STOCHASTIC AND PHYSICAL MODELING OF FUNDAMENTAL BIOLOGICAL PROCESSES

BY

TYLER M. EARNEST

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Physics
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2016

Urbana, Illinois

Doctoral Committee:

Associate Professor Yann R. Chemla, Chair
Professor Zaida Luthey-Schulten, Director of Research
Professor Taekjip Ha, Johns Hopkins University
Assistant Professor Thomas E. Kuhlman

Abstract

Modeling is a necessary tool to understand the large volumes of data generated from quantitative experiments on biological systems. It combines our knowledge of a phenomenon into a succinct mathematical or computational description. In this dissertation, we first describe briefly two applications of modeling in biophysics: loading of the replication clamp into the replisome in the archæon *Methanosarcina acetivorans* and genome packing initiation during the self-assembly of the T4 bacteriophage. We then describe in detail two systems: an improved model of the *lac* genetic switch which includes DNA looping in its gene regulation mechanism, and a spatially resolved, whole-cell model of ribosome biogenesis in *Escherichia coli*, which we then extend to include cell growth and replication of its genome.

For the first system, conditions and parameters affecting the range of bistability of the *lac* genetic switch in *E. coli* are examined for a model which includes DNA looping interactions with the *lac* repressor and a lactose analog. This stochastic gene–mRNA–protein model of the *lac* switch describes DNA looping using a third transcriptional state. We exploit the fast bursting dynamics of mRNA by combining a novel geometric burst approximation with the finite state projection (FSP) method. This limits the number of protein/mRNA states, allowing for an accelerated search of the model's parameter space. We evaluate how the addition of the third transcriptional state changes the bistability properties of the model and find a critical region of parameter space where the phenotypic switching occurs in a range seen in single molecule fluorescence studies. Stochastic simulations show induction in the looping model is preceded by a rare complete dissociation of the loop followed by an immediate burst of mRNA rather than a slower build up of mRNA as in the two-state model. The overall effect of the looped state is to allow for faster switching times while at the same time further differentiating the uninduced and induced phenotypes. Furthermore, the kinetic parameters are consistent with free energies derived from thermodynamic studies suggesting that this minimal model of DNA looping could have a broader range of application.

For the second system, we study the biogenesis of the ribosome. Central to all life is the assembly of the ribosome: a coordinated process involving the hierarchical association of ribosomal protein (r-protein) to the RNAs forming the small and large ribosomal subunits. The process is further complicated by effects arising from the intracellular heterogeneous environment and the location of ribosomal operons within the cell. We provide a simplified model of ribosome biogenesis in slow growing *E. coli.* Kinetic models of *in vitro* small subunit reconstitution at the level of individual r-protein to ribosomal RNA (rRNA) interactions are developed for two temperature regimes. The model at low temperatures predicts the existence of a novel $5' \rightarrow 3' \rightarrow$ central assembly pathway, which we investigate further using molecular dynamics. The high temperature assembly network is incorporated into a model of *in vivo* ribosome biogenesis in slow growing *E. coli.* The model, described in terms of reaction-diffusion master equations, contains 1336 reactions and 251 species that dynamically couple transcription and translation to ribosome assembly. We use the Lattice Microbes (LM) software package to simulate the stochastic production of mRNA, proteins, and ribosome intermediates over a full cell cycle of 120 minutes. The whole-cell model captures the correct growth rate of ribosomes, predicts the localization of early assembly intermediates to the nucleoid region, and reproduces the known assembly timescales for the small subunit with no modifications made to the embedded *in vitro* assembly network.

Finally, we extend the spatially resolved whole-cell model of ribosome biogenesis to include the effects of growth, DNA replication, and cell division. All biological processes are described in terms of reaction-diffusion master equations and solved stochastically using LM. In order to determine the replication parameters, we construct and analyze a series of *E. coli* strains with fluorescently labeled genes distributed evenly throughout their chromosomes. By measuring these cells' lengths and number of gene copies at the single-cell level, we could fit a statistical model of the initiation and duration of chromosome replication. We found that for our slow-growing (120 minute doubling time) *E. coli* cells, replication was initiated 42 minutes into the cell cycle and completed after an additional 42 minutes. While simulations of the biogenesis model produce the correct ribosome and mRNA counts over the cell cycle, the kinetic parameters for transcription and degradation are lower than anticipated from a recent analytical time dependent model of *in vivo* mRNA production. Describing expression in terms of a simple chemical master equation, we show that the discrepancies are due to the lack of non-ribosomal genes in the extended biogenesis model which effects the competition of mRNA for ribosome binding, and suggest corrections to parameters to be used in the whole-cell model when modeling expression of the entire transcriptome.

To Madelyn and Oliver for their infinite love and patience.

Acknowledgements

Science is not a single player game. I have been most fortunate to have the inspiration and assistance of many people, all of whom have directed the course of my studies at the University of Illinois. My undergraduate advisor at the South Dakota School of Mines and Technology was able to—with much reticence on my part—convince me to consider studying biological physics. My studies in the Department of Physics at UIUC took a slightly unconventional course. I had the opportunity to work with two other groups before finding the lab which was a perfect fit to my interests and abilities.

I would like to thank Paul Selvin for taking me on during my first year at UIUC and giving me the opportunity to discover just how bad I am at experimental work. In spite of that, I learned many details of single molecule biophysics that few computational and theoretical biologists are exposed to, giving me an unique perspective on analysis and modeling of experimental data.

I joined Karin Dahmen's group to work on theoretical problems in soft condensed matter, however she got me interested in biology once again and introduced me to Taekjip Ha and Zan Luthey-Schulten. Working with TJ and his group, I received my first exposure to theoretical and computational biology through the design and analysis of models describing the data collected from single molecule experiments. Through the collaboration between Karin and Zan, I was able to assemble a collection of theoretical tools by working with Michael Assaf, from whom I received my first introduction to the chemical master equation. Soon, Karin realized that my interests would be far better satisfied working with Zan and suggested joining her lab. I am forever grateful to her for her insightful suggestion, as it had lead to the most exciting and intellectually simulating

work on simulating whole cells.

I was introduced to whole-cell simulations by Zan Luthey-Schulten and her student Elijah Roberts who were easily able to convince me of the beauty and power of the method. I believe that these sorts of simulations have a great potential to revolutionize computational biophysics, and I feel incredibly lucky to have been involved at its infancy. During my time working in the Luthey-Schulten lab, I was able to work with many brilliant people: Elijah Roberts, Ke Chen, Jonathan Lai, John Cole, Joe Peterson, and Mike Hallock. I would also like to acknowledge the other group members who had to deal with my personality but were spared from working on a project with me directly: Marian Breuer, Zhaleh Ghaemi, Seth Thor, Piyush Labhsetwar, and Marcelo Melo. Though I am greatly indebted to Stack Overflow and Wikipedia, I have learned so much more about computing from Mike Hallock and John Stone.

I would like to thank my committee members—Yann Chemla, Zan Luthey-Schulten, Tom Kuhlman, and TJ Ha—for agreeing to guide me into PhD-hood. Yann Chemla graciously agreed to chair my committee in Karin Dahmen's stead in short notice, which I appreciate greatly. I am grateful to TJ Ha who gave me my first taste of biophysical modeling, and I am especially thankful to him for staying on my committee through his move to Johns Hopkins University. I have always looked forward to our meetings with Tom Kuhlman on our ribosome biogenesis study in progress for the stimulating discussions in which I always seem to learn something new about *E. coli*, and I am excited to continue our work. My advisor, Zan, however is who I am most indebted to. Through her support, I was afforded many opportunities that I doubt many other advisors would be able to offer. Though I accumulated a body of science knowledge before I joined her lab, Zan trained me to be a scientist.

Finally, I would like to thank my fiancée Madelyn McWilliams for being so patient and understanding. The life of a graduate student can be very demanding with a vanishing work/home life boundary, yet Madelyn handled the ups and downs with grace.

This work was supported in part by National Science Foundation through the programs Center of Physics of Living Cells (PHY–0822613, PHY–1430124) Physics of Living Systems Student Research Network (PHY–1026665, PHY–1505008) and the U.S. Department of Energy, Office of Science, *Biological and Environmental Research as part of the Adaptive Biosystems Imaging Scientific Focus Area. This material is based upon work supported by the U.S. Department of Energy, Office of Science, Office of Workforce Development for Teachers and Scientists, Office of Science Graduate Student Research (SCGSR) program. The SCGSR program is administered by the Oak Ridge Institute for Science and Education for the DOE under contract number DE–AC05–06OR23100. This research is part of the Blue Waters sustained-petascale computing project, which is supported by the National Science Foundation (awards OCI–0725070 and ACI–1238993) and the state of Illinois. Blue Waters is a joint effort of the University of Illinois at Urbana-Champaign and its National Center for Supercomputing Applications.*

Table of Contents

List of Tables

List of Figures

List of Algorithms

List of Abbreviations

CAP catabolite activator protein

CDF cumulative distribution function

CME chemical master equation

cryo-EM cryo-electron microscopy

DNAP DNA polymerase

E. coli *Escherichia coli*

EYFP enhanced yellow fluorescent protein

FRET fluorescence resonance energy transfer

FROS fluorescent repressor–operator system

FSP finite state projection

GPU graphics processing unit

LacI *lac* repressor

LacY β-galactoside permease

LacZ β-galactosidase

LM Lattice Microbes

LSU large subunit

MFPT mean first passage time

MPD-RDME multi-particle diffusion RDME

mRNA messenger RNA

PBS phosphate buffered saline

PCNA proliferating cell nuclear antigen

P/C qMS pulse/chase quantitative mass spectrometry

PDF probability distribution function

ppGpp guanosine tetraphosphate

r-protein ribosomal protein

RBM ribosome biogenesis model

RDME reaction–diffusion master equation

RFC replication factor C

RMS root mean square

RNAP RNA polymerase

rRNA ribosomal RNA

SAM semi-analytic model

SSA stochastic simulation algorithm

SSU small subunit

TetR *tet* repressor

TMG thiomethyl-β-D-galactoside

T. thermophilus *Thermus thermophilus*

WKB Wentzel–Kramers–Brillouin

YFP yellow fluorescent protein

Chapter 1

Introductory Theory and Methodology

The studies presented in this dissertation all revolve around some sort of system of chemical reactions. The appropriate theoretical treatment of these systems depends on the concentration and diffusivity of the species involved. Considering the limits of high to low concentration and fast to slow diffusion, leads to four different theoretical representations of the chemical system (Figure 1.1). For high concentration and fast diffusion, spatial dependence and particle number fluctuations can be ignored allowing for the use of deterministic chemical kinetics (Section 1.1). For low concentration and fast diffusion, we can no longer neglect particle number fluctuations can must consider all possible chemical copy number configurations (Section 1.3). For high concentration and slow diffusion spatial dependence must be considered, leading to a partial differential equation description, combining reaction dynamics and diffusion (Section 1.5). Finally with both low concentration and slow diffusion, considering both spatial dependence along with particle fluctuations requires a probabilistic, spatially resolved treatment (Section 1.6).

1.1 Deterministic chemical kinetics

In this work, we will be mainly interested in the dynamics of so called "elementary" chemical reaction kinetics. We will denote an arbitrary chemical reaction of X_i reactants forming Y_i products

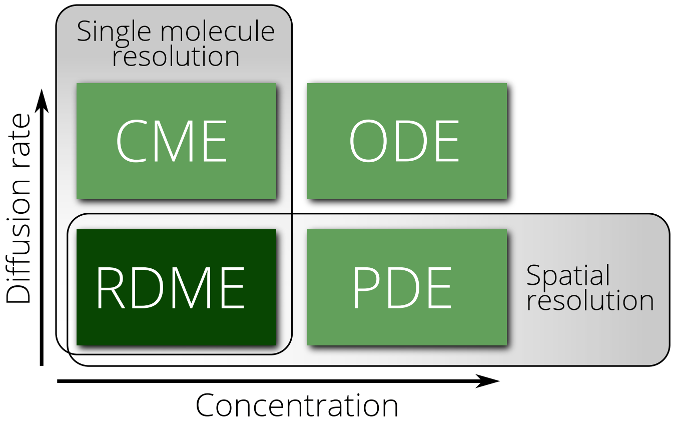


Figure 1.1 Theoretical treatments of biochemical systems are chosen based on how the diffusive timescale compares to the timescale of the phenomenon of interest and how significant the fluctuations due to discrete nature of particles are to the behavior of the system. A description of the system using deterministic rate equations (ODE) is sufficient when single particle fluctuations are insignificant and the dynamics are spatially homogeneous. When particle number fluctuations become important yet the system remains well-mixed, it is necessary to consider the system as a series of stochastic transitions between different copy number states whose dynamics are described by the chemical master equation (CME). If instead the chemicals are slow to diffuse, yet remain in high concentration, a description in terms of reaction–diffusion equations (PDE) must be used. Finally, when the system contains slowly diffusing species found in low copy numbers, the reaction–diffusion master equation (RDME) must be applied.

as

$$\sum_i \underline{\nu}_i \mathrm{X}_i \longrightarrow \sum_i \overline{\nu}_i \mathrm{Y}_i \tag{1.1}$$

where $\underline{\nu}_i$ is the stoichiometry for reactant i and $\overline{\nu}_i$ is the stoichiometry for product i. The rate of conversion through these elementary reactions take the form

$$\frac{\mathrm{d}y}{\mathrm{d}t} = k \prod_i c_i^{\underline{\nu}_i}, \tag{1.2}$$

where i indexes reactants, the extent of reaction y is defined to be

$$y(t) = \frac{c_j(t) - c_j(0)}{\overline{\nu}_j - \underline{\nu}_j}, \tag{1.3}$$

for any chemical species j and the chemical concentration of species j is

$$c_j = \frac{n_j}{N_\mathrm{A}\Omega}, \tag{1.4}$$

where n_i is the particle count, N_A is Avogadro's number, and Ω is the volume of the system. Eq. 1.2 is not true in general. Only if the reaction is elementary, i.e. the reaction occurs in a single step with a single transition state, does this theory apply. To apply this to non-elementary reactions, the reaction mechanism must be known so that each step can represented as an elementary reaction.

The form of Eq. 1.2 can be derived from collision theory, however a simple justification follows from the fact that the reacting molecules must find each other in the reacting volume Ω in order to react. This means that the reaction rate must be proportional to the rate of reaction encounters. The number of reaction encounters per unit time depends on the number of ways that the reactant particles can come together to form a reacting complex. For example, the dimerization

$$\mathrm{2A \longrightarrow B} \tag{1.5}$$

requires two species to combine. There are n_A species, and the number of possible interactions is $n_A(n_A-1)/2$ since in this reaction, a particle cannot interact with itself (the -1 term) and swapping the particles does not change the reaction (the $1/2$ term). This is clearly a binomial coefficient, and indeed the rate law can be written generally

$$\frac{\mathrm{d}y}{\mathrm{d}t} \propto \prod_i \binom{n_i}{\underline{\nu}_j}. \tag{1.6}$$

In macroscopic systems, e.g. $n_i \sim N_\mathrm{A}$, we can expand the binomial coefficient

$$\binom{n_i}{\underline{\nu}_i} = \prod_{k=0}^{\underline{\nu}_i}(n_i - k) = n_i{}^{\underline{\nu}_i} \prod_{k=0}^{\underline{\nu}_i}\left(1 - \frac{k}{n_i}\right) = n_i{}^{\underline{\nu}_i}\left[1 + O\left(\frac{1}{n_i}\right)\right] \tag{1.7}$$

and truncate to zeroth order, from which Eq. 1.2 follows from the definition of the concentration, Eq. 1.4.

In order to maintain consistent units in Eq. 1.2, the dimensions of the chemical rate constant, k, depends on the reaction order,

$$\alpha = \sum_i \underline{\nu}_i \tag{1.8}$$

as

$$[k] = \text{volume}^{\alpha-1}\text{time}^{-1}. \tag{1.9}$$

The reaction constant, k, encodes the details of the reaction kinetics. It depends on the temperature through the Arrhenius equation

$$k = \mathscr{A} \exp \frac{-E_\text{act}}{k_\text{B} T}, \tag{1.10}$$

with T the thermodynamic temperature, k_B Boltzmann's constant, and E_act is the activation energy of the reaction. Other details come about through the pre-exponential factor $\mathscr{A}$, such as the diffusion rates of the reactants, the encounter geometry, and other microscopic details.

Since the systems considered in this dissertation are composed of many chemical reactions, we must define a language to refer to them effectively. We will define the system of chemical equations as

$$\mathsf{S} \cdot \boldsymbol{X} = 0 \tag{1.11}$$

where we have defined the stoichiometric matrix,

$$\mathsf{S} = \overline{\boldsymbol{\nu}} - \underline{\boldsymbol{\nu}}, \tag{1.12}$$

and the stoichiometric vectors have been upgraded to matrices

$$(\underline{\boldsymbol{\nu}}, \overline{\boldsymbol{\nu}}) \in \mathbb{Z}^{N_\text{rxn} \times N_\text{sp}}, \tag{1.13}$$

and $\boldsymbol{X}$ symbolizes both the product and reactant chemical species. The system of chemical rate equations is then

$$\frac{\mathrm{d}\boldsymbol{c}}{\mathrm{d}t} = \mathsf{S}^\text{T} \cdot \boldsymbol{\Gamma} \tag{1.14}$$

where the flux vector is defined as

$$\Gamma_r = k_r \prod_{i=1}^{N_\text{sp}} {c_i}^{S_{ri}}. \tag{1.15}$$

However in a numerical solution to the reaction kinetics, the product in Eq. 1.15 is taken over an index set defined for each reaction to limit unproductive arithmetic.

1.2 Case study: Replication clamp loading during DNA replication*

In order for DNA replication to progress, DNA polymerase (DNAP) must be able to move between nucleobases quickly without disassociating from the replication fork. The faster it translocates along the DNA, the more likely a step would result in its disassociation. If this happens, replication would halt since DNAP would diffuse away. To prevent this, DNAP associates to a molecule acting as a sliding clamp. This clamp is a ring-shaped protein that goes around the DNA and prevents the DNAP from diffusing away if it dissociates from the replisome.

