]
Metabolic target fluxes	v_o	$\mu M/s$	[0, 87000]	See Table C.3.2
Metabolic fluxes (validation)	v_v	$\mu M/s$	[82, 1500]	[229]
Enzyme turnover number	k_{cat}	$1/s$	[0.00063, 38000]	Supp. Materials
Enzyme Michaelis constant	K_m	μM	[0.035, 550000]	Supp. Materials
Metabolite target concentration	C_o	μM	[0.063, 97000]	[226]

Table C.15: Table of parameters for metabolism process.

Algorithm 15: Algorithm for Metabolism

Input : C_i concentration for metabolite i

Input : $C_{o,i}$ concentration target for metabolite i

Input : $k_{cat,j}$ turnover number for enzyme j

Input : $K_{m,i,j}$ Michaelis constant for metabolite i for enzyme j

Input : E_j concentration for enzyme j

Input : S stoichiometric matrix for all reactions

1. Set physical constraints on reaction fluxes

For all reactions: $v_{min,j} = -\inf, v_{max,j} = +\inf$

For thermodynamically irreversible reactions: $v_{min,j} = 0$

If required enzyme not present: $v_{min,j} = v_{max,j} = 0$

2. Calculate kinetic target ($v_{o,j}$) for each reaction j based on the enzymes j and metabolites i associated with each reaction

$$v_{o,j} = k_{cat,j} \cdot E_j \cdot \prod_i \left(\frac{C_i}{K_{m,i,j} + C_i}\right)$$

3. Solve linear optimization problem

$$\text{minimize} \sum_i \left|1 - \frac{C_i}{C_{o,i}}\right| + \lambda \sum_j \left|1 - \frac{v_j}{v_{o,j}}\right|$$

$$\text{subject to} \quad S \cdot v = 0$$

$$v_j \geq v_{min,j}$$

$$v_j \leq v_{max,j}$$

4. Update concentrations of metabolites based on the solution to the linear optimization problem

Result: Metabolites are taken up from the environment and converted into other metabolites for use in other processes

Associated files

wcEcoli Path	File	Type
wcEcoli/models/ecoli/processes	metabolism.py	process
wcEcoli/reconstruction/ecoli/dataclasses/process	metabolism.py	data

Table C.16: Table of files for metabolism.

Energy requirements of cell maintenance

As was the case in our *M. genitalium* simulations, and in many flux balance analysis models, not all of the energy consumed by metabolic pathways, macromolecular polymerization, or other growth and non-growth associated processes is accounted for explicitly in our *E. coli* model. This is primarily due to a lack of experimental data and/or knowledge accounting for its usage. Similar to the *M. genitalium* model, we have incorporated reactions in the metabolic model with two parameters, Growth Associated Maintenance (GAM) and Non-Growth Associated Maintenance (NGAM), which modulate energy consumption associated with growth and cell maintenance.

Associated data

Parameter	Symbol	Units	Value	Reference
Growth associated maintenance	GAM	mmol/g	59.81	[230]
Non-growth associated maintenance	NGAM	mmol/g/h	8.39	[230]

Table C.17: Table of parameters for energy requirements of cell maintenance.

C.3.3 Balanced growth

Chromosome replication

Model implementation

Chromosome replication occurs through three steps that are implemented in the `ChromosomeFormation` and `ChromosomeElongation` processes. First, a round of replication is initiated at a fixed cell mass per origin of replication and generally occurs once per cell cycle (see Algorithm 16). Second, replication forks are elongated up to the maximal expected elongation rate, dNTP resource limitations, and template strand sequence (see Algorithm 17). Finally, replication forks terminate once they reach the end of their template strand and the chromosome immediately decatinates forming two

separate chromosome molecules (see Algorithm 18).

Algorithm 16: Algorithm for DNA replication initiation

Input : m_{cell} cell mass
Input : $m_{critical}$ critical initiation mass
Input : n_{origin} number of origins of replication
Input : $n_{fork,f}$ number of replication forks on forward strand
Input : $n_{fork,r}$ number of replication forks on reverse strand
Input : $n_{chromosome}$ number of chromosome molecules
Input : C length of C period
Input : D length of D period
if $\frac{m_{cell}}{n_{origin}} > m_{critical}$ **then**
  if $n_{origin} > 1$ **then**
    $n_{origin} = n_{origin} + \frac{n_{fork,f}+n_{fork,r}}{2} \cdot n_{chromosome}$
  else
    $n_{origin} = n_{origin} + n_{chromosome}$
  end
  $n_{fork,f} = n_{fork,f} + n_{fork,f} \cdot n_{chromosome}$
  $n_{fork,r} = n_{fork,r} + n_{fork,r} \cdot n_{chromosome}$
end
Result: When cell mass is larger than critical initiation mass m_c another round of replication is initiated with correct number of replication forks

Algorithm 17: Algorithm for DNA replication elongation

Input : e maximal elongation rate of replication fork

Input : p_i position of forks on chromosome where $i = 1$ **to** n_{fork}

Input : δt length of current time step

Input : $c_{dNTP,j}$ counts of dNTP where $j = 1$ **to** 4 for dCTP, dGTP, dATP, dTTP

Input : L_k total length of each strand of chromosome from origin to terminus where $k = 1$ **to** 4 for forward/complement and reverse/complement.

for *each replication fork i on sequence k* **do**

1. Based on replication fork position p_i and maximal elongation rate e determine "stop condition" (s_i) for replication fork assuming no dNTP limitation.

$$s_i = \texttt{min}(p_i + e \cdot \delta t,\ L_k)$$

Stop condition is either maximal elongation rate scaled by the time step or the full length of sequence (i.e. the fork will terminate in this time step).

2. Derive sequence between replication fork position (p_i) and stop condition (s_i).

3. Based on derived sequence calculate the number of dNTPs required to polymerize sequence $c^{req}_{dNTP,i}$.

4. Elongate up to limits:

if $\texttt{all}(c^{req}_{dNTP,i} < c_{dNTP,j})$ **then**

Update the position of each replication fork to stop position

$$p_i = s_i$$

else

Attempt to equally elongate each replication fork update position of each fork to maximal position given the limitation of $c_{dNTP,j}$.

end

5. Update counts of $c_{dNTP,j}$ to reflect polymerization usage.

end

Result: Each replication fork is elongated up to the limit of available sequence, elongation rate, or dNTP limitation

Algorithm 18: Algorithm for DNA replication termination

Input : p_i position of forks on chromosome where $i = 1$ **to** n_{fork}
Input : L_k total length of each strand of chromosome from origin to terminus where $k = 1$ **to** 4 for forward/complement and reverse/complement
Input : d_{queue} a double ended queue data structure that stores time(s) cell division should be triggered
Input : D D-period of cell cycle (time between completion of chromosome replication and cell division)
Input : t Current simulation time
for *each replication fork i on strand k* **do**
  if $p_i == L_k$ **then**
    1. Delete replication fork
    2. Divide remaining replication forks and origins of replication appropriately across the two new chromosome molecules
    3. Calculate time cell should trigger division based on current time of chromosome termination and push onto queue data structure
      d_{queue}.`push`$(t + D)$
  end
end
Result: Replication forks that have terminated are removed. A new chromosome molecule is created separating all remaining replication forks. Timer for D-period is started.

Associated files

`wcEcoli` Path	File	Type
`wcEcoli/models/ecoli/processes`	`chromosome_formation.py`	process
`wcEcoli/models/ecoli/processes`	`chromosome_elongation.py`	process
`wcEcoli/reconstruction/ecoli/dataclasses/process`	`replication.py`	data

Table C.18: Table of files for chromosome replication.

Difference from *M. genitalium* model

The physiology modeled is significantly different from what was implemented in the *M. genitalium* model. Initiation of DNA replication in *E. coli* no longer uses a DnaA based mechanistic model but instead uses a phenomenological model based on a constant mass per origin of replication triggering DNA replication initiation. The action of topoisomerases are not explicitly modeled. Replication forks no longer take into account all of the enzymes in the replisome but are point objects that traverse the chromosome sequence. Some differences exist because the *E. coli* model is not yet a

gene complete model. More importantly, certain changes enabled significant modeling advances in the *E. coli* model. These include modeling the DNA replication cycle over multiple growth rates, cell sizes, and conditions using a single unified framework, and enabling multiple rounds of replication to proceed simultaneously over multiple generations. Both advances were critical to the findings in this publication.

Associated data

Parameter	Symbol	Units	Value	Reference
Chromosome sequence	-	-	-	[231]
Replication fork elongation rate	e	nt/s	967	[232]
Mass per origin at DNA replication initiation	$m_{critical}$	origin/fgDW	[600,975]	Semi-quantitative fit [233]
C period	C	min	40	[234]
D period	D	min	20	[234]

Table C.19: Table of parameters for chromosome replication process.

Cell division

Model implementation

Cell division is modeled in the `ChromosomeElongation` process and `CellDivision` listener in the *E. coli* model. A Helmstetter-Cooper type model of chromosome replication initiation is coupled to cell division, inspired by work from Wallden *et al.* [235]. Chromosome replication initiation occurs at a fixed mass per origin of replication. Each initiation event is coupled to a cell division event after a constant period of time consisting of one round of chromosome replication and cytokinesis. Importantly, this constant period of time can span multiple cell division events.

Cell division itself is modeled as a binomial process where each daughter cell has an equal probability of inheriting the contents of the mother cell. The exception to this is if two chromosomes are present before cell division—each daughter is guaranteed to get one.

Algorithms 18 and 19 provide implementation details.

Algorithm 19: Algorithm for cell division

Input : d_{queue} a double ended queue data structure that stores time(s) cell division should be triggered

Input : c_i counts of all molecules in simulation at cell division where $i = 1$ **to** $n_{species}$

Input : p binomial partition coefficient

Input : n_{chrom} number of chromosome molecules

Input : `rand()` returns a random number from a uniform distribution between 0 and 1

Input : `randint()` returns a random integer either 0 or 1

if $t > d_{queue}$`.peek()` **then**

1. Trigger division and remove division time.

d_{queue}`.pop()`

2. Divide bulk contents of cell binomially. Number partitioned into daughter one is stored in $n_{daughter,1}$ and to daughter two in $n_{daughter,2}$.

for $i = 1$ **to** $n_{species}$ **do**

$n_{daughter,1} = 0$

for $j = 1$ **to** c_i **do**

if `rand()` $> p$ **then**

$n_{daughter,1} = n_{daughter,1} + 1$

end

end

$n_{daughter,2} = c_i - n_{daughter,1}$

end

3. Divide chromosome in binary manner. All replication forks and origins of replication associated with a chromosome molecule are partitioned as well. Number of chromosome molecules partitioned into daughter one is stored in $n_{chrom,daughter,1}$ and to daughter two in $n_{chrom,daughter,2}$.

if `mod(`n_{chrom}`,2)` **then**

$n_{chrom,daughter,1} = \frac{n_{chrom}}{2}$

else

$n_{chrom,daughter,1}$ =`floor(`$\frac{n_{chrom}}{2}$`)` + `randint()`

end

$n_{chrom,daughter,2} = n_{chrom} - n_{chrom,daughter,1}$

end

Result: Cell division is triggered at C+D time after DNA replication initiation. Contents of mother cell is divided between two daughter cells conserving mass.

Due to the interaction of the algorithms for Chromosome Replication and Cell Division, we occasionally obtain outliers when examining initial and added cell masses (clipped distributions are shown

in Figure 4F in the main text). Figures C.3 and C.4 show histograms of these distributions (the same data for Figure 4F, no clipping) and where the outliers fall relative to the x- and y-limits in Figure 4F—the dashed lines in each figure demarcate the x- and y- limits of the plots in Figure 4F. For example, in Figure C.3, the "Glucose minimal + 20 amino acids" histogram has dashed lines at 0.6 and 1.2, and in Figure C.4 has dashed lines at 0.45 and 1.5—these are the x- and y- limits, respectively, for that same plot in Figure 4F. We see that the outliers constitute at most 1-2% of the data.

Associated files

`wcEcoli` Path	File	Type
`wcEcoli/models/ecoli/processes`	`replication_elongation.py`	process
`wcEcoli/models/ecoli/listeners`	`cell_division.py`	listener

Table C.20: Table of files for transcription regulation.

Difference from *M. genitalium* model

The *E. coli* model is not yet a gene complete model and many of the mechanistic details of cell division are not implemented as they were in the *M. genitalium* model. In *E. coli*, cytokinesis, septation, and chromosome segregation are all not modeled explicitly. However, cell division in our *E. coli* model is consistent with growth at multiple growth rates, which was not the case in *M. genitalium*.

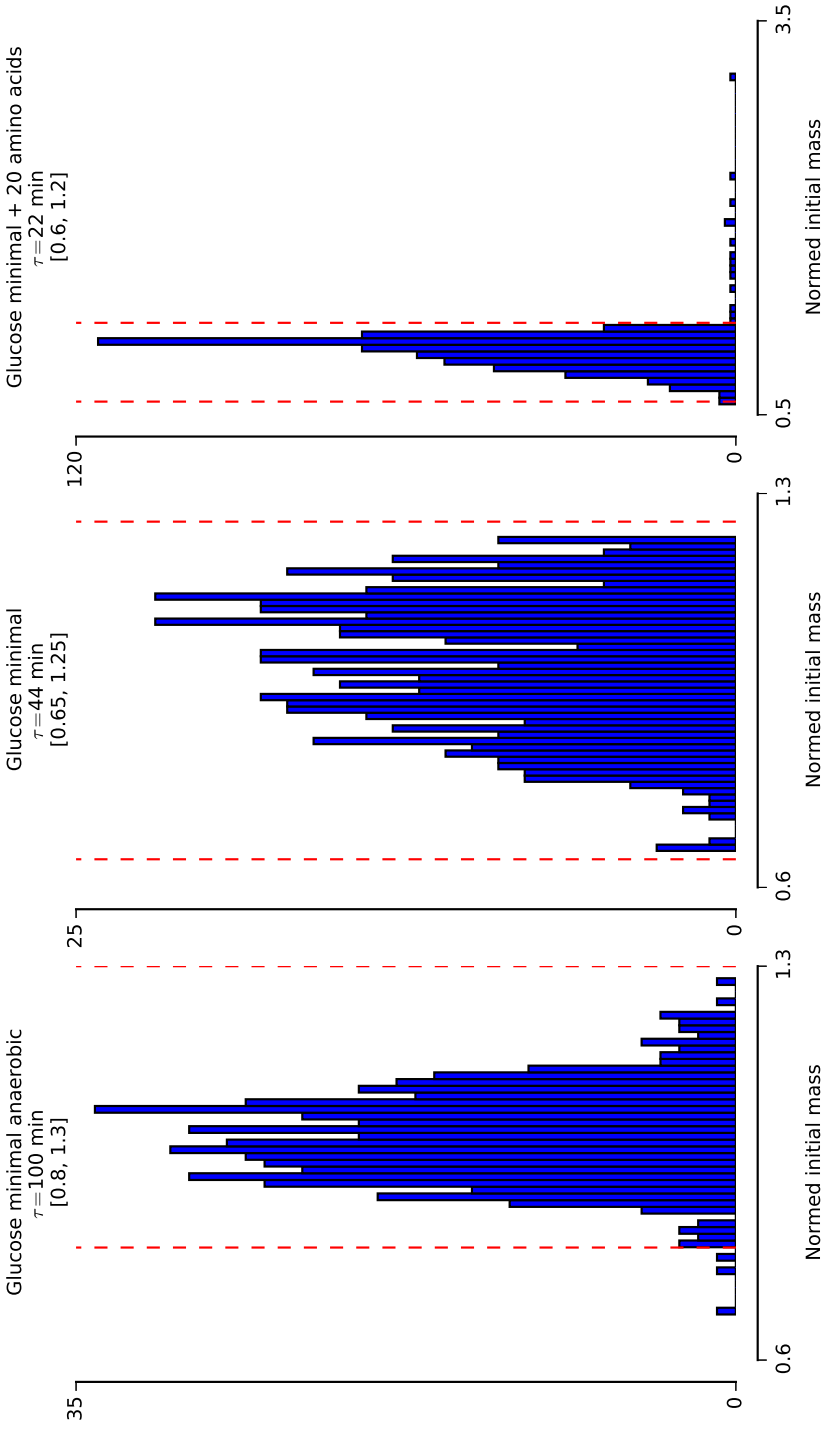


Figure C.3: Distributions of initial cell mass in 3 conditions. Red dashed lines indicate where the x-axis limits for Figure 4F (in the main text) fall relative to these distributions.

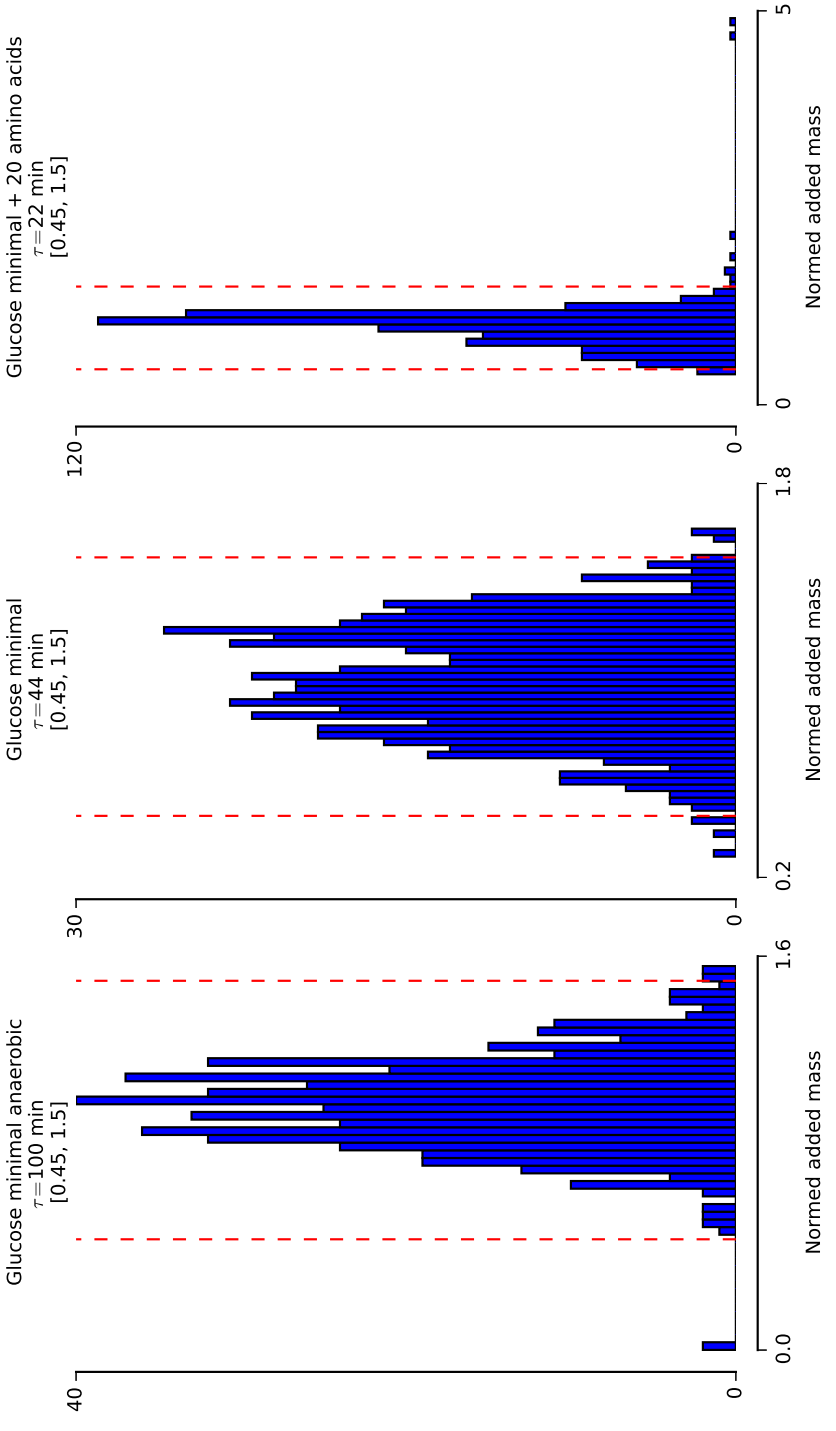


Figure C.4: Distributions of added cell mass in 3 conditions. Red dashed lines indicate where the y-axis limits for Figure 4F (in the main text) fall relative to these distributions.

C.4 Experimental procedures

Below are the materials and methods used to culture *E. coli* K-12 MG1655, perform RNA sequencing, and measure protein half-lives.

C.4.1 RNA sequencing

RNA sequencing was performed to characterize mRNA expression under a variety of environmental conditions. The contributions of this dataset is highlighted most in our evaluation of the Central Dogma model in the main text, but is integral to the function of all processes related to mRNAs.

Materials

Buffers

Stock Solution	**Volume (mL)**
alanine (free)	5
arginine (free)	65
asparagine (free)	40
aspartic acid (free)	5
cysteine (free)	50
glutamic acid (K monohydrate)	5
glutamine (free)	25
glycine (free)	5
histidine (free)	5
isoleucine (free)	5
leucine (free)	10
lysine (HCl)	80
methionine (free)	5
phenylalanine (free)	5
proline (free)	40
serine (free)	5
threonine (free)	40
tryptophan (free)	5
tyrosine (free)	125
valine (free)	5
VA Vitamin Solution	50
H_2O	350
TOTAL	1000

Table C.21: 5x Amino acids, Mix solutions together. Preparation of stock solutions is described in Table C.22
Filter sterilize with 0.2 uM filter. Aliquot and Freeze at -20° C.

Name	**FW**	**Weight (g)**	**Vol (mL)**
alanine (free)	89.09	1.78	25
arginine (free)	174.2	6.98	100
asparagine (free)	132.1	0.66	100
aspartic acid (free)	133.1	1.33	25
cysteine (free)	121.16	0.06	50
glutamic acid (K monohydrate)	203.23	3.05	25
glutamine (free)	146.20	1.8	100
glycine (free)	75.07	1.5	25
histidine (free)	155.15	0.78	25
isoleucine (free)	131.1	0.65	25
leucine (free)	131.2	0.66	100
lysine (HCl)	182.7	7.32	100
methionine (free)	149.2	0.75	25
phenylalanine (free)	165.2	0.83	100
proline (free)	115.10	1.15	25
serine (free)	105.1	8.4	100
threonine (free)	119.1	119	25
tryptophan (free)	204.2	0.26	25
tyrosine (free)	181.2	0.225	125
valine (free)	117.2	0.88	25

Table C.22: Amino Acid Stocks, Make each amino acid separately. Store at -20°C.

	Formula	**5x Salts**
Sodium Phosphate Dibasic (g)	Na_2HPO_4	8.475
Potassium Phosphate Monobasic (g)	KH_2PO_4	3.75
Sodium Chloride (g)	NaCl	0.625
Ammonium Chloride (g)	NH_4Cl	1.25

Table C.23: Salts, bring up to 250 mL each. Autoclave, aliquot into sterile 50 mL tubes.