To get the clamp around the DNA requires a second protein, the clamp loader, to break the ring and place it around the strand. The currently accepted mechanism is that first the clamp loader docks with the sliding clamp, which has self-assembled from three monomers. Then the clamp loader breaks the ring, places it onto the strand, and closes it[3]. However, evidence from the archæon *Methanosarcina acetivorans* shows[2] that a second mechanism is possible (Figure 1.2a). The trimers that compose the clamp—called proliferating cell nuclear antigen (PCNA) in archaea—can assemble onto the clamp loader by using it as a template in an example of reverse-chaperoning. This was shown using an ensemble fluorescence resonance energy transfer (FRET) experiment where half of the clamp monomers were labeled with the fluorescent dye Cy5, a FRET acceptor, and the other half was labeled with Cy3, a FRET donor. The FRET efficiency is related to the number of clamp trimers assembled. It was discovered that upon the addition of clamp loader, called replication factor C (RFC) in archaea, the assembly kinetics of the clamp increased dramatically (Figure 1.2b). Numerous experimental tests were performed to ensure that this result was interpreted correctly[2]. It was also necessary to verify this claim (or at least to ensure consistency) through kinetic modeling.

[*] Section 1.2 contains material reproduced with permission from Liu C, McKinney MC, Chen YH, Earnest TM, Shi X, Lin LJ, Ishino Y, Dahmen K, Cann IKO, and Ha T (2011). Reverse-chaperoning activity of an AAA+ protein. *Biophys. J.*, **100**(5), pp. 1344–1352. doi:10.1016/j.bpj.2011.01.057.

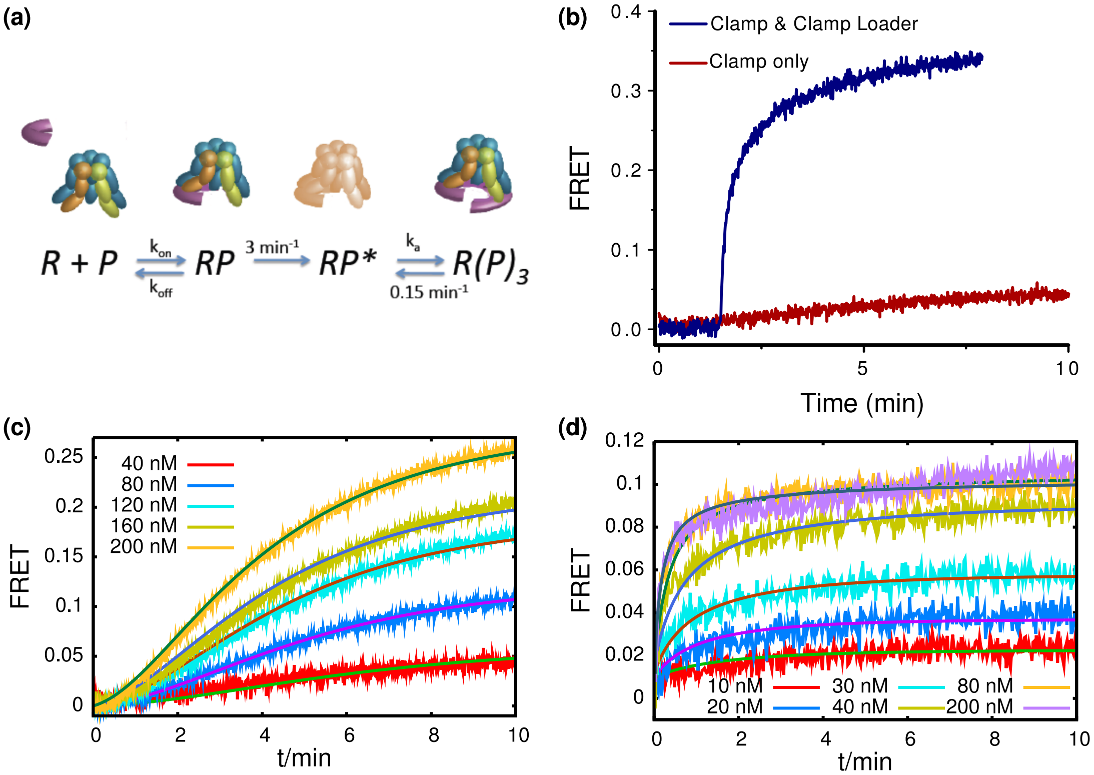


Figure 1.2 (a) Proposed mechanism of replication clamp assembly. Here *R* is clamp loader and *P* is clamp monomer. It is hypothesized that the complex formed from the first monomer binding to the clamp loader must under go a rate limiting step to form an activated complex which can then quickly assemble the remaining two monomers. (b) Bulk measurements of FRET efficiency allow the inference of the quantity of clamp trimers assembled as a function of time. When clamp loader is added at ~2 min, the rate of assembly increases significantly. FRET efficiency of the (c) self-assembly and (d) assisted assembly of the replication clamp, compared to the fitting of the ODE model (Eqs. 1.16a–1.16b). Due to differing experimental conditions, the absolute FRET efficiencies cannot be compared between the two experiments.

1.2.1 Modeling

To fit the ensemble FRET data to the proposed model, the model was adapted to include parallel self-assembly of PCNA and dissociation of the RFC–PCNA complex. The expanded model is

$$\mathrm{R+P} \underset{k'_o}{\overset{k_o}{\rightleftharpoons}} \mathrm{RP} \underset{k'_{act}}{\overset{k_{act}}{\rightleftharpoons}} \mathrm{RP^*+P} \underset{k'_{aa}}{\overset{k_{aa}}{\rightleftharpoons}} \mathrm{RP_2+P} \underset{k'_{ab}}{\overset{k_{ab}}{\rightleftharpoons}} \mathrm{RP_3} \underset{k'_d}{\overset{k_d}{\rightleftharpoons}} \mathrm{R+P_3} \tag{1.16a}$$

$$2P \underset{k'_{ta}}{\overset{k_{ta}}{\rightleftharpoons}} P+P_2 \underset{k'_{tb}}{\overset{k_{tb}}{\rightleftharpoons}} P_3 \tag{1.16b}$$

where R represents RFC and P represents a PCNA monomer. The full scheme used to compute the FRET efficiencies, includes three kinds of PCNA monomer species: P with no label, P' with a Cy3 tag, and P'' with a Cy5 tag. All possible combinations of the PCNA monomers are considered, increasing the number of equations to 27. The predicted signal from FRET is then proportional to the sum of the concentrations of all species which are composed of at least one P' and one P'' subunit. We follow the procedure of Brown and Sethna[4], to fit the solution of the system of chemical rate equations to the FRET efficiency data. To determine the goodness of fit, the cost function

$$C(\boldsymbol{\theta}) = \sum_{\xi=1}^{N_{\mathrm{ex}}} \mathcal{N}_\xi \sum_{i=1}^{N_t(\xi)} [F_\xi(t_i) - A_\xi y_\xi(t_i;\boldsymbol{\theta})]^2 + f(\boldsymbol{\theta}) \tag{1.17}$$

where the ξ index runs over the N_{ex} different experimental conditions, i runs over the $N_t(\xi)$ time points recorded for experiment ξ, t_i and $F_\xi(t_i)$ are the FRET measurements from time point i, $y_\xi(t)$ is the concentration of FRET active species predicted by the model, $\boldsymbol{\theta}$ represents the set of the logarithm of rate constants and

$$\mathcal{N}_\xi = \left(N_{\mathrm{ex}} N_t(\xi) [F_\xi(\infty)]^2\right)^{-1} \tag{1.18}$$

is a normalization factor to ensure that each experiment is considered equally regardless of signal magnitude, where $F_\xi(\infty)$ is the FRET signal at steady state. Since the rate constants span many different orders of magnitude, it is easier to write the cost function as a function of the logarithm of the rate constants $\theta_i = \ln k_i$. The term A_ξ is a scaling parameter which converts from FRET signal to concentration. Since the instrument function which maps the observed FRET signal to concentration is unknown, we assume direct proprotionality, and solve for the A_ξ which best fit the simulated concentration time courses. These prefactors can be obtained by solving $\partial_{A_\xi} C(\boldsymbol{\theta}) = 0$ for A_ξ,

$$A_\xi = \frac{\sum_{i=1}^{N_t(\xi)} F_\xi(t_i) y_\xi(t_i)}{\sum_{i=1}^{N_t(\xi)} [y_\xi(t_i)]^2}. \tag{1.19}$$

The nonleast-squares term $f(\boldsymbol{\theta})$ allows for finer control over the fit results, and is constructed from a sum of functions of the form

$$g(x; x_0, x_1, \alpha) = (x_0 - x)^\alpha \theta(x_0 - x) + (x - x_1)^\alpha \theta(x - x_1) \tag{1.20}$$

where $\theta(x)$ is the Heaviside function. These functions simply impose a penalty for values outside of the interval $[x_0, x_1]$. We use this function to assert our prior knowledge of the biochemistry of the system. This procedure is legitimate since the 14 rate constants are not all independent, i.e. there exist subspaces of the full parameter space for which the cost function is constant. The penalty function merely pushes the solver through these subspaces.

The penalty function is composed of a sum of four terms. The first term ensures that the rate constants are realistic, i.e. not greater than the diffusion limited rate of 10^9 $\mathrm{M}^{-1}\mathrm{s}^{-1}$. The second term imposes the fact that upon addition of RFC, the FRET signal increases by a factor of ~ 20, by ensuring that $y(\infty)/y(0) \approx 20$. The third term imposes that the dimer concentration is very low, by requiring $y_{\mathrm{tri}}/y_{\mathrm{di}} \ll 1$. Finally, the fourth term ensures that the calibration constants, A_ξ, are all similar. Since the self-assembly data were taken using the same PCNA stock, buffer, and incubation times, it is reasonable to assume that the distribution of calibration constants should have a small range compared to the mean. The self-assembly and assisted assembly data are considered separately since the data were collected at different conditions. The constraint function is constructed to minimize $(\max_\xi A_\xi - \min_\xi A_\xi)/\langle A \rangle$.

The chemical rate equations are integrated numerically. The cost function is minimized using simulated annealing[5] and quenched using the Nelder-Mead algorithm[6]. From the search we were

Table 1.1 Rate constants used in PCNA assembly model

Reaction	Parameter	Value	Units
$R + P \longrightarrow RP$	k_o	2.5726	$nM^{-1}min^{-1}$
$RP \longrightarrow R + P$	k'_o	1296.1	min^{-1}
$RP \longrightarrow RP^*$	k_{act}	9.5028	min^{-1}
$RP^* \longrightarrow RP$	k'_{act}	155.46	min^{-1}
$RP^* + P \longrightarrow RP_2$	k_{aa}	25.349	$nM^{-1}min^{-1}$
$RP_2 \longrightarrow RP^* + P$	k'_{aa}	8.8459×10^5	min^{-1}
$RP_2 + P \longrightarrow RP_3$	k_{ab}	2478.5	$nM^{-1}min^{-1}$
$RP_3 \longrightarrow RP_2 + P$	k'_{ab}	8.5213×10^{-7}	min^{-1}
$RP_3 \longrightarrow R + P_3$	k_d	15.944	min^{-1}
$R + P_3 \longrightarrow RP_3$	k'_d	71.678	$nM^{-1}min^{-1}$
$P + P \longrightarrow P_2$	k_{ta}	1.3798	$nM^{-1}min^{-1}$
$P_2 \longrightarrow P + P$	k'_{ta}	6.3886×10^5	min^{-1}
$P_2 + P \longrightarrow P_3$	k_{tb}	0.9943	$nM^{-1}min^{-1}$
$P_3 \longrightarrow P_2 + P$	k'_{tb}	0.21348	min^{-1}

Rate constants obtained through minimizing the RMS error between the measured and model predicted concentration time courses, subject to constraints (Eq. 1.17).

able to find a set of rate constants (Table 1.1) that fit the experimental data well (Figure 1.2cd). This shows that the kinetic model describing the assisted assembly is reasonable and could be the correct mechanism. Only in conjunction with other evidence[2] does it make a convincing argument.

1.3 Stochastic chemical kinetics

Eq. 1.14, being a deterministic, continuum treatment, does not capture the true nature of the reactive dynamics of a chemical system at low particle numbers. The times in which reactions occur are completely randomized due to Brownian motion of reactant and solvent molecules. Any memory of the prior state of the system is washed out after a timescale much shorter than the reaction timescale. The best that we can do is assign probabilities to the reactions and treat the system as a stochastic process. We assume that the system is "well-stirred", meaning that the diffusion timescale is much shorter than the reaction timescale, which allows us to ignore spatial dependence. We also assume that the chemical reaction follow a Poisson process with a rate which depends only on the current number of particles in the system. The defining equation of stochastic chemical kinetics in a "well-stirred" environment is the CME,

$$\frac{\mathrm{d}P}{\mathrm{d}t}(\boldsymbol{x},t) = \sum_{r=1}^{N_{\mathrm{rxn}}} a_r(\boldsymbol{x}-\boldsymbol{S}_r)P(\boldsymbol{x}-\boldsymbol{S}_r,t) - \sum_{r=1}^{N_{\mathrm{rxn}}} a_r(\boldsymbol{x})P(\boldsymbol{x},t) \tag{1.21}$$

where $a_r(\boldsymbol{x})$ is the reaction propensity for reaction r while the system is in state $\boldsymbol{x}$ (a.k.a. transition rate), and $\boldsymbol{x}$ is the state vector

$$\boldsymbol{x}(t) = [x_1(t) \quad x_2(t) \quad \cdots \quad x_{N_{\mathrm{sp}}}(t)]^{\mathrm{T}} \tag{1.22}$$

which enumerates the particle counts for each species in the system, and $\boldsymbol{S}_r$ is the row of the stochastic matrix that corresponds to the change in species numbers resulting from the reaction r. The first summation in Eq. 1.21 is the rate of probability entering the state $\boldsymbol{x}$ due to reactions from neighboring particle number states, while the second summation represents the rate of probability loss from $\boldsymbol{x}$ due to reactions leaving the state. The CME performs bookkeeping on the states: probability lost from one state is immediately recovered in another.

A justification for Eq. 1.21 can be found using simple probabilistic arguments. We define the transition rate $w_{i\to j}$, which gives the probability of a transition from state i to state j per unit time, to be constant with respect to time. The probability to find the system in state x after a short time has passed is

$$P(x,t+\mathrm{d}t) = P(x,t)P(x\circlearrowleft|t\in[t,t+\mathrm{d}t]) + \sum_{y\neq x} P(y,t)P(y\to x|t\in[t,t+\mathrm{d}t]) \tag{1.23}$$

where $P(x\circlearrowleft|t\in[t,t+\mathrm{d}t])$ is the conditional probability that if the system is in x, for $t\in[t,t+\mathrm{d}t]$ the system will not transition out of x. Likewise, $P(y\to x|t\in[t,t+\mathrm{d}t])$ is probability that the state transitions from y to x, conditioned on the time interval and initial state. These conditional probabilities can be written in terms of the transition rates

$$P(i \rightarrow j | t \in [t, t + \mathrm{d}t]) = w_{i\rightarrow j}\mathrm{d}t \tag{1.24a}$$

$$P(i \circlearrowleft | t \in [t, t + \mathrm{d}t]) = 1 - \sum_{j \neq i} w_{i\rightarrow j}\mathrm{d}t. \tag{1.24b}$$

Substituting Eq. 1.24a and Eq. 1.24b into Eq. 1.23 and rearranging produces

$$P(x, t + \mathrm{d}t) = P(x, t) - P(x, y) \sum_{y \neq x} w_{x\rightarrow y}\mathrm{d}t + \sum_{y \neq x} w_{y\rightarrow x} P(y, t)\mathrm{d}t + O(\mathrm{d}t^2). \tag{1.25}$$

The undetermined second order terms arise from the possibility of multiple transitions within $[t, t + \mathrm{d}t]$, and can be made insignificant with sufficiently small $\mathrm{d}t$. Identifying the pieces of the time derivative of $P(x, t)$, we rearrange Eq. 1.25 and take the limit $\mathrm{d}t \rightarrow 0$ to arrive at the CME,

$$\frac{\mathrm{d}P}{\mathrm{d}t}(x, t) = \sum_{x \neq y} w_{y\rightarrow x} P(y, x) - \sum_{y \neq x} w_{x\rightarrow y} P(x, y). \tag{1.26}$$

For a more rigorous derivation of the CME, see Gillespie[7].

Now we must compute the transition rates, i.e. reaction propensities. Again, we will only consider elementary reactions. The reaction propensity is

$$a_r(\boldsymbol{x}) = \kappa_r \prod_{i=1}^{N_{\mathrm{sp}}} \binom{x_i}{S_{ri}}, \tag{1.27}$$

which follows from the same argument as Eq. 1.2, in that the overall rate of a reaction is proportional to the number of ways to the reactants can be grouped. However, here κ_r is the "stochastic rate constant" not the deterministic rate constant k_r. They are related by

$$\kappa_r = (N_{\mathrm{A}}\Omega)^{1-\alpha} k_r \tag{1.28}$$

since the deterministic rate equations are defined in terms of concentrations, where as the CME is defined in terms of absolute numbers.

1.3.1 Stochastic simulations

Generally it is difficult, if not outright impossible to solve the CME for the system of interest. A way around this is to generate trajectories which attempt to sample the underlying probability distribution the CME describes. The most simple algorithm is the Gillespie Direct Method[8,9], also known as the stochastic simulation algorithm (SSA). Starting out with the initial species counts, $\boldsymbol{x}_0$, the stoichiometric matrix, $\mathbf{S}$, and the propensity functions, $a_i(\boldsymbol{x})$, defined for each reaction i, the algorithm steps forward in time by randomly choosing the identity and time of the next reaction event. The relative time that the next reaction fires is exponentially distributed, with rate equal to the sum of all reaction propensities, a_{total}. This is easy to see if you consider the CME for the current state and ignore incoming transitions,

$$\frac{\mathrm{d}P_{\text{react}}}{\mathrm{d}t} = -\sum_{r=1}^{N_{\text{rxn}}} a_r(\boldsymbol{x})P_{\text{react}} = -\left(\sum_{r=1}^{N_{\text{rxn}}} a_r(\boldsymbol{x})\right)P_{\text{react}} = -a_{\text{total}}P_{\text{react}}, \tag{1.29}$$

whose solution is

$$P_{\text{react}}(t) = a_{\text{total}}\mathrm{e}^{-a_{\text{total}}t}. \tag{1.30}$$

The probability that a reaction i fires is then simply

$$P_{\text{rxn}}(i) = \frac{a_i}{a_{\text{total}}}. \tag{1.31}$$

At each step of the SSA, a random reaction time $\tau \sim \text{Exp}(a_{\text{total}})$ is computed, along with a random reaction index $i \sim P_{\text{rxn}}(\boldsymbol{a})$. The current state is advanced by adding τ to the current time, and adding the net change of particles due to reaction i, i.e. the i^{th} row of the stoichiometric matrix to the current particle counts. Further details are given in Algorithm 1.1.

In Gillespie[9], an alternate algorithm was presented as well, called the First Reaction Method. It differs from the direct algorithm in that a putative reaction time,

$$\tau_i \sim \text{Exp}(a_i(\boldsymbol{x})) \tag{1.32}$$

```
Data: Initial particle counts – $\boldsymbol{x}_0$, stoichiometric matrix – $\mathbf{S}$, propensity functions – $a_r(\boldsymbol{x})$,
      and maximum evaluation time – $t_\mathrm{f}$.
Result: Reaction firing times – $\boldsymbol{T}$ and species counts at firing times – $\mathbf{X}$.
$t \longleftarrow 0$;
$\boldsymbol{x} \longleftarrow \boldsymbol{x}_0$;
// The number of events is not known in advance, initialize empty lists
$\mathbf{X} \longleftarrow \emptyset$;
$\boldsymbol{T} \longleftarrow \emptyset$;
while $t < t_\mathrm{f}$ do
    $a_\mathrm{total} \longleftarrow 0$;
    for $i \leftarrow 1$ to $N_\mathrm{rxn}$ do
        $a_\mathrm{total} \longleftarrow a_\mathrm{total} + a_i(\boldsymbol{x})$;
    $\rho_1 \longleftarrow$ uniformRand();
    $\rho_2 \longleftarrow$ uniformRand();
    $\tau \longleftarrow -\frac{\log \rho_1}{a_\mathrm{total}}$;                    // $\tau \sim \mathrm{Exp}(a_\mathrm{total})$
    for $i \leftarrow 1$ to $N_\mathrm{rxn}$ do         // Reaction choice weighted by propensity
        if $a_{r-1}(\boldsymbol{x}) < \rho_2 \cdot a_\mathrm{total} \le a_r(\boldsymbol{x})$ then
            $\boldsymbol{x} \longleftarrow \boldsymbol{x} + \boldsymbol{S}_i$;      // Row $i$ of $\mathbf{S}$ is the net change due to reaction $i$
            break;
    $t \longleftarrow t + \tau$;
    append($\mathbf{X}, \boldsymbol{x}$);
    append($\boldsymbol{T}, t$);
```

Algorithm 1.1 Direct stochastic simulation algorithm

is computed for all reactions each time step. The smallest τ_i identifies both the time and the reaction that fires. These two algorithms are mathematically equivalent[9], however the direct method is more computationally efficient.

Since Gillespie's algorithms were published in 1976, many improved algorithms have been published. From the *for* loops in Algorithm 1.1, it is clear that the computational complexity of the algorithm is $O(N_{\mathrm{rxn}})$. The Next Reaction Method[10], which improves upon the First Reaction Method, is able to achieve $O(\log N_{\mathrm{rxn}})$ complexity while only requiring a single random number per reaction event. The main feature of this method is that it saves the absolute time that each reaction fires, i.e. $t + \tau_i$, for each reaction, and only updates the times if reactions occurred that change the value of the reaction propensity. Techniques have been developed which improve upon the direct method such as partial propensity calculations[11–13] which are $O(N_{\mathrm{sp}})$ instead of $O(N_{\mathrm{rxn}})$, and methods which sort the reactions by propensity to decrease the number of iterations necessary to find a reaction[11,14,15], among others. There are also approximate methods which are appropriate for large particle numbers such as tau leaping[16,17] or for systems with a separation of timescales[18]

1.3.2 Selected features of stochastic chemical systems

To show the necessity for a stochastic, particle-orientated point of view in certain situations, we will investigate two simple systems of reactions.

Michaelis-Menten kinetics

The most simple and well-known model of enzyme catalysis is the Michaelis-Menten model[19],

$$\mathrm{E} + \mathrm{S} \underset{k_{\mathrm{off}}}{\overset{k_{\mathrm{on}}}{\rightleftharpoons}} \mathrm{ES} \xrightarrow{\mathrm{k_{cat}}} \mathrm{E} + \mathrm{P} \tag{1.33}$$

where the substrate S, binds to the active site of the enzyme E, to form the complex ES, which can then either react releasing the product P, or release the substrate. To investigate how low particle numbers affect the time course of the conversion of S to P, we have plotted the deterministic solution along with stochastic trajectories at four system sizes (Figure 1.3a). In order for the stochastic trajectories to match the deterministic solution at each system size, the rate constants must be scaled along with the initial conditions. For example, the equation for the enzyme concentration

$$\frac{\mathrm{d}c_\mathrm{E}}{\mathrm{d}t} = -k_\mathrm{on} c_\mathrm{E} c_\mathrm{S} + (k_\mathrm{off} + k_\mathrm{cat}) c_\mathrm{ES} \tag{1.34}$$

under the transformation

$$c_i \to \chi c_i \tag{1.35a}$$

$$k_\mathrm{off} \to \alpha k_\mathrm{off} \tag{1.35b}$$

$$k_\mathrm{on} \to \beta k_\mathrm{on} \tag{1.35c}$$

$$k_\mathrm{cat} \to \alpha k_\mathrm{cat} \tag{1.35d}$$

becomes

$$\frac{\mathrm{d}}{\mathrm{d}t}(\chi c_\mathrm{E}) = -\beta k_\mathrm{on}(\chi c_\mathrm{E})(\chi c_\mathrm{S}) + \alpha(k_\mathrm{off} + k_\mathrm{cat})(\chi c_\mathrm{ES}), \tag{1.36}$$

is invariant if $\alpha = 1$ and $\beta = 1/\chi$. As the initial count of substrate molecules increases from 10 to 10000 in Figure 1.3a the trajectories approach the deterministic solution. Figure 1.3b shows the deterministic solution juxtaposed with 100 stochastic realizations at a constant system size ($n_\mathrm{S}(0) = 100$, $n_\mathrm{E}(0) = 10$), where some trajectories can be seen to vary from the deterministic solution by 20%. Such low copy numbers are common in single cells.

Lotka-Volterra model

A more profound effect of stochasticity can be seen in the Lotka-Volterra predator–prey model[19],

$$Y \xrightarrow{\alpha} 2Y \tag{1.37a}$$

$$R + Y \xrightarrow{\beta} 2R \tag{1.37b}$$

$$R \xrightarrow{\gamma} \varnothing \tag{1.37c}$$

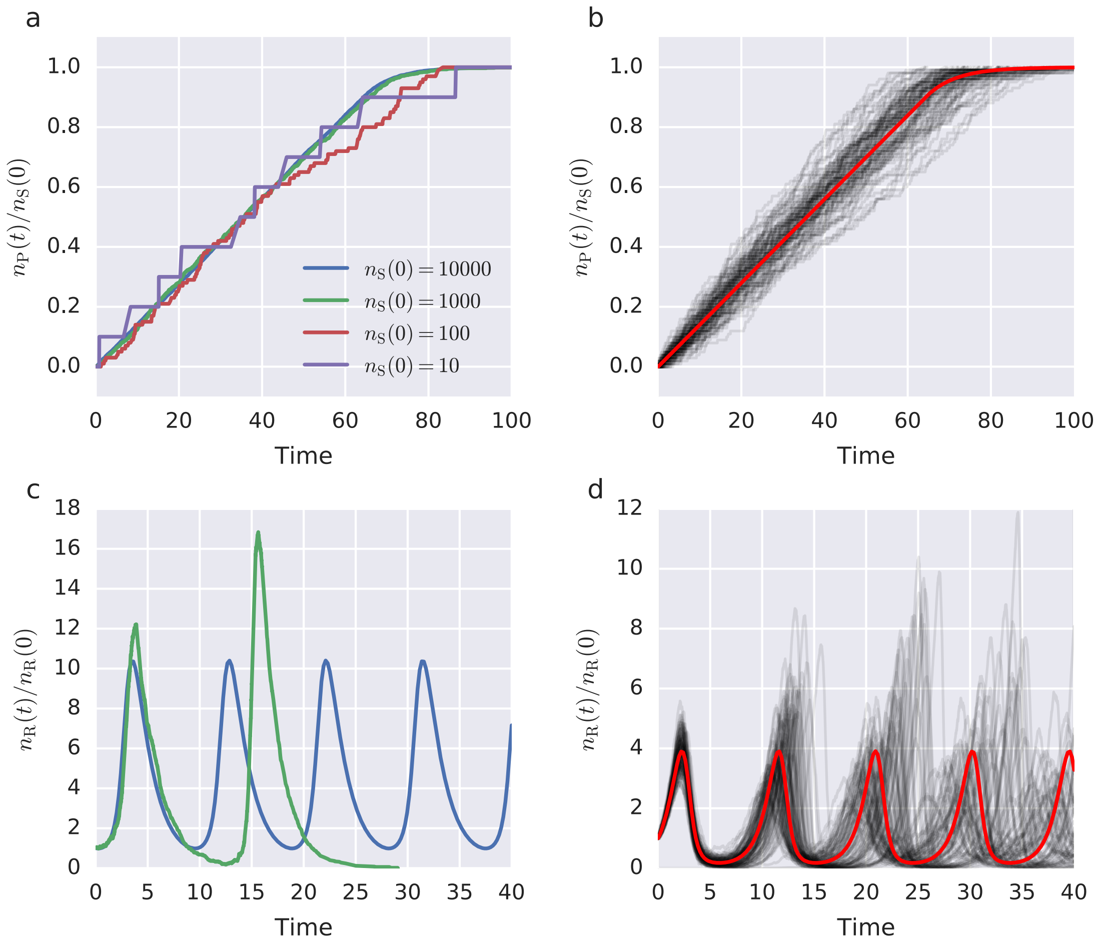


Figure 1.3 Stochasticity in Michaelis-Menten Kinetics. (a) Product of enzymatic conversion at varying system sizes. (b) Many stochastic trajectories with the same parameters and initial conditions (black) compared to the deterministic solution (red).
Stochasticity in the Lotka-Volterra model. (c) Comparison of the deterministic solution to a stochastic realization of the predator counts. Stochastic treatment of the model reveals the existence of extinction events. The initial predator population is 50. At $t = 28$, both population sizes reach zero. (d) Many stochastic realizations (black) compared to the deterministic solution (red). Stochasticity leads to fluctuations in the phase with respect to the deterministic solution.

where Y indicates “prey” species which can reproduce at a rate proportional to the number of prey and R indicates “predator” species, which can either consume a prey species to create another predator or die of natural causes. The deterministic equations are

$$\frac{\mathrm{d}n_\mathrm{Y}}{\mathrm{d}t} = \alpha n_\mathrm{Y} - \beta n_\mathrm{Y} n_\mathrm{R} \tag{1.38a}$$

$$\frac{\mathrm{d}n_\mathrm{R}}{\mathrm{d}t} = \beta n_\mathrm{Y} n_\mathrm{R} - \gamma n_\mathrm{R} \tag{1.38b}$$

and have three fixed points: $n_\mathrm{Y} = 0, n_\mathrm{R} = 0$; $n_\mathrm{Y} = \infty, n_\mathrm{R} = 0$; and $n_\mathrm{Y} = \gamma/\beta, n_\mathrm{R} = \alpha/\beta$ corresponding to the extinction of Y, the extinction of R, and stable coexistence. Aside from the coexistence fixed point and the lines $n_\mathrm{Y} = 0$ and $n_\mathrm{R} = 0$, any set of initial conditions will lead to oscillatory solutions where the predator population lags the prey population.

Consider a stochastic version of the system. If a fluctuation of the population of Y causes $n_\mathrm{Y} \to 0$, it can never recover since the reaction propensity to produce Y is zero. Subsequently, the population of R will fall to zero since the reaction propensity to produce R is also zero. This phenomenon of extinction arises only once stochasticity is taken into account in the Lotka-Volterra system. Figure 1.3c shows an example of this extinction behavior. Another effect of stochasticity in this system is that the population time courses of independent realizations rapidly fall out of phase with the deterministic solution (Figure 1.3d).

1.4 Case study: Viral capsid DNA packing[†]

When the bacteriophage T4 is replicated its entire genome must be packed into its preassembled capsid—the outer protein shell of the virus[21]. To accomplish this, a motor protein called gp17 is used to force the genetic material into the capsid. To investigate the initiation of packing, the mechanism in which the DNA starts being packed by the motor, single molecule studies using capsid–motor complexes were performed. The viral capsids were immobilized on a microscope

[†] Section 1.4 contains material reproduced with permission from Vafabakhsh R, Kondabagil K, Earnest T, Lee KS, Zhang Z, Dai L, Dahmen KA, Rao VB, and Ha T (2014). Single-molecule packaging initiation in real time by a viral DNA packaging machine from bacteriophage T4. *Proc. Natl. Acad. Sci. USA*, **111**(42), pp. 15096–15101. doi:10.1073/pnas.1407235111.

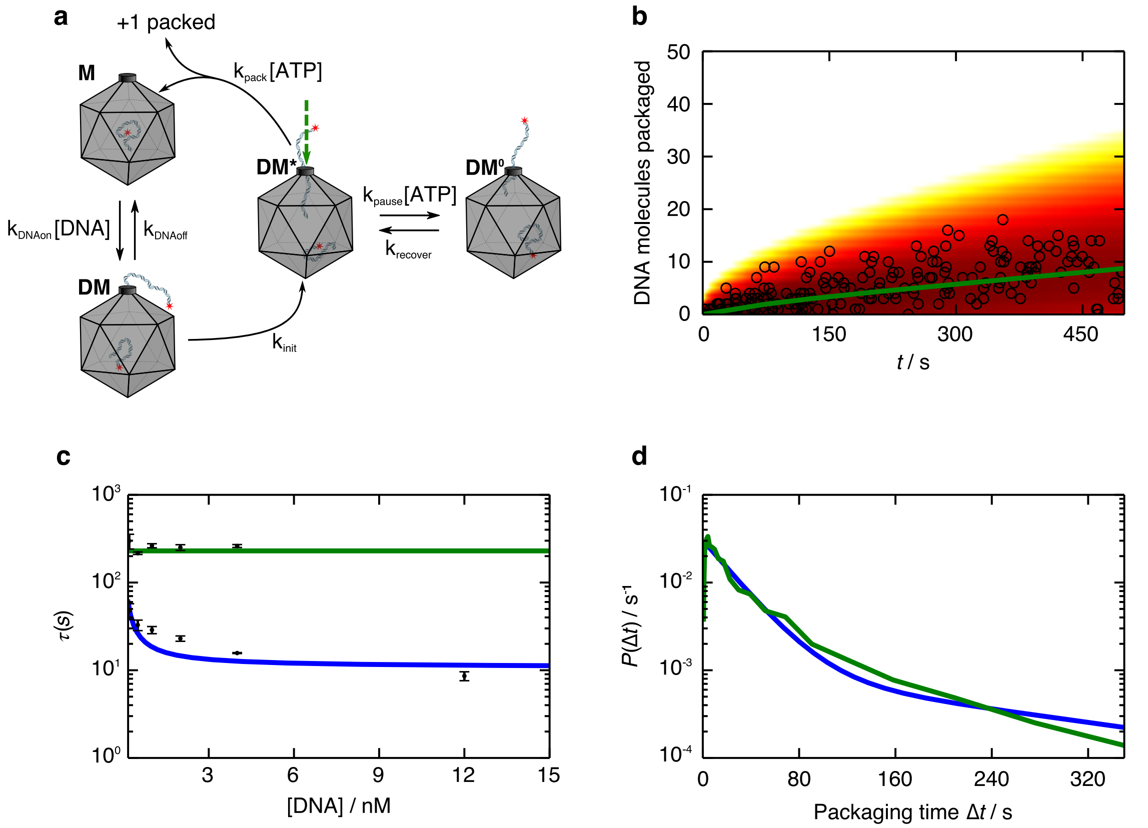


Figure 1.4 (a) Depiction of a packaging model in which DNA binding triggers a conformational change that activates the motor. The activated complex then either packages the DNA or enters a paused state. (b) The fit obtained from the proposed model (green line) to the experimental data for the number of packaged DNA molecules over time (open circles). The heat map shows the probability distribution to find a number of molecules packed at a point in time. (c) Prediction of the model for the short packaging time (blue line) and long packaging time (green line) as a function of DNA concentration. (d) Packaging time distribution predicted from model (blue line) compared to experimental distribution (green line).

coverslip with the gp17 motor installed. Small DNA molecules labeled with a fluorescent dye can be flowed into the coverslip/slide cell and be packed by the motor. When the DNA is packed, the fluorescent molecule becomes tethered to the coverslip. By monitoring the intensity of a spot over time, jumps in the fluorescence intensity can be associated with DNA packing events. The labeled DNA are seen to pack in rapid bursts followed by periods of inactivity. They are also seen to stop packing entirely. Similar research has been published recently[22] on the stop-start behavior of viral genome packing in T4, however this is for much longer strands of DNA and used optical trap techniques.

An order of magnitude difference between the short and long time constants suggests that the packaging initiation events occur in bursts, with periods of activity where multiple DNA molecules are packed consecutively, punctuated by long pauses. This bursting behavior could be produced if the T4 motor can enter a quiescent state where it is trapped in an inactive conformation unable to translocate. One model illustrating such a cycle is depicted in Figure 1.4a. In this model the packaging complex **M** initiates packaging by first associating with a DNA molecule at a rate proportional to the DNA concentration. DNA binding can trigger a conformational change in the motor that results in the transition of the packaging complex **DM** into an activated state **DM*** from which translocation can begin. Packaging then completes at an ATP-dependent rate. However, from this activated **DM*** state the motor can transit into an inactive, quiescent state ($\mathbf{DM^0}$) with a rate that is dependent on ATP concentration. In this state, the motor pauses, possibly because ATP binding and DNA capture are not coordinated. Finally, the motor recovers from the pause and resumes packaging initiation.