	M9 Minimal Glucose	M9 Minimal Glucose + AAs
Volume (mL)	100	100
5x Salts (mL)	20	20
5x Amino acids (mL)	0	20
1 M $MgSO_4$ (μL)	200	200
1 M $CaCl_2$ (μL)	10	10
20% glucose (mL)	2	2
H_2O (mL)	77.8	57.8

Table C.24: Media Formulas

Methods

Cell Growth

Grow cells in requisite media overnight in 37°C incubator. In the morning, inoculate fresh cultures to an OD_{600} 0.02 in 10 mL of media contained in a 125 mL flask. Grow cultures on shaker in 37°C warm room.

RNA Extraction

Harvest cells at OD 0.4. Take 2 mL of cell culture and extract using Quiagen RNAeasy Protect Bacterial Mini Kit (Qiagen #74524) with RNAprotect Bacteria Reagent (# 76506) and RNase-Free DNase Set (# 79254) according to manufacturer's instructions. Performed extraction using lysozyme from ThermoScientific #90082. Measured RNA concentration using NanoDrop.

rRNA Cleanup

Remove rRNA from sample using RiboZero rRNA Removal Kit (Epicentre # MRZGN126), according to manufacturer's instructions.

RNA Quality

RNA quality was assessed using an Agilent 2100 BioAnalyzer (# G2938C), according to manufacturer's instructions by the SFGF facility at Stanford University.

cDNA prep and Sequencing

Library prep was performed by the SFGF facility at Stanford University according to the TruSeq Stranded Total RNA Sample Preparation Guide. Paired-end sequencing with read lengths of 75 bp was performed by the SFGF facility at Stanford University on an Illumina NextSeq 500. Approximately 20 million reads were obtained per sample.

Data availability

Sequencing data is available at GEO with accession number GSE85472.

Analysis

bbmap 34.33 [106] was used to pre-process sequencing data to trim reads, remove reads for common contaminants, and remove reads that map to non-coding RNA. RSEM 1.2.19 [107] was used for downstream processing and calculation of gene expression.

C.4.2 Protein half-life measurement

We tested the Central Dogma model under steady-state conditions, under which we expected that the rate of protein synthesis should equal the rate of decay. As shown in Figure 2G, this proved

to largely be the case in our simulations, with most of the production rates within an order of magnitude of the decay rate. We wondered whether some of the outliers in our comparison might be due to a more nuanced or specific value for the protein half-life. For that, we experimentally determined the half-lives of four well-characterized proteins (RpoH, RcsA, HelD, and PssA), and six protein outliers (DcuR, BioD, Rph, CarA, Pnp, and GshA). As illustrated in Figure 2H, we found that in all cases, the half-life predicted by our model was a better predictor of the data than the N-end rule.

Methods

Cell Growth

His-tagged gene plasmids (from the ASKA library without GFP, [236]) were transformed into MG1655. Duplicates of bacterial cultures were grown overnight at 37°C in M9 minimal media with 0.4% glucose and 20 μg/ml chloramphenicol for plasmid selection. In the morning, bacterial cultures were diluted to OD 0.03, and incubated at 37°C until they reached OD 0.3. At this point, bacterial cultures were diluted 1:2 in minimal media supplemented with IPTG (0.1mM) to induce protein over-expression. Cultures were grown on a shaker in a 37°C warm room for the requisite time of induction.

Time-course sampling

After the requisite time of IPTG induction (see Table C.25), a 9 mL sample was taken to measure the time 0 protein level. Then, 10 μg/mL tetracycline (the 10 mg/mL stock was made in 95% ethanol) was added to the rest of the culture to inhibit protein synthesis. Culture was then returned to 37°C. 9 mL samples were taken at indicated time points (10 min on ice followed by centrifugation for 10 min at 4000g, 4°C) to measure protein levels (see C.25. At each time point, culture OD was measured.

Cell lysates

To lyse cells we used BugBuster Master mix (Millipore, #71456-3) supplemented with Halt protease/phosphatase inhibitor cocktail (Thermo Scientific #78444), following the manufacturer's instructions.

Protein quantification

For protein quantification we used Pierce BCA protein assay kit (Thermo Scientific #23225), following the manufacturer's instructions.

Protein detection

For western blot detection of proteins we ran samples on a Simon machine (Protein Simple) using an

anti-His tag antibody (Novus Biologicals #NB100-64768) for protein detection and an anti-RNAP β antibody (BioLegend #663006) for capillary normalization (loading control).

Analysis

The measured amount of His-tagged protein (N_p) was normalized by the amount of β subunit RNAP loaded as a protein control ($RNAP$), i.e. $N = N_p/RNAP$; and then log-transformed. Linear regression was used to determine the first-order decay rate constant (k_d) as follows:

$$N = N_0 \exp(-k_d t),\ \log(N) = \log(N_0) - k_d t \tag{C.24}$$

Then, half-lives (see Figure C.5 and Figure C.6) were estimated by: $t_{1/2} = \frac{\log(2)}{k_d}$.

Gene	**ASKA Id**	$t_{1/2}$	**Ind.**[1] **(h)**	**Protein**[2] **(μg)**	**Ab. dilution**[3]	**Time points**[4]
RcsA	JW1935	25 min (c*)	2	10	1:100, 1:200	**0, 2, 5, 10, 30** min
RpoH	JW3426	25 min (c*)	2	10	1:100, 1:200	**0, 5, 10, 20**, 40 min
PssA	JW2569	10 h (c**)	2	1	1:100, 1:50	**0, 7, 18** h
HelD	JW0945	10 h (c**)	2	4	1:100, 1:100	**0, 1**, 3, 4, 7, 23 h
CarA	JW0030	2 min***	2	2	1:100, 1:100	**0, 0.17, 1, 4, 18** h
GshA	JW2663	2 min***	1	1	1:100, 1:100	**0, 0.17, 1, 4, 18** h
Pnp	JW5851	2 min***	1	4	1:100, 1:100	**0, 18, 24, 48** h
DcuR	JW4085	10 h	2	2	1:100, 1:100	**0, 0.5, 2.5**, 5.5, 18 h
BioD	JW0761	10 h	2	4	1:100, 1:100	**0, 0.67, 2**, 4, 18 h
Rph	JW3618	10 h	2	4	1:100, 1:100	**0, 0.5, 2.5**, 5.5, 18 h

Table C.25: Parameters used for half-life measurements. *Short half-lives that are well characterized in the literature ([1], [2]). **Control proteins with minimal model discrepancies (half-life with highest confidence = 10 h). ***Short half-lives due to N-end rule, which dictates low stability if protein N-terminal is leucine. [1]IPTG induction used to over-express protein levels. [2]Amount of protein loaded for protein detection. [3]Antibody dilution in anti-His, anti-RNAP. [4]Note that time points highlighted in bold were used to calculate decay constant.

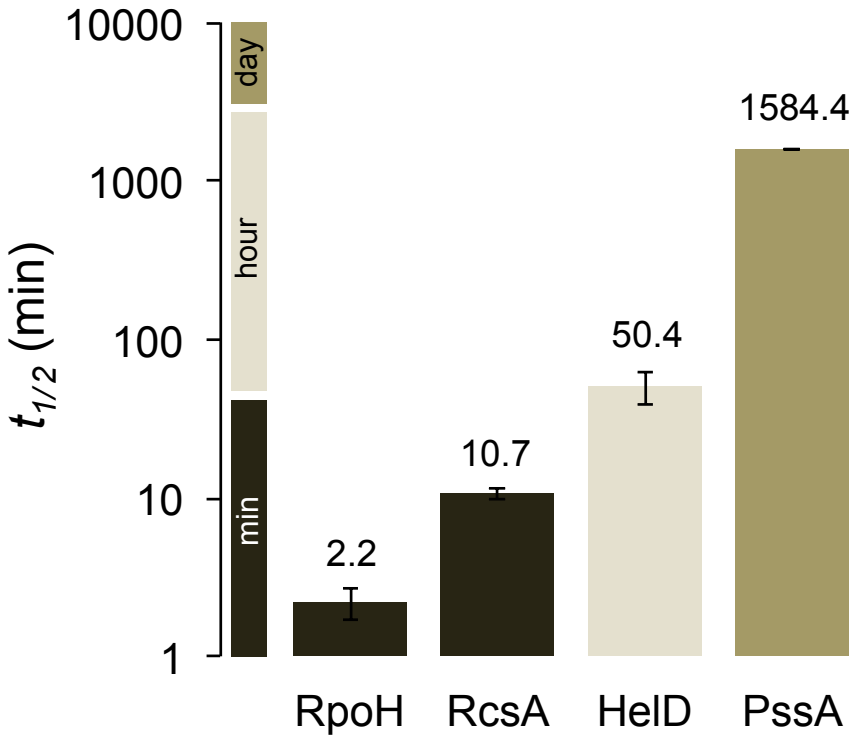


Figure C.5: **Protein half-lives measured for well characterized proteins (RpoH, RcsA), and control proteins with minimal model discrepancies (half-life with highest confidence = 10h).**

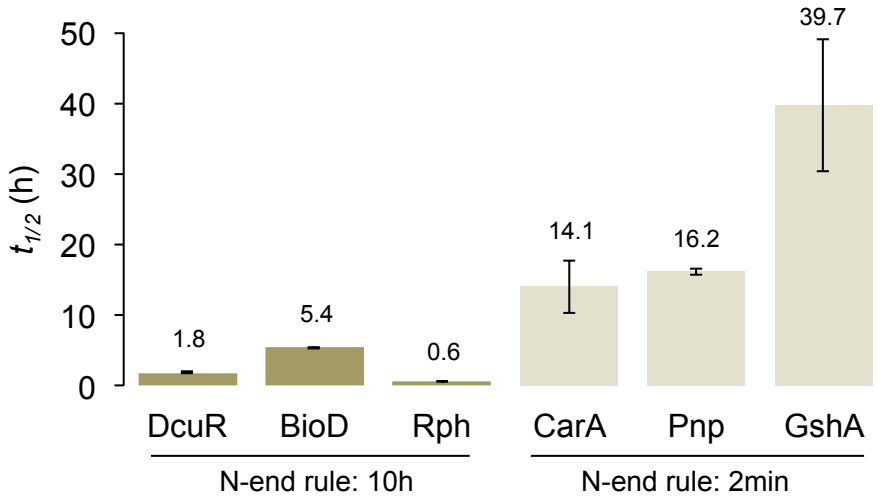


Figure C.6: **Protein half-lives measured.**

Bibliography

[1] Maurizi MR. Proteases and protein degradation in escherichia coli. *Experientia*, 48(2):178–201, 1992.

[2] Kamalendu Nath and Arthur L. Koch. Protein degradation in *Escherichia coli* : I. measurement of rapidly and slowly decaying components. *Journal of Biological Chemistry*, 245(11):2889–2900, 1970.

[3] Nathaniel D Maynard, Elsa W Birch, Jayodita C Sanghvi, Lu Chen, Miriam V Gutschow, and Markus W Covert. A forward-genetic screen and dynamic analysis of lambda phage host-dependencies reveals an extensive interaction network and a new anti-viral strategy. *PLoS Genet*, 6(7):e1001017, 2010.

[4] Tomoya Baba, Takeshi Ara, Miki Hasegawa, Yuki Takai, Yoshiko Okumura, Miki Baba, Kirill A Datsenko, Masaru Tomita, Barry L Wanner, and Hirotada Mori. Construction of *Escherichia coli* K-12 in-frame, single-gene knockout mutants: the keio collection. *Molecular systems biology*, 2(1), 2006.

[5] Nathaniel D Maynard, Derek N Macklin, Karla Kirkegaard, and Markus W Covert. Competing pathways control host resistance to virus via trna modification and programmed ribosomal frameshifting. *Molecular systems biology*, 8(1):567, 2012.