This model can be expressed through the master equations

$$\partial_t P_n^{\mathbf{M}} = k_{\text{pack}}[\text{ATP}]P_{n-1}^{\mathbf{DM}^*} + k_{\text{dnaoff}}P_n^{\mathbf{DM}} - k_{\text{dnaon}}[\text{DNA}]P_n^{\mathbf{M}} \tag{1.39a}$$

$$\partial_t P_n^{\mathbf{DM}} = k_{\text{dnaon}}[\text{DNA}]P_n^{\mathbf{M}} - (k_{\text{dnaoff}} + k_{\text{init}})P_n^{\mathbf{DM}} \tag{1.39b}$$

$$\partial_t P_n^{\mathbf{DM}^*} = k_{\text{init}}P_n^{\mathbf{DM}} + k_{\text{unpause}}P_n^{\mathbf{DM^0}} - (k_{\text{pause}}[\text{ATP}] + k_{\text{pack}}[\text{ATP}])P_n^{\mathbf{DM}^*} \tag{1.39c}$$

$$\partial_t P_n^{\mathbf{DM^0}} = k_{\text{pause}}[\text{ATP}]P_n^{\mathbf{DM}^*} - k_{\text{unpause}}P_n^{\mathbf{DM^0}}, \tag{1.39d}$$

where $P_n^{\mathbf{X}}(t)$ is the probability mass function to find the motor in the state **X** with n molecules packed at time t. The experimental data describing the packing process is in the form of DNA molecules counted per viral motor. We fit this model to that data by maximizing the likelihood function

$$L(\boldsymbol{\theta}) = \prod_{i=1}^{N_{\text{expt.}}} \prod_{j=1}^{N_{\text{motor},i}} \prod_{k=1}^{N_{\text{event},i,j}} P_{n_{i,j,k}}(t_{i,j,k}|[\text{ATP}]_i, [\text{DNA}]_i, \boldsymbol{\theta}), \tag{1.40a}$$

over

$$\boldsymbol{\theta} = \{k_{\text{dnaon}}, k_{\text{dnaoff}}, k_{\text{init}}, k_{\text{pause}}, k_{\text{unpause}}, k_{\text{pack}}\}, \tag{1.40b}$$

where the probability to find a capsid with exactly n packed molecules at time t is

$$P_n(t) = P_n^{\mathbf{M}}(t) + P_n^{\mathbf{DM}}(t) + P_n^{\mathbf{DM}^*}(t) + P_n^{\mathbf{DM^0}}(t). \tag{1.41}$$

The experimental data is described through $t_{i,j,k}$ and $n_{i,j,k}$ which are the packing event times and number of molecules packed respectively for experiment i, capsid j, and event k, where the packing time is measured starting from the addition of DNA. $N_{\text{expt.}}$ is the number of experimental conditions, $N_{\text{motor},i}$ is the number of immobilized capsid/motor complexes for experiment i, and $N_{\text{event},i,j}$ is the number of packing events for motor j.

1.4.1 Numerical solution

To compute Eq. 1.41 we must solve Eqs. 1.39a–1.39d, however an analytic solution will be enormously complicated. Instead, we take advantage of the fact that it is highly unlikely that there will be more than ~50 molecule packing events observed for a single motor. If we modify Eqs. 1.39a–1.39d to only include terms up to a finite N, we receive a linear system of ODEs with $4N$ equations which can be solved numerically. We replace the transitions between $P_{N-1}^{\mathbf{DM}^*}$ to $P_N^{\mathbf{M}}$ with a transition to an absorbing boundary state, ϵ. No transitions from ϵ are allowed, so the probability to find the system in that state increases monotonically with time. To choose the optimal N such that the error in the solution is acceptable and the number of states of the system is minimized, we use $P_\epsilon(t)$ to monitor the total probability lost due to the truncation. If $P_\epsilon(t)$ increases above a threshold, we must restart the solution with a larger value of N. This procedure forms the basis of the finite state projection (FSP)[23,24] which will be discussed in detail in Section 2.3.

Armed with a recipe to compute Eq. 1.41, we use the Nelder-Mead algorithm to minimize the objective function

$$\Phi(\boldsymbol{\theta}) = -\log L(\boldsymbol{\theta}), \tag{1.42}$$

where we take the logarithm in order to ensure that the numerical evaluation of the objective function does not exceed the range of a standard 64-bit floating point number. Figure 1.4b shows an example of the fitting of a single experiment, the long and short packaging time constants,

Table 1.2 Four-state T4 capsid motor packing model parameters.

Parameter	Value	Units
k_{dnaon}	355.1 ± 5.9	$\text{nM}^{-1}\text{s}^{-1}$
k_{dnaoff}	2 ± 103	s^{-1}
k_{init}	761.0 ± 4.0	s^{-1}
k_{unpause}	0.02 ± 0.18	s^{-1}
k_{pause}	383.5 ± 7.0	$\text{mM}^{-1}\text{s}^{-1}$
k_{pack}	515.8 ± 6.6	$\text{mM}^{-1}\text{s}^{-1}$

Model parameters are computed from maximizing Eq. 1.41. Parameter values are given as mean±std, computed from bootstrapping.

predicted from this model, are shown in Figure 1.4d, and an example of the predicted inter-packing time distribution for an experiment is shown in Figure 1.4e.

1.4.2 Analytic solution

Since the transition probabilities do not depend on the number of molecules packed, we can simplify the problem by considering the process of packing a single molecule,

$$\partial_t P_0^{\mathbf{M}} = k_{\text{dnaoff}} P_0^{\mathbf{DM}} - k_{\text{dnaon}}[\text{DNA}] P_0^{\mathbf{M}} \tag{1.43a}$$

$$\partial_t P_0^{\mathbf{DM}} = k_{\text{dnaon}}[\text{DNA}] P_0^{\mathbf{M}} - (k_{\text{dnaoff}} + k_{\text{init}}) P_0^{\mathbf{DM}} \tag{1.43b}$$

$$\partial_t P_0^{\mathbf{DM}^*} = k_{\text{init}} P_0^{\mathbf{DM}} + k_{\text{unpause}} P_0^{\mathbf{DM}^\mathbf{0}} - (k_{\text{pause}}[\text{ATP}] + k_{\text{pack}}[\text{ATP}]) P_0^{\mathbf{DM}^*} \tag{1.43c}$$

$$\partial_t P_n^{\mathbf{DM}^\mathbf{0}} = k_{\text{pause}}[\text{ATP}] P_0^{\mathbf{DM}^*} - k_{\text{unpause}} P_0^{\mathbf{DM}^\mathbf{0}} \tag{1.43d}$$

$$\partial_t P_1^{\mathbf{M}} = k_{\text{pack}}[\text{ATP}] P_0^{\mathbf{DM}^*}, \tag{1.43e}$$

and assume that the motor in state **M** with one molecule packed is an absorbing state. Then, $P_1^{\mathbf{M}}(\Delta t)$ can be interpreted as the cumulative distribution function (CDF) of packing times, Δt, and it then follows that Eq. 1.43e yields the probability distribution function (PDF) of packing times,

$$P_{\Delta t}(t) = \partial_t P_1^{\mathbf{M}}(t) = k_{\text{pack}}[\text{ATP}] P_0^{\mathbf{DM}^*}. \tag{1.44}$$

A closed-form expression for Eq. 1.44 can be derived by first computing the Laplace transform of Eqs. 1.43a–1.43e and solving for $P_1^{\mathbf{M}}(s)$:

$$\hat{P}_{\Delta t}(s) = \frac{k_{\text{init}} k_{\text{dnaon}} k_{\text{pack}} [\text{DNA}] [\text{ATP}] (s + k_{\text{unpause}})}{(s+k_1)(s+k_2)(s+k_3)(s+k_4)}, \tag{1.45}$$

where the rate constants are:

$$\begin{aligned}
k_1 &= \frac{1}{2}\Bigg(k_{\text{init}} + k_{\text{dnaoff}} + k_{\text{dnaon}} [\text{DNA}] \\
&\qquad + \sqrt{(k_{\text{init}} + k_{\text{dnaoff}} + k_{\text{dnaon}} [\text{DNA}])^2 - 4 k_{\text{init}} k_{\text{dnaon}} [\text{DNA}]} \Bigg) && \text{(1.46a)} \\
k_2 &= \frac{1}{2}\Bigg(k_{\text{init}} + k_{\text{dnaoff}} + k_{\text{dnaon}} [\text{DNA}] \\
&\qquad - \sqrt{(k_{\text{init}} + k_{\text{dnaoff}} + k_{\text{dnaon}} [\text{DNA}])^2 - 4 k_{\text{init}} k_{\text{dnaon}} [\text{DNA}]} \Bigg) && \text{(1.46b)} \\
k_3 &= \frac{1}{2}\Bigg(\big(k_{\text{pack}} + k_{\text{pause}}\big) [\text{ATP}] + k_{\text{unpause}} \\
&\qquad + \sqrt{\Big[\big(k_{\text{pack}} + k_{\text{pause}}\big) [\text{ATP}] + k_{\text{unpause}}\Big]^2 - 4 k_{\text{pack}} k_{\text{unpause}} [\text{ATP}]} \Bigg) && \text{(1.46c)} \\
k_4 &= \frac{1}{2}\Bigg(\big(k_{\text{pack}} + k_{\text{pause}}\big) [\text{ATP}] + k_{\text{unpause}} \\
&\qquad - \sqrt{\Big[\big(k_{\text{pack}} + k_{\text{pause}}\big) [\text{ATP}] + k_{\text{unpause}}\Big]^2 - 4 k_{\text{pack}} k_{\text{unpause}} [\text{ATP}].} \Bigg) && \text{(1.46d)}
\end{aligned}$$

To compute the time domain PDF of packing times, we compute the inverse Laplace transform of Eq. 1.45 by explicitly evaluating the contour integral

$$P_{\Delta t}(t) = \frac{1}{2\pi \mathrm{i}} \lim_{T \to \infty} \int_{\gamma - \mathrm{i}T}^{\gamma + \mathrm{i}T} \mathrm{d}s \, \mathrm{e}^{st} \hat{P}_{\Delta t}(s) \tag{1.47}$$

using the residue theorem. This yields a first passage time distribution in the form of a sum of exponentials,

$$\begin{aligned}P_{\Delta t}(t|[\mathrm{ATP}],[\mathrm{DNA}]) = k_{\mathrm{init}}k_{\mathrm{dnaon}}k_{\mathrm{pack}}[\mathrm{DNA}]\,[\mathrm{ATP}]\Bigg(&\frac{(k_{\mathrm{unpause}}-k_1)\mathrm{e}^{-k_1 t}}{(k_2-k_1)(k_3-k_1)(k_4-k_1)} \\ &+\frac{(k_{\mathrm{unpause}}-k_2)\mathrm{e}^{-k_2 t}}{(k_1-k_2)(k_3-k_2)(k_4-k_2)}+\frac{(k_{\mathrm{unpause}}-k_3)\mathrm{e}^{-k_3 t}}{(k_1-k_3)(k_2-k_3)(k_4-k_3)} \\ &+\frac{(k_{\mathrm{unpause}}-k_4)\mathrm{e}^{-k_4 t}}{(k_1-k_4)(k_2-k_4)(k_3-k_4)}\Bigg).\end{aligned} \tag{1.48}$$

1.5 Reaction/diffusion systems

Though reaction–diffusion equation modeling is not used in this dissertation, a brief introduction for the sake of symmetry follows. For chemical systems for which stochastic effects are not important but cannot be assumed to be well-stirred, the diffusion equation can be combined with a reaction term to form the reaction–diffusion equation,

$$\partial_t c_i = \boldsymbol{\nabla}\cdot(\mathbf{D}_i\cdot\boldsymbol{\nabla}c_i)+\boldsymbol{f}(\boldsymbol{c}), \tag{1.49}$$

where $c_i(\boldsymbol{x},t)$ is the space- and time-dependent concentration for species i, $\mathbf{D}_i$ is the (potentially spatially dependent) diffusion tensor for species i, and $\boldsymbol{f}(\boldsymbol{c})$ is a term representing the gain and loss of material due to chemical reactions. If only elementary reactions are considered, then

$$\boldsymbol{f}(\boldsymbol{c}) = \mathbf{S}^{\mathrm{T}}\cdot\boldsymbol{\Gamma}(\boldsymbol{c}(\boldsymbol{x},t)) \tag{1.50}$$

where $\mathbf{S}$ is the stoichiometric matrix (Eq. 1.12) and $\boldsymbol{\Gamma}$ is the flux vector (Eq. 1.15) which now depends on space implicitly.

As a simple example whose result will become useful once we begin to study the ribosome biogenesis model developed in Chapter 3, consider a system with the following reactions,

$$\varnothing \xrightarrow{\lambda} A \tag{1.51a}$$

$$A \xrightarrow{\gamma} \varnothing \tag{1.51b}$$

where the birth process (Eq. 1.51a) only occurs at a point source located at the origin of the reaction volume, Ω, which we take to be all of $\mathbb{R}^3$. The reaction–diffusion equation is then simply

$$\partial_t \phi = D\nabla^2 \phi - \gamma\phi + \lambda\delta(\boldsymbol{x}) \tag{1.52}$$

with D the diffusion coefficient, γ the first order decay rate of A, λ the zeroth order birth rate of A, and $\phi = \phi(\boldsymbol{x}, t)$ the time-dependent concentration field of A, with the boundary conditions

$$\begin{cases} \phi(\boldsymbol{x}, t) = 0 \\ \boldsymbol{\nabla}\phi(\boldsymbol{x}, t) = 0 \end{cases}, \qquad \forall\, t \geq 0,\, \boldsymbol{x} \in \partial\Omega \tag{1.53}$$

This PDE is not difficult to solve. First, we Fourier transform Eq. 1.52 over $\boldsymbol{x}$,

$$\partial_t \hat{\phi} = -D(k^2 + \gamma)\hat{\phi} + \lambda \tag{1.54}$$

and solve the resulting ODE for the time-dependent, Fourier-transformed concentration field

$$\hat{\phi}(\boldsymbol{k}, t) = \frac{\lambda/D}{k^2 + \gamma/D} + \hat{\phi}_0(\boldsymbol{k})\mathrm{e}^{-(k^2+\gamma)Dt}. \tag{1.55}$$

Here our constant of integration was $\hat{\phi}_0(\boldsymbol{k})$, which is the Fourier transformed initial concentration. For the sake of simplicity, let us take $\hat{\phi}_0(\boldsymbol{k}) = n_0$, i.e. a delta distributed initial concentration profile of total amount n_0. Now transforming Eq. 1.55 back, we use the central symmetry of the problem to simplify the Fourier integral and arrive at the solution

$$\phi(r, t) = \frac{\lambda}{4\pi D r}\mathrm{e}^{-\sqrt{\gamma/D}r} + \frac{n_0}{\sqrt{4\pi D t}}\mathrm{e}^{-\gamma D t}\mathrm{e}^{-r^2/4Dt}, \tag{1.56}$$

where the first term is the steady-state concentration profile arising from the central source, and the second term is the usual diffusion kernel arising from the initial amount of chemical at the origin at $t = 0$. This result is applicable to whole-cell simulations where a species is produced at a point in space at a constant rate and has a fast decay time. For example, the 16S rRNA which forms the foundation of the ribosomal small subunit is transcribed from fixed locations within the cell. In order to see any appreciable spatial heterogeneity of the fully assembled particles, we would have to have that

$$\sqrt{\frac{D}{\gamma}} < \ell \tag{1.57}$$

where ℓ is the length of the cell. Taking values of $D = 0.5\ \mu\mathrm{m}^2\,\mathrm{s}^{-1}$ for the diffusion constant of assembly intermediates and $\ell = 4.0\ \mu\mathrm{m}$ for the cell length, the half life of assembly intermediates would have to be on the order of 5 s. This is not the case, however as we will see later the half lives of particular classes of assembly intermediates can be much smaller than this.

1.6 Stochastic chemical kinetics with spatial resolution

In vitro systems can contain small copy numbers of chemical species which diffuse slowly through a complex, crowded environment. To correctly model these sorts of systems requires a stochastic, spatially resolved approach. There are two classes of simulation algorithms for these sorts of problems. First, particle-based methods[25–27] track the position and identity of each particle in space, and evolve their positions in time using Brownian dynamics where the position of each particle i is updated as

$$\boldsymbol{x}(t_{i+1}) = \boldsymbol{x}(t_i) + \frac{1}{\zeta_i}\boldsymbol{f}_i(\{\boldsymbol{x}\}, t)\tau + \sqrt{2D_i}\boldsymbol{\eta}(t)\sqrt{\tau}, \tag{1.58}$$

where τ is the time step, D_i is the diffusion constant which is related to the drag coefficient ζ_i through the Einstein relation $D\zeta = k_\mathrm{B}T$, $\boldsymbol{f}_i(\{\boldsymbol{x}\}, t)$ is the sum of forces acting on the particle, and $\boldsymbol{\eta}(t)$ is a Gaussian random variable with zero mean and unit variance. Reactions between particles are implemented through assigning reaction probabilities to interacting particles if their separation $\|\boldsymbol{x}_i - \boldsymbol{x}_j\|$ is less than the sum of their reaction radii. Excluded volume interactions are accounted for in these methods through the force term, which allows for a realistic simulation of molecular crowding.

The other methods sample solutions from the probability distribution described by the

RDME—a version of the CME generalized to include spatial degrees of freedom. These RDME[28–31] methods do not track individual particles, but rather track the populations of chemical species within subvolumes of the simulation domain. Each subvolume is treated as well-stirred reaction volume, allowing for the reactions in each subvolume to be simulated independently. These methods are generally less computationally expensive than particle-based methods, however excluded volume effects between reacting particles are neglected. Molecular crowding due to other molecules in the cell can be modeled through the introduction of obstacles in the lattice geometry[32,33], however. The use of spatial discretization could lead to reduced accuracy compared to particle methods, however it has been shown that RDME methods approach the same level of accuracy when the reaction radii are much smaller than the lattice spacing[34–37].