[6] Theresa-Marie Rhyne, Melanie Tory, Tamara Munzner, Matthew O Ward, Chris Johnson, and David H Laidlaw. Information and scientific visualization: Separate but equal or happy together at last. In *IEEE Visualization*, volume 3, pages 611–614. Seattle, 2003.

[7] Jonathan R. Karr, Jayodita C Sanghvi, Derek N Macklin, Miriam V Gutschow, Jared M Jacobs, Benjamin Bolival Jr., Nacyra Assad-Garcia, John I Glass, and Markus W Covert. A Whole-Cell Computational Model Predicts Phenotype from Genotype. *Cell*, 150(2):389–401, July 2012.

[8] Jonathan R Karr, Nolan C Phillips, and Markus W Covert. Wholecellsimdb: a hybrid relational/hdf database for whole-cell model predictions. *Database*, 2014:bau095, 2014.

[9] Ruby Lee, Jonathan R. Karr, and Markus W Covert. WholeCellViz: data visualization for whole-cell models. *BMC bioinformatics*, 14(1):253, 2013.

[10] Jayodita C. Sanghvi, Sergi Regot, Silvia Carrasco, Jonathan R. Karr, Miriam Gutschow, Benjamin Jr. Bolival, and Markus W. Covert. Accelerated discovery via a whole-cell model. *Nature Methods*, 10(12):1192–1195, 2013.

[11] Zachary D Blount. The unexhausted potential of *E. coli*. *Elife*, 4:e05826, 2015.

[12] Jacob J Hughey, Miriam V Gutschow, Bryce T Bajar, and Markus W Covert. Single-cell variation leads to population invariance in nf-κb signaling dynamics. *Molecular biology of the cell*, 26(3):583–590, 2015.

[13] Derek N Macklin, Haisam Islam, and Jonathan Lu. Good cell, bad cell: Classification of segmented images for suitable quantification and analysis. `http://cs229.stanford.edu/proj2012/MacklinIslamLu-GoodCellBadCellClassificationOfSegmentedImagesForSuitableQuantificationAndAnalysis.pdf`. Accessed: 2017-01-04.

[14] David A Van Valen, Takamasa Kudo, Keara M Lane, Derek N Macklin, Nicolas T Quach, Mialy M DeFelice, Inbal Maayan, Yu Tanouchi, Euan A Ashley, and Markus W Covert. Deep learning automates the quantitative analysis of individual cells in live-cell imaging experiments. *PLoS Comput Biol*, 12(11):e1005177, 2016.

[15] Derek N Macklin and David A Van Valen. Deep cell docker image. `https://hub.docker.com/r/vanvalen/deepcell/`. Accessed: 2017-01-04.

[16] Keara M Lane, David A Van Valen, Mialy M DeFelice, Derek N Macklin, Ariel Jaimovich, Ambrose Carr, Tobias Meyer, Dana Pe'er, Stephane Boutet, and Markus W Covert. Heterogeneity in NF-κb activity leads to distinct transcriptional phenotypes in single cells. *Cell Systems*, 4(4):458–469.e5, 2017.

[17] Barbara Di Ventura, Caroline Lemerle, Konstantinos Michalodimitrakis, and Luis Serrano. From in vivo to in silico biology and back. *Nature*, 443(7111):527–533, 2006.

[18] Markus W Covert, Eric M Knight, Jennifer L Reed, Markus J Herrgard, and Bernhard O Palsson. Integrating high-throughput and computational data elucidates bacterial networks. *Nature*, 429(6987):92–96, 2004.

[19] 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* metabolism–2011. *Molecular systems biology*, 7(1):535, 2011.

[20] Ines Thiele, Neema Jamshidi, Ronan MT Fleming, and Bernhard Ø Palsson. Genome-scale reconstruction of *Escherichia coli*'s transcriptional and translational machinery: a knowledge base, its mathematical formulation, and its functional characterization. *PLoS Comput Biol*, 5(3):e1000312, 2009.

[21] Marc Güell, Vera van Noort, Eva Yus, Wei-Hua Chen, Justine Leigh-Bell, Konstantinos Michalodimitrakis, Takuji Yamada, Manimozhiyan Arumugam, Tobias Doerks, Sebastian Kühner, et al. Transcriptome complexity in a genome-reduced bacterium. *Science*, 326(5957):1268–1271, 2009.

[22] Sebastian Kühner, Vera van Noort, Matthew J Betts, Alejandra Leo-Macias, Claire Batisse, Michaela Rode, Takuji Yamada, Tobias Maier, Samuel Bader, Pedro Beltran-Alvarez, et al. Proteome organization in a genome-reduced bacterium. *Science*, 326(5957):1235–1240, 2009.

[23] Eva Yus, Tobias Maier, Konstantinos Michalodimitrakis, Vera van Noort, Takuji Yamada, Wei-Hua Chen, Judith AH Wodke, Marc Güell, Sira Martínez, Ronan Bourgeois, et al. Impact of genome reduction on bacterial metabolism and its regulation. *science*, 326(5957):1263–1268, 2009.

[24] JC Atlas, EV Nikolaev, ST Browning, and ML Shuler. Incorporating genome-wide dna sequence information into a dynamic whole-cell model of escherichia coli: application to dna replication. *IET systems biology*, 2(5):369–382, 2008.

[25] Samuel T Browning, Mariajosé Castellanos, and Michael L Shuler. Robust control of initiation of prokaryotic chromosome replication: essential considerations for a minimal cell. *Biotechnology and bioengineering*, 88(5):575–584, 2004.

[26] Mariajose Castellanos, David B Wilson, and Michael L Shuler. A modular minimal cell model: purine and pyrimidine transport and metabolism. *Proceedings of the National Academy of Sciences of the United States of America*, 101(17):6681–6686, 2004.

[27] Mariajose Castellanos, Keiichiro Kushiro, Samuel K Lai, and Michael L Shuler. A genomically/chemically complete module for synthesis of lipid membrane in a minimal cell. *Biotechnology and bioengineering*, 97(2):397–409, 2007.

[28] MM Domach, SK Leung, RE Cahn, GG Cocks, and ML Shuler. Computer model for glucose-limited growth of a single cell of *Escherichia coli* b/r-a. *Biotechnology and bioengineering*, 26(3):203–216, 1984.

[29] Masaru Tomita, Kenta Hashimoto, Koichi Takahashi, Thomas Simon Shimizu, Yuri Matsuzaki, Fumihiko Miyoshi, Kanako Saito, Sakura Tanida, Katsuyuki Yugi, J Craig Venter, et al. E-cell: software environment for whole-cell simulation. *Bioinformatics*, 15(1):72–84, 1999.

[30] Eric H Davidson, Jonathan P Rast, Paola Oliveri, Andrew Ransick, Cristina Calestani, Chiou-Hwa Yuh, Takuya Minokawa, Gabriele Amore, Veronica Hinman, Cesar Arenas-Mena, et al. A genomic regulatory network for development. *science*, 295(5560):1669–1678, 2002.

[31] Jeffrey D Orth, Ines Thiele, and Bernhard Ø Palsson. What is flux balance analysis? *Nature biotechnology*, 28(3):245–248, 2010.

[32] Claire M Fraser, Jeannine D Gocayne, Owen White, Mark D Adams, et al. The minimal gene complement of mycoplasma genitalium. *Science*, 270(5235):197, 1995.

[33] Patrick F Suthers, Madhukar S Dasika, Vinay Satish Kumar, Gennady Denisov, John I Glass, and Costas D Maranas. A genome-scale metabolic reconstruction of mycoplasma genitalium, i ps189. *PLoS Comput Biol*, 5(2):e1000285, 2009.

[34] Markus W Covert, Christophe H Schilling, and Bernhard Palsson. Regulation of gene expression in flux balance models of metabolism. *Journal of theoretical biology*, 213(1):73–88, 2001.

[35] Markus W Covert, Nan Xiao, Tiffany J Chen, and Jonathan R Karr. Integrating metabolic, transcriptional regulatory and signal transduction models in escherichia coli. *Bioinformatics*, 24(18):2044–2050, 2008.

[36] Sriram Chandrasekaran and Nathan D Price. Probabilistic integrative modeling of genome-scale metabolic and regulatory networks in *Escherichia coli* and mycobacterium tuberculosis. *Proceedings of the National Academy of Sciences*, 107(41):17845–17850, 2010.

[37] Harold J Morowitz, Mark E Tourtellotte, Walter R Guild, Elisea Castro, Carl Woese, and Robert C Cleverdon. The chemical composition and submicroscopic morphology of mycoplasma gallisepticum, avian pplo 5969. *Journal of molecular biology*, 4(2):93IN4–103IN5, 1962.

[38] Harold J Morowitz. *Beginnings of cellular life: metabolism recapitulates biogenesis*. Yale University Press, 1992.

[39] Shan Sundararaj, Anchi Guo, Bahram Habibi-Nazhad, Melania Rouani, Paul Stothard, Michael Ellison, and David S Wishart. The cybercell database (ccdb): a comprehensive, self-updating, relational database to coordinate and facilitate in silico modeling of escherichia coli. *Nucleic acids research*, 32(suppl 1):D293–D295, 2004.

[40] Bryson D Bennett, Elizabeth H Kimball, Melissa Gao, Robin Osterhout, Stephen J Van Dien, and Joshua D Rabinowitz. Absolute metabolite concentrations and implied enzyme active site occupancy in escherichia coli. *Nature chemical biology*, 5(8):593–599, 2009.

[41] Yuichi Taniguchi, Paul J Choi, Gene-Wei Li, Huiyi Chen, Mohan Babu, Jeremy Hearn, Andrew Emili, and X Sunney Xie. Quantifying *E. coli* proteome and transcriptome with single-molecule sensitivity in single cells. *Science*, 329(5991):533–538, 2010.

[42] Ji Yu, Jie Xiao, Xiaojia Ren, Kaiqin Lao, and X Sunney Xie. Probing gene expression in live cells, one protein molecule at a time. *Science*, 311(5767):1600–1603, 2006.

[43] Lok-hang So, Anandamohan Ghosh, Chenghang Zong, Leonardo A Sepúlveda, Ronen Segev, and Ido Golding. General properties of transcriptional time series in escherichia coli. *Nature genetics*, 43(6):554–560, 2011.

[44] Tiffany Vora, Alison K Hottes, and Saeed Tavazoie. Protein occupancy landscape of a bacterial genome. *Molecular cell*, 35(2):247–253, 2009.

[45] Benjamin P Bratton, Rachel A Mooney, and James C Weisshaar. Spatial distribution and diffusive motion of rna polymerase in live escherichia coli. *Journal of bacteriology*, 193(19):5138–5146, 2011.

[46] Yoshie Harada, Takashi Funatsu, Katsuhiko Murakami, Yoshikazu Nonoyama, Akira Ishihama, and Toshio Yanagida. Single-molecule imaging of rna polymerase-dna interactions in real time. *Biophysical journal*, 76(2):709–715, 1999.

[47] Richard T Pomerantz and Mike ODonnell. Direct restart of a replication fork stalled by a head-on rna polymerase. *Science*, 327(5965):590–592, 2010.

[48] James B Russell and Gregory M Cook. Energetics of bacterial growth: balance of anabolic and catabolic reactions. *Microbiological reviews*, 59(1):48–62, 1995.

[49] John I Glass, Nacyra Assad-Garcia, Nina Alperovich, Shibu Yooseph, Matthew R Lewis, Mahir Maruf, Clyde A Hutchison, Hamilton O Smith, and J Craig Venter. Essential genes of a minimal bacterium. *Proceedings of the National Academy of Sciences of the United States of America*, 103(2):425–430, 2006.

[50] Aart de Kok, Annechien F Hengeveld, Alejandro Martin, and Adrie H Westphal. The pyruvate dehydrogenase multi-enzyme complex from gram-negative bacteria. *Biochimica et Biophysica Acta (BBA)-Protein Structure and Molecular Enzymology*, 1385(2):353–366, 1998.

[51] J Dennis Pollack, Melissa A Myers, Thomas Dandekar, and Richard Herrmann. Suspected utility of enzymes with multiple activities in the small genome mycoplasma species: the replacement of the missing" household" nucleoside diphosphate kinase gene and activity by glycolytic kinases. *Omics: a journal of integrative biology*, 6(3):247–258, 2002.

[52] Stuart J Cordwell, David J Basseal, J Dennis Pollack, and Ian Humphery-Smith. Malate/lactate dehydrogenase in mollicutes: evidence for a multienzyme protein. *Gene*, 195(2):113–120, 1997.

[53] Hanns-Ludwig SCHMIDT, Walter STÖCKLEIN, Josef DANZER, Peter KIRCH, and Berthold LIMBACH. Isolation and properties of an h2o-forming nadh oxidase from streptococcus faecalis. *European Journal of Biochemistry*, 156(1):149–155, 1986.

[54] Hiroaki Kitano. Systems biology: a brief overview. *Science*, 295(5560):1662–1664, 2002.

[55] Sydney Brenner. Sequences and consequences. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 365(1537):207–212, 2010.

[56] Daniel G Gibson, Gwynedd A Benders, Cynthia Andrews-Pfannkoch, Evgeniya A Denisova, Holly Baden-Tillson, Jayshree Zaveri, Timothy B Stockwell, Anushka Brownley, David W Thomas, Mikkel A Algire, et al. Complete chemical synthesis, assembly, and cloning of a mycoplasma genitalium genome. *science*, 319(5867):1215–1220, 2008.

[57] Daniel G Gibson, John I Glass, Carole Lartigue, Vladimir N Noskov, Ray-Yuan Chuang, Mikkel A Algire, Gwynedd A Benders, Michael G Montague, Li Ma, Monzia M Moodie, et al. Creation of a bacterial cell controlled by a chemically synthesized genome. *science*, 329(5987):52–56, 2010.

[58] Carole Lartigue, John I Glass, Nina Alperovich, Rembert Pieper, Prashanth P Parmar, Clyde A Hutchison, Hamilton O Smith, and J Craig Venter. Genome transplantation in bacteria: changing one species to another. *science*, 317(5838):632–638, 2007.

[59] Carole Lartigue, Sanjay Vashee, Mikkel A Algire, Ray-Yuan Chuang, Gwynedd A Benders, Li Ma, Vladimir N Noskov, Evgeniya A Denisova, Daniel G Gibson, Nacyra Assad-Garcia, et al. Creating bacterial strains from genomes that have been cloned and engineered in yeast. *science*, 325(5948):1693–1696, 2009.

[60] M L Shuler, S Leung, and C C Dick. A Mathematical Model for the Growth of a Single Cell. *Annals of the New York Academy of Sciences*, 326(1):35–52, 1979.

[61] Michael L Shuler, Patricia Foley, and Jordan Atlas. Modeling a minimal cell. *Methods in molecular biology (Clifton, N.J.)*, 881:573–610, 2012.

[62] J.M. Savinell and B O Palsson. Network analysis of intermediary metabolism using linear optimization. I. Development of mathematical formalism. *Journal of Theoretical Biology*, 154(4):421–454, 1992.

[63] A Varma and B 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, 1994.

[64] Tobias Maier, Alexander Schmidt, Marc Güell, Sebastian Kühner, Anne-Claude Gavin, Ruedi Aebersold, and Luis Serrano. Quantification of mRNA and protein and integration with protein turnover in a bacterium. *Molecular Systems Biology*, 7(1), 2011.

[65] Nobuyoshi Ishii, Kenji Nakahigashi, Tomoya Baba, Martin Robert, Tomoyoshi Soga, Akio Kanai, Takashi Hirasawa, Miki Naba, Kenta Hirai, Aminul Hoque, Pei Yee Ho, Yuji Kakazu, Kaori Sugawara, Saori Igarashi, Satoshi Harada, Takeshi Masuda, Naoyuki Sugiyama, Takashi Togashi, Miki Hasegawa, Yuki Takai, Katsuyuki Yugi, Kazuharu Arakawa, Nayuta Iwata, Yoshihiro Toya, Yoichi Nakayama, Takaaki Nishioka, Kazuyuki Shimizu, Hirotada Mori, and Masaru Tomita. Multiple high-throughput analyses monitor the response of *E. coli* to perturbations. *Science (New York, N.Y.)*, 316(5824):593–597, April 2007.

[66] Paola Picotti, Mathieu Clément-Ziza, Henry Lam, David S Campbell, Alexander Schmidt, Eric W Deutsch, Hannes Röst, Zhi Sun, Oliver Rinner, Lukas Reiter, Qin Shen, Jacob J Michaelson, Andreas Frei, Simon Alberti, Ulrike Kusebauch, Bernd Wollscheid, Robert L Moritz, Andreas Beyer, and Ruedi Aebersold. A complete mass-spectrometric map of the yeast proteome applied to quantitative trait analysis. *Nature*, 494(7436):266–270, January 2013.

[67] Christian Miller, Björn Schwalb, Kerstin Maier, Daniel Schulz, Sebastian Dümcke, Benedikt Zacher, Andreas Mayer, Jasmin Sydow, Lisa Marcinowski, Lars Dölken, Dietmar E Martin, Achim Tresch, and Patrick Cramer. Dynamic transcriptome analysis measures rates of mRNA synthesis and decay in yeast. *Molecular Systems Biology*, 7(1):–, January 2011.

[68] Jörg D Hoheisel. Microarray technology: beyond transcript profiling and genotype analysis. *Nature reviews. Genetics*, 7(3):200–210, March 2006.

[69] Zhong Wang, Mark Gerstein, and Michael Snyder. RNA-Seq: a revolutionary tool for transcriptomics. *Nature reviews. Genetics*, 10(1):57–63, January 2009.

[70] Tahrin Mahmood and Ping-Chang Yang. Western blot: technique, theory, and trouble shooting. *North American journal of medical sciences*, 4(9):429–434, September 2012.

[71] Sean R Gallagher. One-dimensional SDS gel electrophoresis of proteins. *Current protocols in molecular biology / edited by Frederick M. Ausubel ... [et al.]*, Chapter 10:Unit 10.2A, August 2006.

[72] Terrence S Furey. ChIP-seq and beyond: new and improved methodologies to detect and characterize protein-DNA interactions. *Nature reviews. Genetics*, 13(12):840–852, December 2012.

[73] Elijah Roberts, Andrew Magis, Julio O Ortiz, Wolfgang Baumeister, and Zaida Luthey-Schulten. Noise contributions in an inducible genetic switch: a whole-cell simulation study. *PLoS Computational Biology*, 7(3):e1002010, March 2011.

[74] Timothy K Lee, Elissa M Denny, Jayodita C Sanghvi, Jahlionais E Gaston, Nathaniel D Maynard, Jacob J Hughey, and Markus W Covert. A noisy paracrine signal determines the cellular NF-kappaB response to lipopolysaccharide. *Science Signaling*, 2(93):ra65, 2009.

[75] Fuchou Tang, Catalin Barbacioru, Yangzhou Wang, Ellen Nordman, Clarence Lee, Nanlan Xu, Xiaohui Wang, John Bodeau, Brian B Tuch, Asim Siddiqui, Kaiqin Lao, and M Azim Surani. mRNA-Seq whole-transcriptome analysis of a single cell. *Nature methods*, 6(5):377–382, May 2009.

[76] Alfredo J Ibáñez, Stephan R Fagerer, Anna Mareike Schmidt, Pawel L Urban, Konstantins Jefimovs, Philipp Geiger, Reinhard Dechant, Matthias Heinemann, and Renato Zenobi. Mass spectrometry-based metabolomics of single yeast cells. *Proceedings of the National Academy of Sciences*, 110(22):8790–8794, May 2013.

[77] Ron Caspi, Tomer Altman, Joseph M Dale, Kate Dreher, Carol A Fulcher, Fred Gilham, Pallavi Kaipa, Athikkattuvalasu S Karthikeyan, Anamika Kothari, Markus Krummenacker, Mario Latendresse, Lukas A Mueller, Suzanne Paley, Liviu Popescu, Anuradha Pujar, Alexander G Shearer, Peifen Zhang, and Peter D Karp. The MetaCyc database of metabolic pathways and enzymes and the BioCyc collection of pathway/genome databases. *Nucleic acids research*, 38(Database issue):D473–9, January 2010.

[78] Ida Schomburg, Antje Chang, Sandra Placzek, Carola Söhngen, Michael Rother, Maren Lang, Cornelia Munaretto, Susanne Ulas, Michael Stelzer, Andreas Grote, Maurice Scheer, and Dietmar Schomburg. BRENDA in 2013: integrated reactions, kinetic data, enzyme function data, improved disease classification: new options and contents in BRENDA. *Nucleic acids research*, 41(Database issue):D764–72, January 2013.

[79] U Wittig, R Kania, M Golebiewski, M Rey, L Shi, L Jong, E Algaa, A Weidemann, H Sauer-Danzwith, S Mir, O Krebs, M Bittkowski, E Wetsch, I Rojas, and W Muller. SABIO-RK–database for biochemical reaction kinetics. *Nucleic acids research*, 40(D1):D790–D796, December 2011.

[80] T Barrett, S E Wilhite, P Ledoux, C Evangelista, I F Kim, M Tomashevsky, K A Marshall, K H Phillippy, P M Sherman, M Holko, A Yefanov, H Lee, N Zhang, C L Robertson, N Serova,

S Davis, and A Soboleva. NCBI GEO: archive for functional genomics data sets–update. *Nucleic acids research*, 41(D1):D991–D995, December 2012.

[81] Jonathan R. Karr, Jayodita C Sanghvi, Derek N Macklin, Abhishek Arora, and Markus W Covert. WholeCellKB: model organism databases for comprehensive whole-cell models. *Nucleic acids research*, 41(D1):D787–D792, 2013.

[82] Jenny Finkel, Shipra Dingare, Christopher D Manning, Malvina Nissim, Beatrice Alex, and Claire Grover. Exploring the boundaries: gene and protein identification in biomedical text. *BMC bioinformatics*, 6 Suppl 1:S5, 2005.

[83] Ines Thiele and Bernhard O Palsson. Reconstruction annotation jamborees: a community approach to systems biology. *Molecular Systems Biology*, 6:361, April 2010.

[84] Announcement: Reducing our irreproducibility. *Nature*, 496(7446):398–398, April 2013.

[85] Leland H Hartwell, John J Hopfield, Stanislas Leibler, and Andrew W Murray. From molecular to modular cell biology. *Nature*, 402:C47–C52, December 1999.

[86] Richard Bonneau, David J Reiss, Paul Shannon, Marc Facciotti, Leroy Hood, Nitin S Baliga, and Vesteinn Thorsson. The Inferelator: an algorithm for learning parsimonious regulatory networks from systems-biology data sets de novo. *Genome Biology*, 7(5):R36, 2006.

[87] Javier Carrera, Guillermo Rodrigo, and Alfonso Jaramillo. Model-based redesign of global transcription regulation. *Nucleic acids research*, 37(5):e38, April 2009.

[88] CUDA Toolkit Documentation. `http://docs.nvidia.com/cuda/index.html`.

[89] MPI Documents. `http://www.mpi-forum.org/docs/`.