The RDME is defined as

$$\begin{aligned}\frac{\mathrm{d}P(\boldsymbol{x},t)}{\mathrm{d}t} =& \sum_{\nu}^{V}\sum_{r}^{R}\left[-a_r(\boldsymbol{x}_\nu)P(\boldsymbol{x}_\nu,t) + a_r(\boldsymbol{x}_\nu - \boldsymbol{S}_r)P(\boldsymbol{x}_\nu - \boldsymbol{S}_r,t)\right] \\ &+ \sum_{\nu}^{V}\sum_{\xi}^{\pm\hat{\imath},\hat{\jmath},\hat{k}}\sum_{\alpha}^{N}\left[-d_\nu^\alpha x_\nu^\alpha P(\boldsymbol{x},t) + d_{\nu+\xi}^\alpha (x_{\nu+\xi}^\alpha + 1)P(\boldsymbol{x} + \mathbf{1}_{\nu+\xi}^\alpha - \mathbf{1}_\nu^\alpha, t)\right],\end{aligned} \tag{1.59}$$

where $P(\boldsymbol{x},t)$ is the probability distribution to find a configuration $\boldsymbol{x}$ at time t. The configuration vector $\boldsymbol{x}$ contains the number of species present at each individual lattice site, e.g.

$$\boldsymbol{x} = \left[\boldsymbol{x}_{1,1,1}\ \boldsymbol{x}_{1,1,2}\ \cdots\ \boldsymbol{x}_{1,1,N_\mathrm{z}}\ \boldsymbol{x}_{1,2,1}\ \cdots\ \cdots\ \boldsymbol{x}_{1,N_\mathrm{y}N_\mathrm{z}}\ \cdots\ \cdots\ \cdots\ \boldsymbol{x}_{N_\mathrm{x},N_\mathrm{y},N_\mathrm{z}}\right]^\mathrm{T} \tag{1.60a}$$

$$\boldsymbol{x}_{i,j,k} = \left[x_{i,j,k}^{1}\ \cdots\ x_{i,j,k}^{N_\mathrm{sp}}\right]^\mathrm{T}. \tag{1.60b}$$

The first term in Eq. 1.59 describes the flow of probability between different copy number states at every lattice site. The reaction propensities $a_r(\boldsymbol{x}_\nu)$ give the transition probabilities due to reaction r firing at site ν, and are computed following Eq. 1.27. The r row of the stoichiometry matrix $\boldsymbol{S}$ is the change in species counts when reaction r occurs. The second term describes the flow of probability due to diffusion between neighboring lattice sites, indexed by ξ. For a cubic lattice with spacing λ, the diffusive propensity is

$$d_\nu^\alpha = \frac{D_\nu^\alpha}{\lambda^2}, \tag{1.61}$$

and is computed by treating diffusion as a discrete random walk of step size λ and associating the diffusion constant D_ν^α with the discrete step probability. The notation $\mathbf{1}_\nu^\alpha$ represents a single molecule of species α in volume, i.e. $(\mathbf{1}_\nu^\alpha)_\mu^\beta = \delta_{\alpha\beta}\delta_{\mu\nu}$.

We utilize Lattice Microbes (LM)—a suite of highly parallel graphics processing unit (GPU) accelerated algorithms for stochastic simulations of complex biochemical reaction networks under realistic cellular conditions[28,29,32,33,38,39] developed by the Luthey-Schulten group—for all RDME simulations presented in this dissertation. The software includes implementations of both CME and RDME sampling algorithms, including a unique multiple particle diffusion (MPD-RDME)[32] algorithm, which splits the diffusion operator into x-, y-, and z-components, allowing for more efficient processing on the GPU. LM trajectories are capable of reaching hour long timescales—orders of magnitude longer than competing codes[25–27,30]. By taking short time steps such that active particles are unlikely to take part in multiple reactions, the subvolumes are rendered independent, and can be calculated in parallel (implementation details can be found in Roberts et al.[32], Roberts et al.[28], and Hallock et al.[29]).

The multi-particle diffusion RDME (MPD-RDME) algorithm represents the simulation volume as a cubic lattice, where each site contains a finite number of particles. The particles are represented by an array of integers, where the value of each integer greater than zero identifies the presence of a particle and its species type and a value of zero indicates an vacancy. The simulation loop proceeds by executing the GPU-based procedures (called kernels) for diffusion in the x, y, and z directions sequentially, followed by the reaction kernel. The simulation time is updated $t_{i+i} = t_i + \tau$, and once $t > t_{\text{final}}$ the loop exits and the simulation terminates. These kernels are executed in parallel on the GPU where each thread is responsible for a single site. The simulation algorithm takes regular time steps, as opposed to the Gillespie direct algorithm which takes time steps of varying length, sampled from an exponential distribution. This could lead to inaccurate sampling of the underlying probability distribution, however each lattice site must be synchronized in time to all other sites in order to perform diffusion.

```
Data: Initial particle lattice – $\boldsymbol{x}$, stoichiometric matrix – $\mathsf{S}$, propensity functions – $a_r(\boldsymbol{x}_v)$,
      time step – $\tau$, and maximum evaluation time – $t_\mathrm{f}$.
Result: Final particle lattice – $\boldsymbol{x}$
$t \longleftarrow 0$;
while $t < t_\mathrm{end}$ do
    for $\xi \in \{x, y, z\}$ do
        parfor $v \in V$ do                                    // Execute on GPU
            diffusionKernel($\xi, v$);
    parfor $v \in V$ do                                        // Execute on GPU
        reactionKernel($v$);
    $t \longleftarrow t + \tau$;
```

Algorithm 1.2 The multi-particle diffusion RDME algorithm

```
// Perform a single diffusion time step at the lattice site $v$ in the $\pm\xi$
   direction.
for $\alpha \in \boldsymbol{x}_v$ do
    $\rho \longleftarrow$ uniformRand();
    // Here $v \pm \xi$ refers to the index of the neighboring site in the $\pm\xi$
       direction, and $v \rightarrow v \pm \xi$ indicates the site transition probability
       if the starting and ending site types differ.
    if $\rho \le d^{\alpha}_{v \rightarrow v+\xi}$ then
        moveParticle($\alpha, v, v + \xi$);
    else if $\rho > 1 - d^{\alpha}_{v \rightarrow v-\xi}$ then
        moveParticle($\alpha, v, v - \xi$);
```

Function 1.3 diffusionKernel(ξ, v)

During a time step $[t, t+\tau]$, the probability of a reaction occurring is simply

$$P_\mathrm{react} = \int_0^\tau \mathrm{d}t\, a_\mathrm{total} \mathrm{e}^{-a_\mathrm{total} t} = 1 - \mathrm{e}^{-a_\mathrm{total} \tau} \tag{1.62}$$

following Eq. 1.30. Each time step, a random number $\rho \sim \mathrm{Uniform}(0,1)$ is drawn and if $\rho < P_\mathrm{react}$, then a reaction will occur at that time step, chosen using Eq. 1.31 as in the exact stochastic

```
// Perform a single reaction time step at the lattice site $v$.
for $r \leftarrow 1$ to $N_{\mathrm{rxn}}$ do
    $a_{\mathrm{total}} \longleftarrow a_{\mathrm{total}} + a_r(\boldsymbol{x}_v)$;
$\rho_1 \longleftarrow$ uniformRand();
if $\rho_1 \leq 1 - \mathrm{e}^{-a_{\mathrm{total}} t}$ then
    $\rho_2 \longleftarrow$ uniformRand();
    for $r \leftarrow 1$ to $N_{\mathrm{rxn}}$ do
        if $a_{r-1}(\boldsymbol{x}_v) < \rho_2 \cdot a_{\mathrm{total}} \leq a_r(\boldsymbol{x}_v)$ then
            executeReaction($v, \boldsymbol{S}_r$);
            break;
```

Function 1.4 reactionKernel(v)

simulation algorithm. The diffusion kernels proceed similarly. The probability that the particle leaves its site is

$$P_{\mathrm{react}} = 1 - \mathrm{e}^{-a_{\mathrm{dif}}\tau}, \tag{1.63}$$

where a_{dif} is the sum of the two diffusive propensities to transition along the diffusion kernel axis, e.g. $2d_v^{\alpha}$ in the case where the site types of the sites -1, 0, and +1 are all identical. Algorithm 1.2 summarizes the MPD-RDME algorithm. The details of the lattice modifying functions executeReaction() and moveParticle() are involved. Adding or removing a particle from the lattice must be done such that the array of particles remains compacted, i.e. ensuring that all vacancies are after the particles. Moving a particle requires communication between threads, since each thread is responsible for a single lattice site. A description of how this is achieved will require a digression into how GPU hardware is exposed to the programmer.

When a kernel is launched on the GPU, many instances of the code will be executing at once. These instances are grouped into thread blocks, where each thread in a block has access to small amount (NVIDIA TITAN X: 48 kB) of low latency memory shared among all threads in the block, referred to as shared memory. Reading and writing to this memory can be 100× faster than the main GPU memory, referred to as global memory (NVIDIA TITAN X: 12 GB), the large pool of memory where the full lattice data structure is stored. Sharing the list of particle movements must be done through shared memory in order to maximize performance. However due to the small size of the available shared memory, the thread blocks must be significantly smaller than the full lattice. When a block of the lattice is loaded into shared memory, so is its “apron”. The apron includes all lattice sites bordering the lattice block. These sites are not modified, however the movement of their particles is recorded so that diffusion from apron sites into the lattice block is accounted for. Since a particle can move at most one lattice site per time step, an apron of a single lattice site is sufficient. The outcome of diffusion will be computed multiple times in these apron sites from other thread blocks, so the outcome must be identical each time. The seed of the random number generator is computed from the site index, so that two different threads will compute the same random diffusion events for the same site. The main benefit of splitting the diffusion operator into orthogonal axes is that it decreases the size of the apron dramatically, reducing the number of unproductive calculations.

The nature of the MPD-RDME algorithm places constraints on the model parameters and the coarseness of the lattice. The largest diffusion constant in the system and the lattice spacing dictates the largest valid time step,

$$\tau < \frac{\lambda^2}{2\max_\alpha D_\alpha}, \tag{1.64}$$

that can be taken. This relationship is a consequence of the fact that diffusion in the RDME is a discrete random walk. The decoupling of reactions from diffusion used in this method relies on a separation between diffusive and reaction timescales. We define the diffusive timescale to be

$$\tau_\mathrm{D} = \frac{\lambda^2}{6D_\mathrm{max}} \tag{1.65}$$

and the reaction timescale to be

$$\tau_\mathrm{R} = \frac{1}{a_\mathrm{max}}, \tag{1.66}$$

where a_max is the largest reaction propensity. Then

$$\tau_\mathrm{R} \gg \tau_\mathrm{D} \tag{1.67}$$

implies that

$$\lambda \ll \sqrt{\frac{6D_{\max}}{a_{\max}}}. \tag{1.68}$$

Substituting in the expressions for reaction propensities (Eq. 1.27), we see that there are upper and lower bounds on the lattice size:

$$\lambda \ll \left(\frac{6D_{\max}}{\Gamma^{(0)}_{\max} N_{\mathrm{A}}}\right)^{1/5} \qquad \text{(zeroth-order)} \tag{1.69}$$

$$\lambda \ll \sqrt{\frac{6D_{\max}}{\Gamma^{(1)}_{\max}}} \qquad \text{(first-order)} \tag{1.70}$$

$$\lambda \gg \frac{\Gamma^{(2)}_{\max}}{6D_{\max} N_{\mathrm{A}}} \qquad \text{(second-order),} \tag{1.71}$$

where $\Gamma^{(i)}_{\max}$ is the maximum i^{th}-order flux (Eq. 1.15) evaluated using typical lattice site concentrations.

In the implementation of the MPD-RDME algorithm used by LM, the simulation volume is represented by an $N_{\mathrm{x}} \times N_{\mathrm{y}} \times N_{\mathrm{z}} \times N_{\mathrm{p}}$ array of integers, where $N_{\mathrm{x,y,z}}$ are the number of lattice sites in each dimension and N_{p} is the lattice occupancy. The finite lattice occupancy is a consequence of the GPU oriented design of LM, and allows the GPU to access the lattice memory in a regular pattern. This implies that the maximum lattice size available to the modeler is constrained by the total concentration of particles in the system. Currently, LM allows for either 8 or 16 particles per site. If a reaction or diffusion event causes any subvolume to exceed its capacity, the computation on the GPU must be placed on hold so that a process can run on the host to correct the overflow. Particles in the offending site are redistributed among the neighboring subvolumes which have empty particle slots available. Frequent overflows will cause the computational efficiency to plummet due to the repeated shuffling of the lattice data between the host and GPU.

The probability of an overflow occurring due to diffusion can be computed by considering particle placement as a series of Bernoulli trials[32]. Consider an empty lattice containing L_{s} subvolumes, each having a maximum occupancy of $n_{\max}$, to which we add N particles. The trial in this case is whether or not a particle is placed randomly at a particular lattice site. If all lattice sites are equally likely to receive a particle, then the probably of the success of a single trial is $p = 1/L_\mathrm{s}$. The probability that a particular subvolume receives n particles then follows the binomial distribution

$$P(n) = \binom{N}{n}\left(\frac{1}{L_\mathrm{s}}\right)^{n}\left(1-\frac{1}{L_\mathrm{s}}\right)^{N-n}. \tag{1.72}$$

The overflow probability of a single site is then $1 - \sum_{1\le n\le n_\mathrm{max}} P(n)$, from which it follows that the expected number of overflows, N_of, is

$$\mathbb{E}[N_\mathrm{of}|n_\mathrm{max}, L_\mathrm{s}, N] = L_\mathrm{s}\left[1-\sum_{1\le n\le n_\mathrm{max}}\binom{N}{n}\left(\frac{1}{L_\mathrm{s}}\right)^{n}\left(1-\frac{1}{L_\mathrm{s}}\right)^{N-n}\right]. \tag{1.73}$$

An acceptable number of time steps between overflows should be 100 – 1000. Eq. 1.73 can be solved numerically to find the appropriate lattice occupancy for a required particle density.

Chapter 2

DNA looping increases the range of bistability in a stochastic model of the *lac* genetic switch[*]

2.1 Introduction

The *lac* circuit in *Escherichia coli* is one of the most well-studied examples of gene regulation, dating back to the classic experiments by Novick and Weiner[41] in 1957 and Jacob and Monod[42] in 1961. Since then much has been learned about the system, including the effect of DNA looping on the effectiveness of the switch[43–46]. The *lac* operon controls the translation of genes necessary for the utilization of lactose. Three gene products are translated from the operon: β-galactosidase (LacZ) is responsible for cleaving the β-1,4 glycosidic bond in lactose to yield glucose and galactose, β-galactoside transacetylase transfers an acetyl group to aid in lactose metabolism, and β-galactoside permease (LacY) is a membrane bound transporter protein, which actively imports lactose from the environment into the cell. A fourth constitutive protein, *lac* repressor (LacI), is coded for upstream of the *lac* promoter. This protein binds to the *lac* operators

[*] Chapter 2 contains material reproduced from Earnest TM, Roberts E, Assaf M, Dahmen K, and Luthey-Schulten Z (2013). DNA looping increases the range of bistability in a stochastic model of the lac genetic switch. *Phys. Biol.*, **10**(2), p. 026002. doi:10.1088/1478-3975/10/2/026002. © IOP Publishing. Reproduced with permission. All rights reserved.

to inhibit transcription. When lactose enters the cell, some of it is converted to the *lac* inducer, allolactose. This binds to LacI, decreasing its binding affinity for the *lac* operators, which allow for more *lac* proteins to be translated. The *lac* circuit responds to glucose concentration by only producing the *lac* proteins when glucose is unavailable. During glucose starvation, cyclic adenosine monophosphate is produced which binds to the catabolite activator protein (CAP), allowing CAP to bind upstream of the *lac* promoter and recruit RNA polymerase (RNAP) to start transcription. Thus the *lac* switch acts as an AND gate to the signals "low glucose" and "high lactose", only switching on when both are true. We are primarily concerned with the switch's response to the lactose signal.

Since producing more LacY leads to higher intracellular inducer concentrations, a positive feedback loop is set up allowing the cell to switch between two phenotypes: uninduced (LO), which produces a basal level of *lac* proteins and induced (HI), where the *lac* proteins are produced at their maximum rate. By increasing the extracellular inducer concentration the population of cells switch between uninduced, to a heterogeneous mixture of induced and uninduced, to all induced. Cells do not persist in intermediate states. These heterogeneous populations could enjoy a fitness advantage since some fraction of cells will always be prepared for changes in environmental conditions[47–49]. Although the *lac* system does not appear to be bistable except when a gratuitous inducer such as thiomethyl-β-D-galactoside (TMG) is used[50], this system is useful for studying the general phenomena of genetic switches. In the presence of minimal external glucose, LacI controls the production of messenger RNA (mRNA). The *lac* repressor binds to the main operator (O_1) to prevent transcription. The binding affinity of LacI to the operator is controlled by the number of inducer molecules bound to the repressor. The *lac* repressor is a homo-tetramer, which can bind to two operators simultaneously: each dimer can bind to an operator individually. Each monomer can bind one inducer molecule for a maximum of four inducer molecules bound to the repressor.

The DNA sequence near the *lac* operon contains two auxiliary operators, which allow the local structure of the DNA to assume a looped conformation. DNA loops are ubiquitous in all domains of life as a regulatory tool, including transcriptional regulation in prokaryotes, enhancer sequences in eukaryotes, the lysis/lysogeny switch in phage λ, site-specific recombination, and DNA replication[51]. These loops occur when a protein or complex binds to two different sites along the DNA molecule, which could be separated by tens to thousands of base pairs. The regulatory effect can be understood as increasing the effective concentration of transcription factor near the binding site since the protein cannot diffuse away from the DNA unless both binding sites are dissociated[52]. All three operators O_1, O_2, and O_3 are involved in the formation of DNA loops. Binding to either of the auxiliary operators, O_2 or O_3 only, is not sufficient to suppress transcription. However if both auxiliary operators are removed, the repression level is reduced by a factor of 100[46]. Thus DNA loops appear to increase the ability of LacI to repress transcription.

A molecular mechanism for the induction of the *lac* operon both in the presence and absence of DNA loops was presented by Choi et al.[53] [43,53]. In their study mutants were designed such that LacY was labeled with yellow fluorescent protein (YFP). Using the non-metabolizable inducer TMG, they were able to observe bistability in genetically identical populations of *E. coli* at concentrations of 40–70 μM of extracellular TMG. However using a population of mutants with both auxiliary operators removed, all cells deterministically switched into the induced state for concentrations of inducer as low as 20 μM. They argued that LacI must dissociate from both operators for induction to occur and that basal levels of LacY are due to partial dissociation of the DNA loop, leaving the operator downstream of the *lac* promoter free, allowing single transcripts to be produced infrequently before the loop reforms.

The *lac* switch has been a frequent subject of mathematical modeling. There have been many attempts to describe the system deterministically using chemical rate equations[50,54–58] with varying levels of complexity. These models are successful in capturing much of the experimental behavior of the switch, such as identifying the number of phenotypes and the mean species numbers, however they cannot capture the full range of behavior of the *lac* system since they neglect the stochastic and discrete nature of chemical reactions. Indeed, it was shown in Stamatakis and Mantzaris[54] that changing certain ratios of parameters that have no effect on a deterministic model, can have a strong effect on the corresponding stochastic model. It was also shown in Vilar et al.[56] that the ranges of bistability predicted by deterministic modeling are much larger than what is realizable in a stochastic model. Deterministic models are also unable to compute switching times between states of induction since switching is a stochastic phenomenon. Stochastic models of *lac* have been developed to take this deficiency into account[53–55]. It is also possible to model genetic switches by not invoking a microscopic model, and instead focusing on the experimental copy number trajectories using the MaxCal method[59,60]. This technique can provide a description of the intrinsic noise arising from small copy numbers[61–63]. Theory developed in general for stochastic gene expression[64–71] can also provide a reasonable description of the system. However, none to our knowledge attempt to include the effect of DNA loops on the range of bistability and switching rates.

Here we present a stochastic treatment of a gene–mRNA–protein model of the *lac* operon in *E. coli* interacting with extracellular inducer that includes transitions to looped DNA states. We develop a novel combination of the finite state projection (FSP)[23] and geometric burst approximation (see below) to render the CME computationally tractable and show that rare random loop dissociation events are responsible for the LO → HI phenotypic transition. Our results show that the process of induction in the *lac* operon is preceded by the total dissociation of the DNA loop. We show that the microscopic mechanism of switching in the three-state model is fundamentally different than the mechanism in a model without DNA looping. The looped state alters the switching dynamics such that fast switching times between the metastable states are possible while minimizing noise within those states. A model without the looped state shows a minimal range of bistability below 20 μM of external inducer concentration[33], whereas the three-state model shows the full range of bistability as observed in experiments.

2.2 Model

In this work, we consider two models for gene expression from the *lac* operon. The first model is the standard two-state model of gene expression in which the two states represent the DNA's transcriptional state, either repressed or active. The second model adds an additional third state to account for the possibility of a potentially long-lived looped state with different transcriptional properties. Figure 2.1 shows a schematic representation of the two models. All parameters are defined and values given in Table 2.1.

2.2.1 Two-state model — no DNA looping

The two-state model contains two transcriptional states for the DNA, *Off* and *On* where the operon is transcriptionally inactive and active, respectively. DNA looping in not possible in this two-state model. When the operon is in the *On* state, transcription proceeds as a first order reaction:

$$On \xrightarrow{k_{\text{ts}}} On + m. \tag{2.1a}$$

Protein (Y) is translated at a rate that is proportional to the mRNA (m) copy number

$$m \xrightarrow{k_{\text{tl}}} m + Y \tag{2.1b}$$

and protein and mRNA degrade at a rate proportional to their respective abundances

$$m \xrightarrow{k_{\text{degm}}} \varnothing, \tag{2.1c}$$

$$Y \xrightarrow{k_{\text{degp}}} \varnothing. \tag{2.1d}$$

Due to interactions between the *lac* repressor and inducer molecules, switching between the active and inactive transcriptional states occurs at a rate dependent on the inducer concentration inside the cell,

$$On \underset{k_{\text{fn}}([\text{I}])}{\overset{k_{\text{nf}}([\text{I}])}{\rightleftharpoons}} Off. \tag{2.1e}$$

The switching rate functions $k_{\text{fn}}([\text{I}])$ and $k_{\text{nf}}([\text{I}])$ are determined by the microscopic interactions between the repressor and inducer. It was previously shown[33] that a particular microscopic model for these interactions gave rise to switching rates that were first-order for a given constant inducer concentration and with functional dependencies on inducer concentration that could be well-described by Hill-like functions. We fit the simulation data from Roberts et al.[33] (see

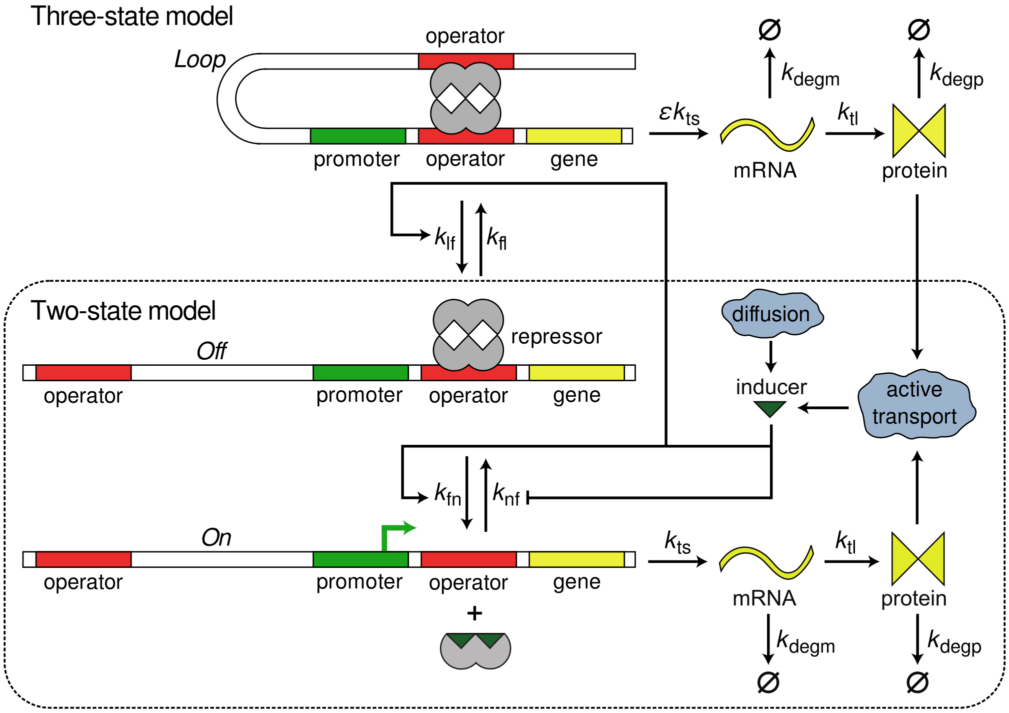


Figure 2.1 Cartoon schematic of the three-state model of the *lac* circuit. The two-state model is exactly the same except without the *Loop* state. There are three states controlling the transcriptional state of the switch: *On* producing transcripts at the nominal rate with O_1 free, *Off* where no transcription is possible due to O_1 being bound to LacI, and *Loop* which is a coarse-grained state representing any of the possible looped states and singly bound DNA/Repressor states which do not inhibit transcription. This state models the proposed phenomenon of transcriptional leakage in which fluctuations of the repressor/DNA complex can allow rare transcription events to occur while in an otherwise looped conformation[43]. The switch can be induced by adding external inducer to the environment. The inducer is then transported into the cell via diffusion and active transport by lactose permease where it binds to the *lac* repressor, decreasing its binding affinity for the *lac* operators. This decreased affinity allows the *lac* repressor to fall off and allow RNA polymerase to bind and transcribe mRNA. The system now transitions from *Loop* through *Off* to *On* where transcripts are produced at a higher rate, setting up a positive feedback loop where more transporter protein is translated.

Appendix A) to the functions

$$k_{\text{fn}}([\text{I}]) = k_{\text{fn}}^0 + (k_{\text{fn}}^1 - k_{\text{fn}}^0)\frac{[\text{I}]^{H_{\text{fn}}}}{I_{\text{fn}}{}^{H_{\text{fn}}} + [\text{I}]^{H_{\text{fn}}}} \tag{2.2}$$

and

$$k_{\text{nf}}([\text{I}]) = k_{\text{nf}}^0 + (k_{\text{nf}}^1 - k_{\text{nf}}^0)\frac{[\text{I}]^{H_{\text{nf}}}}{I_{\text{nf}}{}^{H_{\text{nf}}} + [\text{I}]^{H_{\text{nf}}}}. \tag{2.3}$$

and obtain Hill-like expressions for the rate functions in our model in terms of internal inducer concentration. Here the parameters k_x^0 and k_x^1 determine the minimum and maximum transition rates, I_x is the internal inducer concentration that the transition rate is $\frac{1}{2}(k_x^0 + k_x^1)$, and H_x is the Hill coefficient, which is proportional to the slope at $[I] = I_x$.

However, including the inducer as a separate species would dramatically increase the complexity of the model. Instead we express the transition rates in terms of the number of LacY proteins in the cell membrane,

$$On \underset{k_{\text{fn}}(Y)}{\overset{k_{\text{nf}}(Y)}{\rightleftharpoons}} Off. \tag{2.4a}$$

Consider that inducer (I) can enter the cell either through passive diffusion from the environment (with constant concentration $[\text{I}]_{ex}$),

$$I_{\text{ex}} \underset{k_{\text{id}}}{\overset{k_{\text{id}}}{\rightleftharpoons}} I, \tag{2.4b}$$

or through active transport by the LacY protein (modeled using irreversible Michaelis-Menten kinetics),

$$Y + I_{\text{ex}} \underset{k_{\text{yioff}}}{\overset{k_{\text{yion}}}{\rightleftharpoons}} YI \xrightarrow{k_{\text{it}}} Y + I. \tag{2.4c}$$

If we assume that inducer responds very quickly to a change in the LacY concentration, we can solve for the steady-state concentration of internal inducer as a function of the LacY and external inducer concentrations:

$$[\text{I}]([\text{Y}], [\text{I}]_{\text{ex}}) = [\text{I}]_{ex}\left(1 + \frac{k_{\text{it}}}{k_{\text{id}}} \cdot \frac{[\text{Y}]}{[\text{I}]_{ex} + K_M}\right) \tag{2.5}$$

with

$$K_M = \frac{k_{\text{yioff}} + k_{\text{it}}}{k_{\text{yion}}}. \tag{2.6}$$

Substituting Equation Eq. 2.5 into Equations Eq. 2.2 and Eq. 2.3 we obtain expressions for $k_{\text{fn}}(Y)$ and $k_{\text{nf}}(Y)$ with a little algebra.

2.2.2 Three-state model — with DNA looping

Since the *lac* operon has three operators to which LacI can bind, the transcriptional states are more complicated than *On* and *Off* alone. The simplest modification to the two-state model to handle this complexity is to add a third state—*Loop*—that describes the operon when it is in a looped conformation with the repressor. It was hypothesized that in a looped conformation the LacI–O_1 binding can fluctuate allowing RNAP to bind and transcribe while O_1 is unoccupied with LacI nearby[43]. We introduce the *Loop* state to model this effect. This third state should not be interpreted as a specific conformation of the operon/repressor complex. Instead it should be understood as a coarse-grained state that represents all DNA/repressor looped conformations (O_1-O_2 and O_1-O_3) as well as bound conformations in which transcription is not repressed (O_2, O_3, or both bound.) This state allows for the slow leakage of transcripts. We model this leakage by allowing transcription from the *Loop* state at a rate ϵ times the normal *On* transcription rate by adding the reactions

$$\mathit{Off} \underset{k_{\text{lf}}([\text{I}])}{\overset{k_{\text{fl}}}{\rightleftharpoons}} \mathit{Loop} \tag{2.7a}$$

and

$$\mathit{Loop} \xrightarrow{\epsilon k_{\text{ts}}} \mathit{Loop} + m \tag{2.7b}$$

to our model. The leakage factor ϵ is conditional probability to have repressor bound to O_2, O_3, or both O_2 and O_3 while in the *Loop* state. The schematic representation of the two models are shown in Figure 2.1. This three-state model is similar to another three-state promoter model, which has been investigated recently[72]. This work focused on the stochastic mutual repressor model and described the state of the operators using three transcriptional states. Their methods would be difficult to implement for our models due to the noise in the translation rate arising from fluctuating mRNA abundance.

We estimated ϵ from Choi et al.[43]. They measured the distribution of YFP counts in a mutant with the *lac* circuit modified to disable positive feedback while allowing DNA loops to form. This experiment was conducted by replacing LacY with a Tsr-YFP fusion protein, which also localizes in the cell membrane, but cannot import allolactose into the cell. These distributions were well-described by a gamma distribution parameterized by the burst rate a and the burst size b. This approximate result was derived from a gene–mRNA–protein model with one transcriptional state[66]. The burst rate $a \equiv k_{\mathrm{ts}}/k_{\mathrm{degp}}$ can be interpreted as the number of transcripts produced per cell cycle and the burst size $b \equiv k_{\mathrm{tl}}/k_{\mathrm{degm}}$ refers to the average number of protein transcribed from a single mRNA. The burst rate and the burst size were measured for TMG concentrations between 0 and 200 μM and found to be relatively constant over this range: $0.34 < a < 1.25$ and $1.29 < b < 2.59$. We used this measurement to estimate that $5.7 \times 10^{-4} < \epsilon < 2.1 \times 10^{-3}$ and assumed $\epsilon = 8.3 \times 10^{-4}$ for our model. The two transcription rates k_{ts} and ϵk_{ts} are responsible for the different burst sizes observed in Choi et al.[43]. "Small" bursts of transcription occur due to leakage whereas "large" bursts occur due to full dissociation of the complex (*Loop* → *Off* → *On*.)

Since the *On* ↔ *Off* switching rate functions were well-described by a Hill function, we assume that the *Loop* to *Off* rate function could also take that form. Thus

$$k_{\mathrm{lf}}([\mathrm{I}]) = k_{\mathrm{lf}}^{0} + (k_{\mathrm{lf}}^{1} - k_{\mathrm{lf}}^{0}) \frac{[\mathrm{I}]^{H_{\mathrm{lf}}}}{{I_{\mathrm{lf}}}^{H_{\mathrm{lf}}} + [\mathrm{I}]^{H_{\mathrm{lf}}}}. \tag{2.8}$$

We will assume that the *Off* to *Loop* rate is constant in inducer concentration, since transitions into the looped state must occur via thermal fluctuations that take the singularly bound inducer/DNA complex into a conformation in which the unoccupied binding site on the *lac* repressor can reach a free operator. It is not possible to determine these switching rates directly from the data available from the literature, so we search the parameter space to determine for which values of these parameters the system exhibits the desired response. Of the parameters presented in Table 2.1, only the five parameters describing the constant *Off* → *Loop* rate and the *Loop* → *Off*

Hill function Eq. 2.8 are predicted. The remaining parameters are taken from the literature[33,43,54].

2.2.3 Deterministic and stochastic representation

The deterministic rate equations for mRNA and LacY abundance are

$$\frac{\mathrm{d}x}{\mathrm{d}t} = k_{\mathrm{ts}}\mathscr{F}(y) - k_{\mathrm{degm}}x \tag{2.9a}$$

$$\frac{\mathrm{d}y}{\mathrm{d}t} = k_{\mathrm{tl}}x - k_{\mathrm{degp}}y \tag{2.9b}$$

where x and y are (continuous) concentration variables for mRNA and LacY respectively. Here we have defined the total transcription probability,

$$\mathscr{F}(y) = \frac{P_{\mathrm{on}}(y) + \epsilon P_{\mathrm{loop}}(y)}{P_{\mathrm{on}}(y) + P_{\mathrm{off}}(y) + P_{\mathrm{loop}}(y)}, \tag{2.10}$$

where the $P_z(y)$ functions are the probability to find the system in the transcriptional state z at fixed protein abundance y and are computed simply from considering the steady state of a Markov process consisting of the three transcriptional states alone. This process is represented by

$$\mathit{On} \underset{k_{\mathrm{fn}}(y)}{\overset{k_{\mathrm{nf}}(y)}{\rightleftharpoons}} \mathit{Off} \underset{k_{\mathrm{lf}}(y)}{\overset{k_{\mathrm{fl}}}{\rightleftharpoons}} \mathit{Loop}, \tag{2.11}$$

which at steady state is governed by the master equation

$$\mathbf{A}\cdot\mathbf{P} = \begin{pmatrix} -k_{\mathrm{nf}}(y) & k_{\mathrm{fn}}(y) & 0 \\ k_{\mathrm{nf}}(y) & -(k_{\mathrm{fn}}(y) + k_{\mathrm{fl}}) & k_{\mathrm{lf}}(y) \\ 0 & k_{\mathrm{fl}} & -k_{\mathrm{lf}}(y) \end{pmatrix} \begin{pmatrix} P_{\mathrm{on}} \\ P_{\mathrm{off}} \\ P_{\mathrm{loop}} \end{pmatrix} = 0, \tag{2.12}$$

whose normalized solution is

$$P_{\mathrm{on}} = \mathscr{C}^{-1} k_{\mathrm{fn}}(y) k_{\mathrm{lf}}(y) \tag{2.13a}$$

$$P_{\mathrm{off}} = \mathscr{C}^{-1} k_{\mathrm{nf}}(y) k_{\mathrm{lf}}(y) \tag{2.13b}$$

$$P_{\mathrm{loop}} = \mathscr{C}^{-1} k_{\mathrm{fl}} k_{\mathrm{nf}}(y) \tag{2.13c}$$

with $\mathscr{C} = k_{\mathrm{fn}}(y)k_{\mathrm{lf}}(y) + k_{\mathrm{nf}}(y)\big(k_{\mathrm{fl}} + k_{\mathrm{lf}}(y)\big)$, and y is computed from [I] by Eq. 2.5.

To investigate the fixed points of equations Eqs. 2.9a–2.9b, we take the time derivatives to be zero and solve for y to get

$$\frac{\mathrm{d}y}{\mathrm{d}t} = k_{\mathrm{degp}}[\mathscr{N}\mathscr{F}(y) - y] = 0 \tag{2.14}$$

where

$$\mathscr{N} = \frac{k_{\mathrm{ts}} k_{\mathrm{tl}}}{k_{\mathrm{degm}} k_{\mathrm{degp}}} \tag{2.15}$$

is approximately the mean population of the induced state. For a set of parameters exhibiting bistability over some range of $[\mathrm{I}]_{\mathrm{ex}}$, the form of $\mathscr{F}(y)$ allows the dynamical system to exist in three distinct phases. For low values of $[\mathrm{I}]_{\mathrm{ex}}$, the system has a single stable fixed point at n_{LO} corresponding to the uninduced phenotype. For high values of $[\mathrm{I}]_{\mathrm{ex}}$ the system has a single fixed point at n_{HI} corresponding to the induced phenotype. For intermediate values of $[\mathrm{I}]_{\mathrm{ex}}$, there is a range in which the dynamical system has three fixed points: two stable fixed points n_{LO} and n_{HI} and an unstable fixed point n_0. This external inducer regime corresponds to a heterogeneous population containing both phenotypes. The locations of these fixed points are not fixed, but can change with the inducer concentration. For the transcription state transition functions considered here, Eq. 2.14 is generally not solvable analytically due to the form of transcriptional state rate functions used.