[90] Jeremy Gunawardena. Silicon dreams of cells into symbols. *Nature Biotechnology*, 30(9):838–840, September 2012.

[91] Ron O Dror, Robert M Dirks, J P Grossman, Huafeng Xu, and David E Shaw. Biomolecular simulation: a computational microscope for molecular biology. *Annual Review of Biophysics*, 41:429–452, 2012.

[92] Rae Silver, Kwabena Boahen, Sten Grillner, Nancy Kopell, and Kathie L Olsen. Neurotech for neuroscience: unifying concepts, organizing principles, and emerging tools. *The Journal of neuroscience : the official journal of the Society for Neuroscience*, 27(44):11807–11819, October 2007.

[93] Business Intelligence and Analytics Software. `http://www.tableausoftware.com/`.

[94] Tableau Technology — Tableau Software. `http://www.tableausoftware.com/products/technology`.

[95] Pak Chung Wong, Han-Wei Shen, Christopher R Johnson, Chaomei Chen, and Robert B Ross. The Top 10 Challenges in Extreme-Scale Visual Analytics. *IEEE Computer Graphics and Applications*, 32(4):63–67, 2012.

[96] Miriah Meyer, Tamara Munzner, Angela DePace, and Hanspeter Pfister. MulteeSum: a tool for comparative spatial and temporal gene expression data. *IEEE transactions on visualization and computer graphics*, 16(6):908–917, November 2010.

[97] M Meyer, B Wong, M Styczynski, T Munzner, and H Pfister. Pathline: A Tool For Comparative Functional Genomics. *Computer Graphics Forum*, 29(3):1043–1052, August 2010.

[98] Miriah Meyer, Tamara Munzner, and Hanspeter Pfister. MizBee: a multiscale synteny browser. *IEEE transactions on visualization and computer graphics*, 15(6):897–904, November 2009.

[99] Steve Ashby, Pete Beckman, Jackie Chen, Phil Colella, Bill Collins, Dona Crawford, Jack Dongarra, Doug Kothe, Rusty Lusk, and Paul Messina. The opportunities and challenges of exascale computing—summary report of the advanced scientific computing advisory committee (ASCAC) subcommittee. US Department of Energy Office of Science. *US Department of Energy Office of Science*, 2010.

[100] The MIT License (MIT) — Open Source Initiative. `http://opensource.org/licenses/MIT`.

[101] Whole-cell parameter estimation DREAM challenge - syn1876068. `https://www.synapse.org/#!Synapse:syn1876068`.

[102] Oliver Purcell, Bonny Jain, Jonathan R. Karr, Markus W Covert, and Timothy K Lu. Towards a whole-cell modeling approach for synthetic biology. *Chaos (Woodbury, N.Y.)*, 23(2):025112, June 2013.

[103] Cell line: 1974-2014. `http://www.cell.com/pb/assets/raw/journals/research/cell/cell-timeline-40/Timeline_4_PF.pdf`. Accessed: 2017-02-12.

[104] Frederick Carl Neidhardt, John L Ingraham, and Moselio Schaechter. *Physiology of the bacterial cell: a molecular approach*. 1990.

[105] Ez-rdm composition. `http://www.teknova.com/EZ-RICH-DEFINED-MEDIUM-KIT-p/m2105.htm`. Accessed: 2017-01-31.

[106] Bbtools. `http://jgi.doe.gov/data-and-tools/bbtools/`. Accessed: 2017-02-16.

[107] Bo Li and Colin N Dewey. Rsem: accurate transcript quantification from rna-seq data with or without a reference genome. *BMC bioinformatics*, 12(1):323, 2011.

[108] Tanja Magoc, Derrick Wood, and Steven L Salzberg. Edge-pro: estimated degree of gene expression in prokaryotic genomes. *Evolutionary Bioinformatics*, 9:127, 2013.

[109] Cole Trapnell, Adam Roberts, Loyal Goff, Geo Pertea, Daehwan Kim, David R Kelley, Harold Pimentel, Steven L Salzberg, John L Rinn, and Lior Pachter. Differential gene and transcript expression analysis of rna-seq experiments with tophat and cufflinks. *Nature protocols*, 7(3):562–578, 2012.

[110] Estimating number of transcripts from rna-seq measurements. `https://liorpachter.wordpress.com/2014/04/30/estimating-number-of-transcripts-from-rna-seq-measurements-and-why-i-believe-in-paywall/`. Accessed: 2017-02-16.

[111] Javier Carrera, Raissa Estrela, Jing Luo, Navneet Rai, Athanasios Tsoukalas, and Ilias Tagkopoulos. An integrative, multi-scale, genome-wide model reveals the phenotypic landscape of escherichia coli. *Molecular systems biology*, 10(7):735, 2014.

[112] Socorro Gama-Castro, Heladia Salgado, Alberto Santos-Zavaleta, Daniela Ledezma-Tejeida, Luis Muñiz-Rascado, Jair Santiago García-Sotelo, Kevin Alquicira-Hernández, Irma Martínez-Flores, Lucia Pannier, Jaime Abraham Castro-Mondragón, et al. Regulondb version 9.0: high-level integration of gene regulation, coexpression, motif clustering and beyond. *Nucleic acids research*, 44(D1):D133–D143, 2016.

[113] D Bray and Robert B Bourret. Computer analysis of the binding reactions leading to a transmembrane receptor-linked multiprotein complex involved in bacterial chemotaxis. *Molecular biology of the cell*, 6(10):1367–1380, 1995.

[114] Rakesh S Laishram and Jayaraman Gowrishankar. Environmental regulation operating at the promoter clearance step of bacterial transcription. *Genes & development*, 21(10):1258–1272, 2007.

[115] Karsten Rippe, Norbert MuÈcke, and Alexandra Schulz. Association states of the transcription activator protein ntrc from *E. coli* determined by analytical ultracentrifugation. *Journal of molecular biology*, 278(5):915–933, 1998.

[116] Ecocyc dna segments. `https://ecocyc.org/ECOLI/NEW-IMAGE?object=DNA-Segments`. Accessed: 2017-01-10.

[117] Ecocyc mrna segments. `https://ecocyc.org/ECOLI/NEW-IMAGE?object=mRNA-Segments`. Accessed: 2017-01-10.

[118] Jeremiah J Faith, Boris Hayete, Joshua T Thaden, Ilaria Mogno, Jamey Wierzbowski, Guillaume Cottarel, Simon Kasif, James J Collins, and Timothy S Gardner. Large-scale mapping

and validation of *Escherichia coli* transcriptional regulation from a compendium of expression profiles. *PLoS biol*, 5(1):e8, 2007.

[119] Wenyan Jiang, David Bikard, David Cox, Feng Zhang, and Luciano A Marraffini. Rna-guided editing of bacterial genomes using crispr-cas systems. *Nature biotechnology*, 31(3):233–239, 2013.

[120] Polly M Fordyce, Doron Gerber, Danh Tran, Jiashun Zheng, Hao Li, Joseph L DeRisi, and Stephen R Quake. De novo identification and biophysical characterization of transcription-factor binding sites with microfluidic affinity analysis. *Nature biotechnology*, 28(9):970–975, 2010.

[121] Shasha Chong, Chongyi Chen, Hao Ge, and X Sunney Xie. Mechanism of transcriptional bursting in bacteria. *Cell*, 158(2):314–326, 2014.

[122] Andreas Kremling, Katja Bettenbrock, and Ernst Dieter Gilles. Analysis of global control of *Escherichia coli* carbohydrate uptake. *BMC systems biology*, 1(1):42, 2007.

[123] 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.

[124] Piyush Labhsetwar, John Andrew Cole, Elijah Roberts, Nathan D Price, and Zaida A Luthey-Schulten. Heterogeneity in protein expression induces metabolic variability in a modeled *Escherichia coli* population. *Proceedings of the National Academy of Sciences*, 110(34):14006–14011, 2013.

[125] Elsa W Birch, Nicholas A Ruggero, and Markus W Covert. Determining host metabolic limitations on viral replication via integrated modeling and experimental perturbation. *PLoS Comput Biol*, 8(10):e1002746, 2012.

[126] Elsa W Birch, Madeleine Udell, and Markus W Covert. Incorporation of flexible objectives and time-linked simulation with flux balance analysis. *Journal of theoretical biology*, 345:12–21, 2014.

[127] Amit Varma, Brian W Boesch, and Bernhard O Palsson. Stoichiometric interpretation of *Escherichia coli* glucose catabolism under various oxygenation rates. *Applied and environmental microbiology*, 59(8):2465–2473, 1993.

[128] Arren Bar-Even, Elad Noor, Yonatan Savir, Wolfram Liebermeister, Dan Davidi, Dan S Tawfik, and Ron Milo. The moderately efficient enzyme: evolutionary and physicochemical trends shaping enzyme parameters. *Biochemistry*, 50(21):4402–4410, 2011.

[129] FHC Crick. Project k:" the complete solution of e. coli". *Perspectives in Biology and Medicine*, 17(1):67–70, 1973.

[130] Open Science Collaboration et al. Psychology. estimating the reproducibility of psychological science. science 349: aac4716, 2015.

[131] Monya Baker. 1,500 scientists lift the lid on reproducibility. *Nature*, 533(7604):452–454, 2016.

[132] Sir Arthur Eddington. *New Pathways in Science*, page 211. Cambridge University Press, Cambridge UK, 1935.

[133] Horace Freeland Judson. *The Eighth Day of Creation: Makers of the Revolution in Biology*, page 93. Cold Spring Harbor Laboratory Press, Cold Spring Harbor NY USA, twenty-fifth anniversary edition edition, 1996.

[134] ML Shuler and MM Domach. Mathematical models of the growth of individual cells. ACS Publications, 1983.

[135] Michael L Shuler, Patricia Foley, and Jordan Atlas. Modeling a minimal cell. *Microbial Systems Biology: Methods and Protocols*, pages 573–610, 2012.

[136] Masaru Tomita. Whole-cell simulation: a grand challenge of the 21st century. *Trends in biotechnology*, 19(6):205–210, 2001.

[137] Jennifer L Reed, Iman Famili, Ines Thiele, and Bernhard O Palsson. Towards multidimensional genome annotation. *Nature Reviews Genetics*, 7(2):130–141, 2006.

[138] John A Cole and Zaida Luthey-Schulten. Whole cell modeling: from single cells to colonies. *Israel journal of chemistry*, 54(8-9):1219–1229, 2014.

[139] Ingrid M Keseler, Amanda Mackie, Alberto Santos-Zavaleta, Richard Billington, César Bonavides-Martínez, Ron Caspi, Carol Fulcher, Socorro Gama-Castro, Anamika Kothari, Markus Krummenacker, et al. The ecocyc database: reflecting new knowledge about *Escherichia coli* k-12. *Nucleic Acids Research*, page gkw1003, 2016.

[140] Ji Yu, Jie Xiao, Xiaojia Ren, Kaiqin Lao, and X Sunney Xie. Probing gene expression in live cells, one protein molecule at a time. *Science*, 311(5767):1600–1603, 2006.

[141] Javier Carrera, Raissa Estrela, Jing Luo, Navneet Rai, Athanasios Tsoukalas, and Ilias Tagkopoulos. An integrative, multi-scale, genome-wide model reveals the phenotypic landscape of escherichia coli. *Molecular systems biology*, 10(7):735, 2014.

[142] Socorro Gama-Castro, Heladia Salgado, Alberto Santos-Zavaleta, Daniela Ledezma-Tejeida, Luis Muñiz-Rascado, Jair Santiago García-Sotelo, Kevin Alquicira-Hernández, Irma Martínez-Flores, Lucia Pannier, Jaime Abraham Castro-Mondragón, et al. Regulondb version 9.0: high-level integration of gene regulation, coexpression, motif clustering and beyond. *Nucleic acids research*, 44(D1):D133–D143, 2016.