The probability to be in a particular transcriptional state at a specific mRNA and protein abundance can be determined by solving the CME for the system[73]. These equations govern the time evolution of the probability, P^s_{mn}, to find m mRNA molecules and n proteins, when being in transcriptional state s. Writing out the transcriptional states explicitly we have

$$\begin{aligned}
\frac{\mathrm{d}P_{mn}^{\mathrm{on}}}{\mathrm{d}t} =& k_{\mathrm{fn}}(n)P_{mn}^{\mathrm{off}} - k_{\mathrm{nf}}(n)P_{mn}^{\mathrm{on}} \\
&+ \left[k_{\mathrm{ts}}(\mathsf{E}_M^{-1}-1) + k_{\mathrm{degm}}(\mathsf{E}_M^{+1}-1)m + k_{\mathrm{tl}}(\mathsf{E}_N^{-1}-1)m - k_{\mathrm{degp}}(\mathsf{E}_N^{+1}-1)n\right] P_{mn}^{\mathrm{on}} && \text{(2.16a)} \\
\frac{\mathrm{d}P_{mn}^{\mathrm{off}}}{\mathrm{d}t} =& k_{\mathrm{nf}}(n)P_{mn}^{\mathrm{on}} + k_{\mathrm{lf}}(n)P_{mn}^{\mathrm{loop}} - (k_{\mathrm{fn}}(n) + k_{\mathrm{fl}})P_{mn}^{\mathrm{off}} \\
&+ \left[k_{\mathrm{degm}}(\mathsf{E}_M^{+1}-1)m + k_{\mathrm{tl}}(\mathsf{E}_N^{-1}-1)m - k_{\mathrm{degp}}(\mathsf{E}_N^{+1}-1)n\right] P_{mn}^{\mathrm{off}} && \text{(2.16b)} \\
\frac{\mathrm{d}P_{mn}^{\mathrm{loop}}}{\mathrm{d}t} =& k_{\mathrm{fl}}P_{mn}^{\mathrm{off}} - k_{\mathrm{lf}}(n)P_{mn}^{\mathrm{loop}} \\
&+ \left[\epsilon k_{\mathrm{ts}}(\mathsf{E}_M^{-1}-1) + k_{\mathrm{degm}}(\mathsf{E}_M^{+1}-1)m + k_{\mathrm{tl}}(\mathsf{E}_N^{-1}-1)m - k_{\mathrm{degp}}(\mathsf{E}_N^{+1}-1)n\right] P_{mn}^{\mathrm{loop}} && \text{(2.16c)}
\end{aligned}$$

where $\mathsf{E}_{M,N}^{\pm i}$ are step operators that increment/decrement the mRNA index (M) or protein index (N) by i. The rates k_{tl}, k_{degm}, and k_{ts},k_{degp} are the mRNA birth and death rates and protein birth and death rates respectively. The functions $k_{\mathrm{nf}}(n)$, $k_{\mathrm{fn}}(n)$, k_{fl}, $k_{\mathrm{lf}}(n)$, are the transcriptional state switching rates for *On*→*Off*, *Off*→*On*, *Off*→*Loop*, and *Loop*→*Off*.

2.3 Methods

The CME Eqs. 2.16a–2.16c, being two-dimensional, are unsolvable analytically. An approximate solution was derived[65] for the two-state system without feedback in the limit

$$\gamma \equiv \frac{k_{\mathrm{degm}}}{k_{\mathrm{degp}}} \to \infty, \tag{2.17}$$

however this approximation is infeasible here since they used a generating function technique which is not applicable when the feedback functions are non-polynomial. A solution in this same limit is also possible for the two-state system with arbitrary feedback functions. This solution was derived using the Wentzel–Kramers–Brillouin (WKB) approximation[74] to the CME, which allowed for the accurate calculation of the mean switching times between the metastable states[64]. We attempted a similar solution to our three-state problem by taking the CME Eqs. 2.16a–2.16c in the quasi-stationary approximation and applying the WKB ansatz $P(x,y) \sim \mathrm{e}^{-\mathcal{N}S(x,y)}$.

Table 2.1 Selected parameter set from mean of parameter search.

Symbol	Description	Value	Units
Selected from parameter search			
k_{lf}^{0}	*Loop*→*Off* rate at zero inducer	1.45×10^{-4}	s^{-1}
k_{lf}^{1}	*Loop*→*Off* rate at saturating inducer	3.31×10^{-3}	s^{-1}
H_{lf}	Hill coefficient for *Loop*→*Off*	3.12	
I_{lf}	Hill concentration for *Loop*→*Off*	3940.	μM
k_{fl}	Constant *Off*→*Loop* rate	3.83×10^{-3}	s^{-1}
Fixed parameters			
k_{nf}^{0}	*On*→*Off* rate at zero inducer	5.04×10^{-2}	s^{-1}
k_{nf}^{1}	*On*→*Off* rate at saturating inducer	5.04×10^{-4}	s^{-1}
H_{nf}	Hill coefficient for *On*→*Off*	1.	
I_{nf}	Hill concentration for *On*→*Off*	17.4	μM
k_{fn}^{0}	*Off*→*On* rate at zero inducer	6.30×10^{-4}	s^{-1}
k_{fn}^{1}	*Off*→*On* rate at saturating inducer	3.15×10^{-1}	s^{-1}
H_{fn}	Hill coefficient for *Off*→*On*	1.67	
I_{fn}	Hill concentration for *Off*→*On*	5680.	μM
k_{ts}	Transcription rate	1.26×10^{-1}	s^{-1}
k_{tl}	Translation rate	4.43×10^{-2}	s^{-1}
ϵ	Leakage factor[a]	8.33×10^{-4}	
k_{degm}	mRNA degradation rate	1.11×10^{-2}	s^{-1}
k_{degp}	LacY degradation rate	2.10×10^{-4}	s^{-1}
k_{it}	Turnover number for active transport	1.20×10^{1}	s^{-1}
k_{id}	Rate of LacY diffusion across membrane	2.33×10^{-3}	s^{-1}
K_{m}	Michaelis constant for active transport	400.	μM
V_{cell}	Volume of cell	8.00×10^{-16}	L

[a] Estimated from Choi et al.[43].

The fixed parameters were taken from Roberts et al.[33] or extracted from fits to data therein.

This substitution yields a stationary Hamilton-Jacobi equation to leading order in $1/\mathcal{N}$ with hamiltonian

$$\mathcal{H}(x,y,p_x,p_y) = \mathcal{O}_{\mathrm{off}}\mathcal{O}_{\mathrm{on}} + \mathcal{O}_{\mathrm{off}}\mathcal{O}_{\mathrm{loop}} - \mathcal{O}_{\mathrm{off}}\mathcal{O}_{\mathrm{on}}\mathcal{O}_{\mathrm{off}}\mathcal{O}_{\mathrm{loop}}. \tag{2.18}$$

Here the species numbers are scaled by the system size Eq. 2.15, $x = m/\mathcal{N}$, $y = n/\mathcal{N}$, p_i is the momentum conjugate to coordinate i ($p_i = \partial_i S$) and

$$\begin{aligned}
\mathcal{O}_{\mathrm{on}} = {} & \frac{1}{k_{\mathrm{nf}}(y)k_{\mathrm{fn}}(y)}\big[k_{\mathrm{ts}}(\mathrm{e}^{p_x}-1) + k_{\mathrm{degm}}(\mathrm{e}^{-p_x}-1)x \\
& + k_{\mathrm{tl}}(\mathrm{e}^{p_y}-1)x - (\mathrm{e}^{-p_y}-1)y\big] - \frac{1}{k_{\mathrm{fn}}(y)} && \text{(2.19a)}\\
\mathcal{O}_{\mathrm{off}} = {} & k_{\mathrm{degm}}(\mathrm{e}^{-p_x}-1)x + k_{\mathrm{tl}}(\mathrm{e}^{p_y}-1)x - (\mathrm{e}^{-p_y}-1)y - k_{\mathrm{fn}}(y) - k_{\mathrm{fl}} && \text{(2.19b)}\\
\mathcal{O}_{\mathrm{loop}} = {} & \frac{1}{k_{\mathrm{lf}}(y)k_{\mathrm{fl}}}\big[k_{\mathrm{ts}}\epsilon(\mathrm{e}^{p_x}-1) + k_{\mathrm{degm}}(\mathrm{e}^{-p_x}-1)x \\
& + k_{\mathrm{tl}}(\mathrm{e}^{p_y}-1)x - (\mathrm{e}^{-p_y}-1)y\big] - \frac{1}{k_{\mathrm{fl}}} && \text{(2.19c)}
\end{aligned}$$

in units where $k_{\mathrm{degp}} \equiv 1$. However, due to the fast switching times between the metastable states (see below) the WKB method is inapplicable here since it requires that the escape time from the metastable state be exponentially large in the typical system's size[75–77].

Direct numerical solutions will be computationally intensive and not feasible for optimization due to the dimensionality of the resulting system of ODEs. Reasonable values of maximum protein number and maximum mRNA number lead to a system of 450 000 equations. In order to make progress we must employ some method of approximation.

2.3.1 FSP solution of CME using the geometric burst approximation for mRNA

In bacteria the ratio γ is typically large. We can use this fact to eliminate the mRNA degree of freedom from the CME. It is well-known that it is not sufficient to replace the mRNA dynamics with the average mRNA abundance while studying the switching times[64,78,79]. Transcriptional noise will lead to increased noise in protein abundance, which will affect the switching rates between induced and uninduced phenotypes. Another option would be to apply the quasi-steady-state approximation to the CME[80], however due to the stochastic switching of the transcriptional states, the transcription rate is not constant and thus the mRNA distribution is not at steady-state. We need to apply a different tack.

To remove the mRNA degree of freedom while still accounting for transcriptional noise we will exploit the fact that the number of protein molecules translated from a single mRNA is distributed geometrically. In the limit $\gamma \to \infty$ an effective CME can be written without mRNA dependence by assuming that translation occurs in bursts with sizes that are geometrically distributed[81]. Figure 2.2b describes this pictorially. Since the probability that a single mRNA molecule is translated n times before decaying is

$$P_{\mathrm{tl.\,dec.}}(n) = \left(\frac{k_{\mathrm{tl}}}{k_{\mathrm{tl}} + k_{\mathrm{degm}}}\right)^{n} \left(1 - \frac{k_{\mathrm{tl}}}{k_{\mathrm{tl}} + k_{\mathrm{degm}}}\right) = P_{\mathrm{tl}}^{n}(1 - P_{\mathrm{tl}}), \tag{2.20}$$

the translation terms in Eqs. 2.16a–2.16c can be rewritten as

$$k_{\mathrm{tl}}(\mathsf{E}_N^{-1} - 1)m \to k_{\mathrm{ts}} \left[(1 - P_{\mathrm{tl}}) \sum_{r=0}^{n} \left(P_{\mathrm{tl}}\mathsf{E}_N^{-1}\right)^{r} - 1\right], \tag{2.21}$$

and we can drop the transcription terms, mRNA degradation terms, and mRNA indices. For example, consider the one-state model with the master equation

$$\begin{aligned} \frac{\mathrm{d}P_{mn}}{\mathrm{d}t} =& k_{\mathrm{ts}}(\mathsf{E}_M^{-1} - 1)P_{mn} + k_{\mathrm{degm}}(\mathsf{E}_M^{+1} - 1)mP_{mn} \\ &+ k_{\mathrm{tl}}(\mathsf{E}_N^{-1} - 1)mP_{mn} - k_{\mathrm{degp}}(\mathsf{E}_N^{+1} - 1)nP_{mn}. \end{aligned} \tag{2.22}$$

This transforms to

$$\frac{\mathrm{d}P_n}{\mathrm{d}t} = k_{\mathrm{ts}} \left[(1 - P_{\mathrm{tl}}) \sum_{r=0}^{n} \left(P_{\mathrm{tl}}\mathsf{E}^{-1}\right)^{r} - 1\right] P_n - k_{\mathrm{degp}}(\mathsf{E}^{+1} - 1)nP_n. \tag{2.23}$$

All mRNA dependence is removed and now the translation term is that of a multi-step process where for a protein state n, transitions into that state can come from any protein abundance state from 0 to $n-1$. Transitions out of the state n for translation can go to any state greater than n. These transitions are accounted for by the 1 in the translation term since the geometric distribution is normalized and the probability for a single transcription is P_{tl}. The protein decay term is not affected by this manipulation. This substitution is an adiabatic approximation that assumes that all mRNA is created or degraded between each protein reaction. This approximation is only valid when there is a large difference in timescales between the mRNA and protein dynamics. For *E. coli*, $\gamma \approx 53$ based on degradation rates assumed for our model (cf. Table 2.1). This ratio is sufficient large that the mRNA and protein dynamics are well-separated. For the three-state model, this approximation reduces the dimensionality of the underlying system of equations by a factor of m_{max} (≈ 50) and significantly decreases the computational time. Analytically, this reduces Eqs. 2.16a–2.16c to a single dimension. A similar geometric bursting approximation for mRNA was used in a previous calculation[82] however only the average and variance of the protein abundance at steady-state could be calculated. To compute the full probability distribution of protein abundances and the switching rates between the induced and uninduced phenotypes a different method of solution must be employed.

We numerically integrate the CME using the FSP to compute an approximate solution[23,24]. This method is significantly faster than directly sampling the CME using the stochastic simulation algorithm (SSA). Computational time can be over 100 times shorter using this method compared to using the SSA with a sufficient number of realizations to be comparable to the error achieved from the FSP. To achieve this level of performance, this method truncates the state space at copy numbers that are not expected to be well-populated. The error due to the truncation is controlled adding fictitious states denoted *sinks* that accumulate probability that leaves the projection. Any transitions to states outside the projection are rerouted to these states. For example see Figure 2.2a, where the sink states are labeled η. The total probability lost due to the projection is the sum of probability in the sinks, which estimates the total error due to the truncation of state space. To find the optimal size of the projection, the system of ODEs is integrated while monitoring the accumulated probability in the sinks. If the sink probability increases past a threshold, the calculation is restarted with a larger projection space. The method for increasing the projection space depends on the type of problem being considered[23,83,84]. For our simple

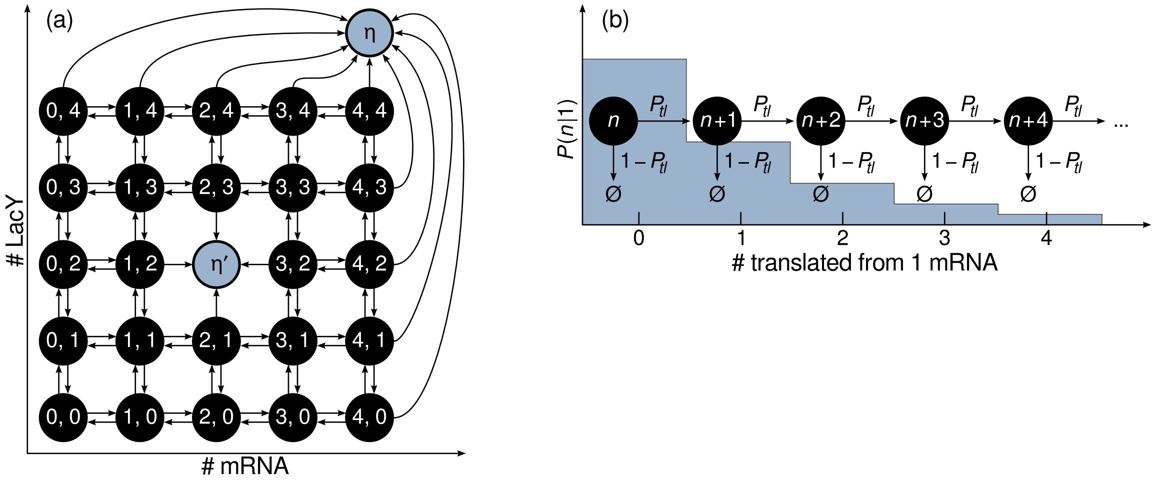


Figure 2.2 (a) Finite state projection. Consider a gene–mRNA–protein model having only one transcriptional state. The FSP method truncates the state space, eliminating states that are not well-populated. Here we have chosen the projection $n, m < 5$. Transitions that would lead to n or $m = 5$ are rerouted to the sink η. As the equations are evolved in time, the total probability accumulated in η is monitored. If η increases past a threshold, the calculation is restarted with a projection including more states. This technique can also compute mean first passage times. The state $n = 2, m = 2$ is replaced by a sink state η' from which no probability can flow out. The accumulated probability in η' is the distribution of first passage times to $n = 2, m = 2$. (b) Geometric burst approximation. Each time a mRNA is transcribed, it can be translated increasing n by 1 or be degraded and become unable to be transcribed further. This process leads to a geometric distribution of protein translated from a single mRNA.

1-D CME, we choose for our state projection $n \in [0, \alpha\mathcal{N}]$ where α is 1.25 initially and is increased when the calculation must be restarted.

Writing the CME in matrix form would lead to a large non-sparse matrix due to the tail of the burst distribution, because each state (n, s) can transition to (m, s) where $n < m \leq \alpha\mathcal{N}$. Considering protein burst reactions alone, this situation leads to a block diagonal matrix where each of the three blocks are lower triangular. This problem can be solved by truncating the geometric series used in the burst approximation leading to a generator with a band structure. Any transitions from translation where the change in protein number is larger than a threshold are rerouted to a sink. Error due to the truncation of the geometric expansion is monitored by watching the accumulation in this sink. We found that 120 terms in the expansion were sufficient to achieve a total error of less than 10^{-7}. Since Eqs. 2.16a–2.16c are not explicitly dependent on time, we integrated the equations using an off-the-shelf matrix exponentiation package[85].

2.3.2 Mean First Passage Times

We are interested in the mean switching rates between the induced and uninduced metastable states. These rates can be computed from the mean first passage time (MFPT) for the system to evolve from a stable fixed point of the deterministic rate equations Eq. 2.14—either n_{LO} or n_{HI}—to the unstable fixed point n_0. Although there are elegant techniques available for estimating the stability of metastable states[86–88], we can compute these rates quickly and accurately using the geometric burst approximation coupled with the FSP. We compute the MFPT by adding an additional sink at n_0, and integrating the equations with the initial condition that the copy number probability at the initial stable fixed point is unity[89].