[143] Alexander Schmidt, Karl Kochanowski, Silke Vedelaar, Erik Ahrné, Benjamin Volkmer, Luciano Callipo, Kèvin Knoops, Manuel Bauer, Ruedi Aebersold, and Matthias Heinemann. The quantitative and condition-dependent *Escherichia coli* proteome. *Nature biotechnology*, 34(1):104–110, 2016.

[144] A Bachmair, D Finley, and A Varshavsky. In vivo half-life of a protein is a function of its amino-terminal residue. *Science*, 234(4773):179–186, 1986.

[145] 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* metabolism–2011. *Molecular systems biology*, 7(1):535, 2011.

[146] Daniel S Weaver, Ingrid M Keseler, Amanda Mackie, Ian T Paulsen, and Peter D Karp. A genome-scale metabolic flux model of *Escherichia coli* k–12 derived from the ecocyc database. *BMC systems biology*, 8(1):79, 2014.

[147] Bryson D Bennett, Elizabeth H Kimball, Melissa Gao, Robin Osterhout, Stephen J Van Dien, and Joshua D Rabinowitz. Absolute metabolite concentrations and implied enzyme active site occupancy in escherichia coli. *Nature chemical biology*, 5(8):593–599, 2009.

[148] Jeffrey D Orth, Ines Thiele, and Bernhard Ø Palsson. What is flux balance analysis? *Nature biotechnology*, 28(3):245–248, 2010.

[149] Markus W Covert, Christophe H Schilling, and Bernhard Palsson. Regulation of gene expression in flux balance models of metabolism. *Journal of theoretical biology*, 213(1):73–88, 2001.

[150] Piyush Labhsetwar, John Andrew Cole, Elijah Roberts, Nathan D Price, and Zaida A Luthey-Schulten. Heterogeneity in protein expression induces metabolic variability in a modeled *Escherichia coli* population. *Proceedings of the National Academy of Sciences*, 110(34):14006–14011, 2013.

[151] Elsa W Birch, Madeleine Udell, and Markus W Covert. Incorporation of flexible objectives and time-linked simulation with flux balance analysis. *Journal of theoretical biology*, 345:12–21, 2014.

[152] Yoshihiro Toya, Nobuyoshi Ishii, Kenji Nakahigashi, Takashi Hirasawa, Tomoyoshi Soga, Masaru Tomita, and Kazuyuki Shimizu. 13c-metabolic flux analysis for batch culture of *Escherichia coli* and its pyk and pgi gene knockout mutants based on mass isotopomer distribution of intracellular metabolites. *Biotechnology progress*, 26(4):975–992, 2010.

[153] Måns Ehrenberg, Hans Bremer, and Patrick P Dennis. Medium-dependent control of the bacterial growth rate. *Biochimie*, 95(4):643–658, 2013.

[154] Hans Bremer and Patrick P Dennis. Modulation of chemical composition and other parameters of the cell at different exponential growth rates. *EcoSal Plus*, 3(1), 2008.

[155] William D Donachie. Relationship between cell size and time of initiation of dna replication. *Nature*, 219(5158):1077–1079, 1968.

[156] Stephen Cooper and Charles E Helmstetter. Chromosome replication and the division cycle of escherichia colibr. *Journal of molecular biology*, 31(3):519–540, 1968.

[157] Mats Wallden, David Fange, Ebba Gregorsson Lundius, Özden Baltekin, and Johan Elf. The synchronization of replication and division cycles in individual *E. coli* cells. *Cell*, 166(3):729–739, 2016.

[158] John T Sauls, Dongyang Li, and Suckjoon Jun. Adder and a coarse-grained approach to cell size homeostasis in bacteria. *Current opinion in cell biology*, 38:38–44, 2016.

[159] Yu Tanouchi, Anand Pai, Heungwon Park, Shuqiang Huang, Rumen Stamatov, Nicolas E Buchler, and Lingchong You. A noisy linear map underlies oscillations in cell size and gene expression in bacteria. *Nature*, 523(7560):357–360, 2015.

[160] Nathalie Q Balaban, Jack Merrin, Remy Chait, Lukasz Kowalik, and Stanislas Leibler. Bacterial persistence as a phenotypic switch. *Science*, 305(5690):1622–1625, 2004.

[161] Kyle R Allison, Mark P Brynildsen, and James J Collins. Heterogeneous bacterial persisters and engineering approaches to eliminate them. *Current opinion in microbiology*, 14(5):593–598, 2011.

[162] Ingrid M Keseler, Amanda Mackie, Martin Peralta-Gil, Alberto Santos-Zavaleta, Socorro Gama-Castro, César Bonavides-Martínez, Carol Fulcher, Araceli M Huerta, Anamika Kothari, Markus Krummenacker, et al. Ecocyc: fusing model organism databases with systems biology. *Nucleic acids research*, 41(D1):D605–D612, 2013.

[163] Ecocyc transcription unit definition. `https://ecocyc.org/ECOLI/NEW-IMAGE?object=Transcription-Units`. Accessed: 2017-01-10.

[164] Matthew Boitano. Personal Communication.

[165] Hans Bremer and Patrick P Dennis. Modulation of chemical composition and other parameters of the cell by growth rate. *Escherichia coli and Salmonella: cellular and molecular biology,*, 2:1553–1569, 1996.

[166] Clyde A Hutchison, Ray-Yuan Chuang, Vladimir N Noskov, Nacyra Assad-Garcia, Thomas J Deerinck, Mark H Ellisman, John Gill, Krishna Kannan, Bogumil J Karas, Li Ma, et al. Design and synthesis of a minimal bacterial genome. *Science*, 351(6280):aad6253, 2016.

[167] Adam Arkin, John Ross, and Harley H McAdams. Stochastic kinetic analysis of developmental pathway bifurcation in phage λ-infected *Escherichia coli* cells. *Genetics*, 149(4):1633–1648, 1998.

[168] Kerwyn Casey Huang, Yigal Meir, and Ned S Wingreen. Dynamic structures in escherichia coli: spontaneous formation of mine rings and mind polar zones. *Proceedings of the National Academy of Sciences*, 100(22):12724–12728, 2003.

[169] Xiaoxue Wang and Thomas K Wood. Toxin-antitoxin systems influence biofilm and persister cell formation and the general stress response. *Applied and environmental microbiology*, 77(16):5577–5583, 2011.

[170] Ido Golding, Johan Paulsson, Scott M Zawilski, and Edward C Cox. Real-time kinetics of gene activity in individual bacteria. *Cell*, 123(6):1025–1036, 2005.

[171] Lacramioara Bintu, Nicolas E Buchler, Hernan G Garcia, Ulrich Gerland, Terence Hwa, Jané Kondev, and Rob Phillips. Transcriptional regulation by the numbers: models. *Current opinion in genetics & development*, 15(2):116–124, 2005.

[172] Gene-Wei Li, David Burkhardt, Carol Gross, and Jonathan S Weissman. Quantifying absolute protein synthesis rates reveals principles underlying allocation of cellular resources. *Cell*, 157(3):624–635, 2014.

[173] Michael Gale, Seng-Lai Tan, and Michael G Katze. Translational control of viral gene expression in eukaryotes. *Microbiology and molecular biology reviews*, 64(2):239–280, 2000.

[174] SK Jang, HG Kräusslich, MJ Nicklin, GM Duke, AC Palmenberg, and E Wimmer. A segment of the 5'nontranslated region of encephalomyocarditis virus rna directs internal entry of ribosomes during in vitro translation. *Journal of virology*, 62(8):2636–2643, 1988.

[175] STEFAN Schwartz, BK Felber, EM Fenyö, and GN Pavlakis. Env and vpu proteins of human immunodeficiency virus type 1 are produced from multiple bicistronic mrnas. *Journal of virology*, 64(11):5448–5456, 1990.

[176] Waltraud Schmidt-Puchta, Diana Dominguez, Diana Lewetag, and Thomas Hohn. Plant ribosome shunting in vitro. *Nucleic acids research*, 25(14):2854–2860, 1997.

[177] Maja Hemmings-Mieszczak, Gerhard Steger, and Thomas Hohn. Alternative structures of the cauliflower mosaic virus 35 s rna leader: implications for viral expression and replication. *Journal of molecular biology*, 267(5):1075–1088, 1997.

[178] Raymond F Gesteland and John F Atkins. Recoding: dynamic reprogramming of translation. *Annual review of biochemistry*, 65(1):741–768, 1996.

[179] JB Harford. Translation-targeted therapeutics for viral diseases. *Gene expression*, 4(6):357–367, 1994.

[180] CG Kurland. Translational accuracy and the fitness of bacteria*. *Annual review of genetics*, 26(1):29–50, 1992.

[181] Miranda Shehu-Xhilaga, Suzanne M Crowe, and Johnson Mak. Maintenance of the gag/gag-pol ratio is important for human immunodeficiency virus type 1 rna dimerization and viral infectivity. *Journal of virology*, 75(4):1834–1841, 2001.

[182] Dominic Dulude, Yamina A Berchiche, Karine Gendron, Léa Brakier-Gingras, and Nikolaus Heveker. Decreasing the frameshift efficiency translates into an equivalent reduction of the replication of the human immunodeficiency virus type 1. *Virology*, 345(1):127–136, 2006.

[183] Pavel V Baranov, Raymond F Gesteland, and John F Atkins. Recoding: translational bifurcations in gene expression. *Gene*, 286(2):187–201, 2002.

[184] Ewan P Plant, Rasa Rakauskaitė, Deborah R Taylor, and Jonathan D Dinman. Achieving a golden mean: mechanisms by which coronaviruses ensure synthesis of the correct stoichiometric ratios of viral proteins. *Journal of virology*, 84(9):4330–4340, 2010.

[185] Margaret E Levin, Roger W Hendrix, and Sherwood R Casjens. A programmed translational frameshift is required for the synthesis of a bacteriophage λ tail assembly protein. *Journal of molecular biology*, 234(1):124–139, 1993.

[186] Jun Xu. A conserved frameshift strategy in dsdna long tailed bacteriophages. *University of Pittsburgh, Pittsburgh, PA*, 2001.

[187] Jaunius Urbonavičius, Qiang Qian, Jérôme MB Durand, Tord G Hagervall, and Glenn R Björk. Improvement of reading frame maintenance is a common function for several trna modifications. *The EMBO Journal*, 20(17):4863–4873, 2001.

[188] Patrick A Limbach, Pamela F Crain, and James A McCloskey. Summary: the modified nucleosides of rna. *Nucleic acids research*, 22(12):2183–2196, 1994.

[189] Mathias Sprinzl and Konstantin S Vassilenko. Compilation of trna sequences and sequences of trna genes. *Nucleic acids research*, 33(suppl 1):D139–D140, 2005.

[190] Andrei Alexandrov, Irina Chernyakov, Weifeng Gu, Shawna L Hiley, Timothy R Hughes, Elizabeth J Grayhack, and Eric M Phizicky. Rapid trna decay can result from lack of nonessential modifications. *Molecular cell*, 21(1):87–96, 2006.

[191] Eric M Phizicky and Anita K Hopper. trna biology charges to the front. *Genes & development*, 24(17):1832–1860, 2010.

[192] Kristina Nilsson, Hans K Lundgren, Tord G Hagervall, and Glenn R Björk. The cysteine desulfurase iscs is required for synthesis of all five thiolated nucleosides present in trna from salmonella enterica serovar typhimurium. *Journal of bacteriology*, 184(24):6830–6835, 2002.

[193] Gunilla Jäger, Ramune Leipuviene, Michael G Pollard, Qiang Qian, and Glenn R Björk. The conserved cys-x1-x2-cys motif present in the ttca protein is required for the thiolation of cytidine in position 32 of trna from salmonella enterica serovar typhimurium. *Journal of bacteriology*, 186(3):750–757, 2004.

[194] Fabien Pierrel, Thierry Douki, Marc Fontecave, and Mohamed Atta. Miab protein is a bifunctional radical-s-adenosylmethionine enzyme involved in thiolation and methylation of trna. *Journal of Biological Chemistry*, 279(46):47555–47563, 2004.

[195] Yoshiho Ikeuchi, Naoki Shigi, Jun-ichi Kato, Akiko Nishimura, and Tsutomu Suzuki. Mechanistic insights into sulfur relay by multiple sulfur mediators involved in thiouridine biosynthesis at trna wobble positions. *Molecular cell*, 21(1):97–108, 2006.

[196] Béatrice Py and Frédéric Barras. Building fe–s proteins: bacterial strategies. *Nature Reviews Microbiology*, 8(6):436–446, 2010.

[197] Arthur Landy and Wilma Ross. Viral integration and excision: Structure of the lambda att sites: Dna sequences have been determined for regions involved in lambda site-specific recombination. *Science (New York, NY)*, 197(4309):1147, 1977.

[198] Hugo D Urbina, Jonathan J Silberg, Kevin G Hoff, and Larry E Vickery. Transfer of sulfur from iscs to iscu during fe/s cluster assembly. *Journal of Biological Chemistry*, 276(48):44521–44526, 2001.

[199] Francesco Bonomi, Stefania Iametti, Anna Morleo, Dennis Ta, and Larry E Vickery. Studies on the mechanism of catalysis of iron- sulfur cluster transfer from iscu [2fe2s] by hsca/hscb chaperones. *Biochemistry*, 47(48):12795–12801, 2008.

[200] Juanjuan Yang, Jacob P Bitoun, and Huangen Ding. Interplay of isca and iscu in biogenesis of iron-sulfur clusters. *Journal of Biological Chemistry*, 281(38):27956–27963, 2006.

[201] Yasuhiro Takahashi and Minoru Nakamura. Functional assignment of the orf2-iscs-iscu-isca-hscb-hsca-fdx-0rf3 gene cluster involved in the assembly of fe-s clusters in escherichia coli. *Journal of Biochemistry*, 126(5):917–926, 1999.

[202] Christopher J Schwartz, Jennifer L Giel, Thomas Patschkowski, Christopher Luther, Frank J Ruzicka, Helmut Beinert, and Patricia J Kiley. Iscr, an fe-s cluster-containing transcription factor, represses expression of *Escherichia coli* genes encoding fe-s cluster assembly proteins. *Proceedings of the National Academy of Sciences*, 98(26):14895–14900, 2001.

[203] Ravi Kambampati and Charles T Lauhon. Evidence for the transfer of sulfane sulfur from iscs to thii during the in vitro biosynthesis of 4-thiouridine inescherichia coli trna. *Journal of Biological Chemistry*, 275(15):10727–10730, 2000.

[204] Ravi Kambampati and Charles T Lauhon. Mnma and iscs are required for in vitro 2-thiouridine biosynthesis in escherichia coli. *Biochemistry*, 42(4):1109–1117, 2003.

[205] Drik Elseviers, Lynn A Petrullo, and Patricia J Gallagher. Novel *E. coli* mutants deficient in blosynthesis of 5-methylamlnomethyl-2-thiouridine. *Nucleic acids research*, 12(8):3521–3534, 1984.

[206] Takashi Kunisawa, Shigehiko Kanaya, and Elizabeth Kutter. Comparison of synonymous codon distribution patterns of bacteriophage and host genomes. *DNA Research*, 5(6):319–326, 1998.

[207] Julius B Lucks, David R Nelson, Grzegorz R Kudla, and Joshua B Plotkin. Genome landscapes and bacteriophage codon usage. *PLoS Comput Biol*, 4(2):e1000001, 2008.

[208] Rong Shi, Ariane Proteau, Magda Villarroya, Ismaïl Moukadiri, Linhua Zhang, Jean-François Trempe, Allan Matte, M Eugenia Armengod, and Miroslaw Cygler. Structural basis for fe–s cluster assembly and trna thiolation mediated by iscs protein–protein interactions. *PLoS Biol*, 8(4):e1000354, 2010.

[209] Gabor L Igloi. Interaction of trnas and of phosphorothioate-substituted nucleic acids with an organomercurial. probing the chemical environment of thiolated residues by affinity electrophoresis. *Biochemistry*, 27(10):3842–3849, 1988.

[210] Ramune Leipuviene, Qiang Qian, and Glenn R Björk. Formation of thiolated nucleosides present in trna from salmonella enterica serovar typhimurium occurs in two principally distinct pathways. *Journal of bacteriology*, 186(3):758–766, 2004.

[211] Magdeleine Hung, Pratiksha Patel, Susan Davis, and Simon R Green. Importance of ribosomal frameshifting for human immunodeficiency virus type 1 particle assembly and replication. *Journal of virology*, 72(6):4819–4824, 1998.

[212] Kirill A Datsenko and Barry L Wanner. One-step inactivation of chromosomal genes in *Escherichia coli* k-12 using pcr products. *Proceedings of the National Academy of Sciences*, 97(12):6640–6645, 2000.

[213] Masanari Kitagawa, Takeshi Ara, Mohammad Arifuzzaman, Tomoko Ioka-Nakamichi, Eiji Inamoto, Hiromi Toyonaga, and Hirotada Mori. Complete set of orf clones of *Escherichia coli* aska library (a complete set of *E. coli* k-12 orf archive): unique resources for biological research. *DNA research*, 12(5):291–299, 2006.

[214] H. Bremer and G Churchward. An examination of the Cooper-Helmstetter theory of DNA replication in bacteria and its underlying assumptions. *Journal of Theoretical Biology*, 69(4):645–654, December 1977.

[215] Anubhav Jain, Shyue Ping Ong, Wei Chen, Bharat Medasani, Xiaohui Qu, Michael Kocher, Miriam Brafman, Guido Petretto, Gian-Marco Rignanese, Geoffroy Hautier, Daniel Gunter, and Kristin A. Persson. Fireworks: a dynamic workflow system designed for high-throughput applications. *Concurrency and Computation: Practice and Experience*, 27(17):5037–5059, 2015. CPE-14-0307.R2.

[216] Hans Bremer and Patrick P Dennis. Modulation of chemical composition and other parameters of the cell at different exponential growth rates. *EcoSal Plus*, 3(1), 2008.

[217] Jonathan A Bernstein, Arkady B Khodursky, Pei-Hsun Lin, Sue Lin-Chao, and Stanley N Cohen. Global analysis of mrna decay and abundance in *Escherichia coli* at single-gene resolution using two-color fluorescent dna microarrays. *Proceedings of the National Academy of Sciences*, 99(15):9697–9702, 2002.

[218] J B Russell and G M Cook. Energetics of bacterial growth: balance of anabolic and catabolic reactions. *Microbiological reviews*, 59(1):48–62, March 1995.

[219] Gene-Wei Li, David Burkhardt, Carol Gross, and Jonathan S Weissman. Quantifying absolute protein synthesis rates reveals principles underlying allocation of cellular resources. *Cell*, 157(3):624–635, 2014.

[220] Alexander Schmidt, Karl Kochanowski, Silke Vedelaar, Erik Ahrné, Benjamin Volkmer, Luciano Callipo, Kèvin Knoops, Manuel Bauer, Ruedi Aebersold, and Matthias Heinemann. The quantitative and condition-dependent *Escherichia coli* proteome. *Nature biotechnology*, 34(1):104–110, 2016.

[221] J W Tobias, T E Shrader, G Rocap, and A Varshavsky. The N-end rule in bacteria. *Science (New York, N.Y.)*, 254(5036):1374–1377, November 1991.

[222] Ingrid M Keseler, Amanda Mackie, Martin Peralta-Gil, Alberto Santos-Zavaleta, Socorro Gama-Castro, César Bonavides-Martínez, Carol Fulcher, Araceli M Huerta, Anamika Kothari, Markus Krummenacker, Mario Latendresse, Luis Muñiz-Rascado, Quang Ong, Suzanne Paley, Imke Schröder, Alexander G Shearer, Pallavi Subhraveti, Mike Travers, Deepika Weerasinghe, Verena Weiss, Julio Collado-Vides, Robert P Gunsalus, Ian Paulsen, and Peter D Karp. EcoCyc: fusing model organism databases with systems biology. *Nucleic acids research*, 41(Database issue):D605–12, January 2013.

[223] Elsa W Birch, Madeleine Udell, and Markus W Covert. Incorporation of flexible objectives and time-linked simulation with flux balance analysis. *Journal of Theoretical Biology*, 345:12–21, March 2014.

[224] Peter D Karp, Daniel Weaver, Suzanne Paley, Carol Fulcher, Aya Kubo, Anamika Kothari, Markus Krummenacker, Pallavi Subhraveti, et al. The EcoCyc database. *EcoSal Plus*, 2014.

[225] 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.

[226] Bryson D Bennett, Elizabeth H Kimball, Melissa Gao, Robin Osterhout, Stephen J Van Dien, and Joshua D Rabinowitz. Absolute metabolite concentrations and implied enzyme active site occupancy in escherichia coli. *Nature chemical biology*, 5(8):593–599, 2009.

[227] Daniel S Weaver, Ingrid M Keseler, Amanda Mackie, Ian T Paulsen, and Peter D Karp. A genome-scale metabolic flux model of *Escherichia coli* K-12 derived from the EcoCyc database. *BMC Systems Biology*, 8:79, June 2014.

[228] Ida Schomburg, Antje Chang, Sandra Placzek, Carola Söhngen, Michael Rother, Maren Lang, Cornelia Munaretto, Susanne Ulas, Michael Stelzer, Andreas Grote, Maurice Scheer, and Dietmar Schomburg. BRENDA in 2013: integrated reactions, kinetic data, enzyme function data, improved disease classification: new options and contents in BRENDA. *Nucleic acids research*, 41(Database issue):D764–72, January 2013.

[229] Yoshihiro Toya, Nobuyoshi Ishii, Kenji Nakahigashi, Takashi Hirasawa, Tomoyoshi Soga, Masaru Tomita, and Kazuyuki Shimizu. 13c-metabolic flux analysis for batch culture of *Escherichia coli* and its pyk and pgi gene knockout mutants based on mass isotopomer distribution of intracellular metabolites. *Biotechnology progress*, 26(4):975–992, 2010.

[230] Adam M Feist, Christopher S Henry, Jennifer L Reed, Markus Krummenacker, Andrew R Joyce, Peter D Karp, Linda J Broadbelt, Vassily Hatzimanikatis, and Bernhard Ø Palsson. A genome-scale metabolic reconstruction for *Escherichia coli* k-12 mg1655 that accounts for 1260 orfs and thermodynamic information. *Molecular systems biology*, 3(1):121, 2007.

[231] F R Blattner, G Plunkett, C A Bloch, N T Perna, V Burland, M Riley, J Collado-Vides, J D Glasner, C K Rode, G F Mayhew, J Gregor, N W Davis, H A Kirkpatrick, M A Goeden, D J Rose, B Mau, and Y Shao. The complete genome sequence of *Escherichia coli* K-12. *Science (New York, N.Y.)*, 277(5331):1453–1462, September 1997.

[232] Hans Bremer and Patrick Dennis. Modulation of chemical composition and other parameters of the cell by growth rate. *Escherichia coli and Salmonella: cellular and molecular biology,*, 2:1553–1569, 1996.