The probability accumulated in this sink, $\eta(t)$, is the cumulative distribution function (CDF) of first passage times. Aside from an initial transient period, the time evolution of $\eta(t)$ is well-described by $\eta(t) = 1 - \mathrm{e}^{-t/\tau}$. We could then integrate the equations out to a time t_f where $\eta(t_f) \approx 1$ and extract the MFPT τ by curve fitting. To save computational resources we use a different approach which does not require the equations to be integrated to t_f.

The mean of a distribution can be computed from its CDF by integrating the complementary

CDF over its full domain. We will use this fact to estimate the MFPT by integrating over the known part of the full domain and adding to this value an estimate of the rest of the integral. At each time t_s during the integration of the CME, we integrate the complementary CDF $1-\eta(t)$ numerically up to the current time step t_s and estimate the contribution from the integral from t_s to ∞ by assuming the complementary CDF decays exponentially for times greater than t_s. Thus at each time step we compute

$$\tau(t_s) = \int_0^{t_s} (1-\eta(t))\,\mathrm{d}t + \frac{1}{r(t_s)}\eta(t_s) \tag{2.24}$$

where

$$r(t_s) = -\frac{\mathrm{d}}{\mathrm{d}t}\ln\bigl(1-\eta(t)\bigr)\Big|_{t=t_s}. \tag{2.25}$$

We stop when the change in $\tau(t_s)$ per time step is less than a threshold:

$$\left|\frac{\mathrm{d}\tau(t)}{\mathrm{d}t}\right|_{t=t_s} < \eta_{\text{converge}}. \tag{2.26}$$

For a tolerance of total error less than 5×10^{-4}, the algorithm requires ≈ 300 steps to converge.

2.3.3 Calculating the range of bistability

We define the macroscopically bistable range to be the range of inducer concentrations where the probability to be either LO or HI is greater than 10%. We find this range for a given set of parameters by first finding the range of external inducer concentrations that lead to three fixed points in the deterministic rate equations by counting the roots of Eq. 2.14. This range of concentrations is then searched, looking for where the probability to be in the induced state

$$P(\text{HI}) = \frac{\tau_{\text{HI}}}{\tau_{\text{HI}} + \tau_{\text{LO}}} \tag{2.27}$$

is 0.1 and 0.9. Here τ_x is the MFPT out of the phenotype x. We then search this bistable range for the concentration that either induction state is likely ($P(\text{HI}) = 0.5$), and ensure that the rate at the candidate concentration is greater the minimum acceptable value. The search is performed using the MFPT computed from geometric burst/FSP method using using standard root finding algorithms.

2.3.4 Sampling the CME using the stochastic simulation algorithm

To test our results from the FSP calculations, we used the SSA[8] to compute the PDF and MFPT. To look at the microscopic dynamics of switching events, the trajectories from the SSA simulations were segmented into induced, uninduced, and switching sections using thresholding heuristics that allow for arbitrarily long switching trajectories.

The heuristics first mark regions of the trajectory as LO if the protein number is lower than the unstable fixed point, and HI otherwise. An acceptable switching event is a segment of time t_0 long that does not leave LO (HI) followed by an intermediate segment less than δt_{max} long that fluctuates between the two states followed by a segment of time t_0 long that stays in HI (LO). The lead-in and lead-out times will not affect the results since the model is Markovian. We chose $t_0 = 10$ cell cycles. The maximum switching time δt_{max} was chosen so greater than 99.9% of the trajectories with long enough lead-in and lead-out times were accepted. The necessary δt_{max} depends on the particular parameter set but was generally of the order of 50 cell cycles.

2.4 Results

2.4.1 Looping parameters that reproduce *lac* bistability

A primary goal for the work is to determine if a minimal three-state model is able to recover the full range of bistability that was experimentally observed for the *lac* circuit with looping, compared to the behavior observed for the circuit where looping was prohibited. However, the model's looped state represents a coarse-grained state composed of many different microscopic states and no experimental data regarding microscopic transition rates are known. We therefore searched the space of all possible parameters related to the loop state to characterize how its addition changes the behavior of model. The *Off* → *Loop* transition has a single parameter k_{fl} and the *Loop* → *Off* transition function has four parameters: k_{lf}^{0}, k_{lf}^{1}, I_{lf}, and H_{lf}. We randomly choose parameter values from this five-dimensional space while keeping all other model parameters fixed and tested the model's bistability properties.

To limit the size of the search, we bounded the parameter space such that only biologically reasonable parameter sets were sampled. The *Off*→*Loop* transition models the physical process of forming a DNA loop. If it is too slow to observe, the three-state model reverts the two-state model so we set the minimum rate of k_{fl} to be one event per 100 cell cycles. Likewise, loop formation requires the binding of a repressor-operator complex to a second operator and must be slower than repressor binding, so we set the maximum of k_{fl} to be one-half the rate of repressor binding, $k_{\mathrm{nf}}^{0}/2$.

For the *Loop*→*Off* transition Hill function $k_{\mathrm{lf}}(n)$, the parameter I_{lf} determines the approximate inducer concentration at which the transition rate will start to increase, i.e., at what inducer concentration the loop starts to become destabilized. We bounded this parameter such that the denominator in Eq. 2.8 is $\mathcal{O}(1)$ for inducer concentrations between 1 μM and 100 μM. This bound ensures that the transition rate function is able to switch between its low and high ranges over the experimentally relevant range of inducer concentrations. If I_{lf} was too large or too small, $k_{\mathrm{lf}}(n)$ would be effectively constant over the full range of LacY abundances and not provide feedback. The Hill coefficient H_{lf} determines the sharpness of the response and was restricted to be between 2 and 4 since inducer–repressor binding is cooperative with 4 binding sites. An *in vitro* study of DNA loop stability for the wild-type *lac* operon reported the mean lifetime of the loop to be on the order of the cell cycle[90]. In light of this observation, for the minimum transition rate k_{lf}^{0} (in the absence of inducer) we generously bounded this value by setting the minimum rate for the search to be one dissociation per ten cell cycles and the maximum rate to be one-half of the repressor unbinding, $k_{\mathrm{fn}}^{0}/2$, since by definition loop dissociation must be slower than repressor unbinding. The maximum transition rate k_{lf}^{1} (in the presence of saturating inducer) was restricted to be faster than twice the zero inducer rate k_{lf}^{0} but slower than 100 events per cell cycle.

We scanned 125 113 unique parameter sets by uniformly drawing a random value for each parameter from the above ranges. For each set we calculated the range of macroscopic bistability, defined at the low end by the inducer concentration where 10% of the steady state population is induced and at the high end by the inducer concentration where 90% of the population is induced (see section Section 2.3.3). Whereas the two-state model exhibits bistability near 10 μM (10.1–10.9 μM), the three-state model can produce bistability at inducer concentrations anywhere from 10.6–518 μM for sets of parameters constrained to the range above. Additionally, the range of macroscopic bistability for the two-state model is very small, 0.8 μM, while the addition of a third state can produce much wider ranges, up to 300 μM. As the overall switching rate to *Loop* decreases, the bistability range increases on average.

For each set we calculated the mean switching time at the inducer concentration where 50% of the steady-state population is uninduced and 50% is induced. At this concentration the time to switch to LO or to HI is identical and we use this time ($\tau_{50\%}$) as a measure of a parameter set's characteristic switching time. For the two-state model, $\tau_{50\%}$ was 55 cell cycles, but with the three-state model $\tau_{50\%}$ values could be obtained anywhere from 26–9.5×10^5 cell cycles. Decreasing I_{fl} has the effect on average of increasing the switching time $\tau_{50\%}$ (see Section A.3.)

Having performed a parameter search for the looped state, we then selected for further study only those parameter sets that reproduced the range of bistability seen in one experiment[43]. A parameter set was deemed acceptable if its range of bistability contained the range 50–60 μM and $\tau_{50\%}$ was less than 100 cell cycles. Approximately 0.9% of the random samples satisfied these criteria. Figure 2.3 shows how the accepted parameters were distributed in the search space.

The absolute value for several of the parameters appear to be unimportant to the switching properties. The Hill coefficient H_{lf} and the basal *Loop*→*Off* rate k_{lf}^{0} are well-distributed within their allowed ranges. Likewise, the *Off*→*Loop* rate k_{fl} is uniformly distributed below 45 cell cycles. On the other hand, the inducer transition concentration I_{lf} and the maximum *Loop*→*Off* rate k_{lf}^{1} are more likely to take on values in certain ranges of the search space, high and low, respectively.

Some parameters showed a degree of mutual dependence. The basal *Loop*→*Off* rate k_{lf}^{0} and the *Off*→*Loop* rate k_{fl} have a strong linear dependence ($r^2 \approx 0.6$). Together with their uniform distribution, this dependence suggest that the total fraction of the time spent in the looped state in the absence of inducer, which determines the basal permease copy number, is an important

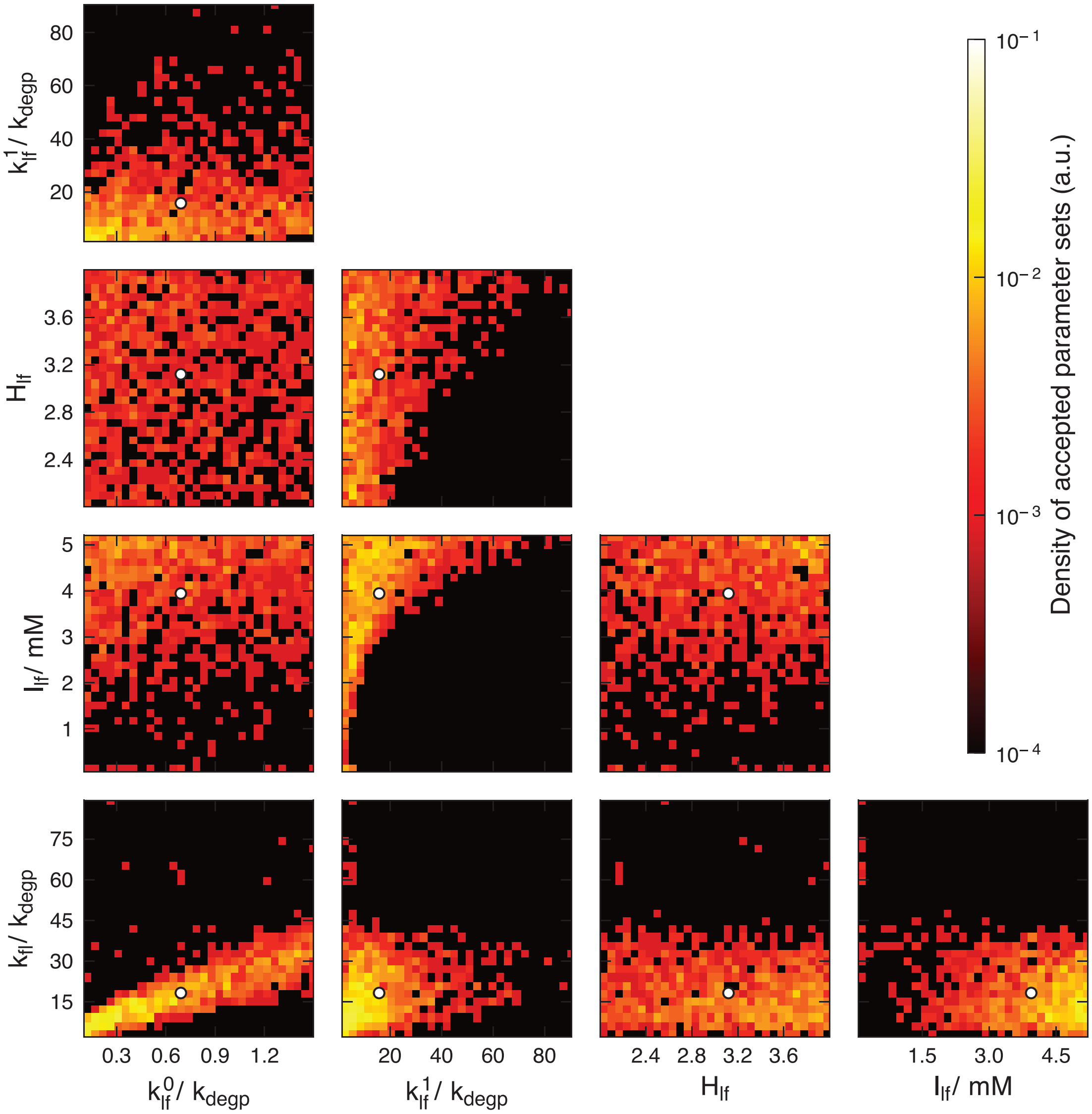


Figure 2.3 Two-dimensional distributions of parameter values in sets with bistability ranges including 50–60 μM. Lighter areas indicate greater density. A large range of parameter sets exhibiting bistability with fast switching rate were found. The dot marks the mean of the distribution. This parameter set is investigated in later figures.

property for switching, but the rate of cycling between the looped and unlooped states is relatively unimportant.

The inducer transition concentration I_{lf} and the maximum *Loop*→*Off* rate also appear to be dependent. All of the accepted sets fall above a convex region of the I_{lf}–k^1_{lf} phase space: $k^1_{\mathrm{lf}} \lesssim {I_{\mathrm{lf}}}^4$. If the maximum *Loop*→*Off* transition rate is high, then the transition concentration must also be high and if the transition concentration is low the *Loop*→*Off* transition rate must be low. A similar but less pronounced dependence is seen between the maximum *Loop*→*Off* rate and the Hill coefficient H_{lf}. This dependence is simply because there appears to be a maximum *Loop*→*Off* rate. When I_{lf} is large, $k_{\mathrm{lf}}(n)$ will not be able to reach its saturating level so values of k^1_{lf} larger than the maximum rate can be used.

The distribution of calculated switching properties for the accepted parameter sets is shown in Figure 2.4a-e. The onset of bistability, the external inducer concentration necessary for 10% induction probability can begin, as low as 30 μM and can last through 120 μM. The width of the bistable range can be from 20–80 μM. $\tau_{50\%}$ values fall between 25–100 cell cycles. The properties of LacY in these cells is also of interest. The uninduced states were centered at zero LacY for all acceptable sets and the induced state means were found at 2350±10.

We tested the sensitivity of the model to changing the leakage factor ϵ. Over the likely range of ϵ from 5.7×10^{-4} to 2.1×10^{-3}, the range of bistability only changed from 23.0 μM to 22.75 μM. The switching lifetime $\tau_{50\%}$ only changed from 54.7 to 53.8 cell cycles.

2.4.2 Details of a representative parameter set

We chose as a representative parameter set the mean of the acceptable parameter set distribution since that approximates the centroid of the hypervolume that encloses all acceptable parameters. In Figure 2.3 this parameter set is indicated by a white point, the details of these parameters are enumerated in Table 2.1, and the transcription state transition functions are plotted in Figure 2.8. The PDF computed for this parameter set and the two-state model are shown in Figure 2.7 compared to the PDF computed from the SSA. The combined FSP and geometric burst approximation predict the probability distributions with a high degree of agreement. The total probability lost

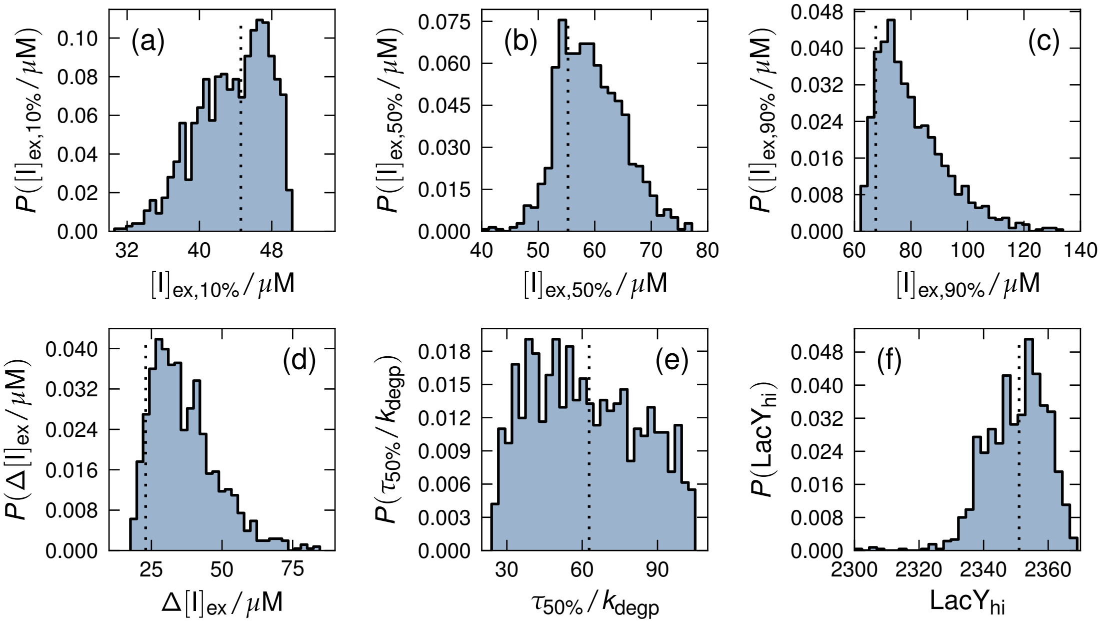


Figure 2.4 Distributions of calculated values from the set of parameters with bistability. The dotted line marks the values computed from the selected set in Figure 2.3. These are distributions of inducer concentrations at (a) 10%, (b) 50%, and (c) 90% probability to be induced, (d) extent of the bistable range, (e) metastable state lifetimes at 50% induction probability, and (f) average LacY count in the induced state.

from transitions out of the projection was less than 10^{-4}, and the probability lost from truncating the geometric distribution was less than 10^{-7}. The deviations of the FSP PDF from the SSA PDF are due to the adiabaticity assumption of the geometric burst approximation, not from probability loss from truncation. A numerical solution of the CME for the two-state model with explicit mRNA dependence showed exact agreement with the simulated distribution (see Section A.2) with drastically increased computational cost.

The dependence of induction state lifetime on the external inducer concentration is generally exponential as shown in Figure 2.5. Since changing the inducer concentration just affects the effective transcription rate, this functional dependence is in agreement with what is known about the dependence of switching rates in genetic networks[78]. This parameter set is bistable within the external inducer concentration range of 42–66 μM using 10% minority as the bistability metric. The switching time at equal induction probability is 63 cell cycles. For the two-state model, the bistability range was 10.1–11.0 μM with a switching time of 52 cell cycles. These values are in agreement with the results presented in Choi et al.[43] however the bistability of the non-looping mutant was only tested down to 20 μM external inducer so a transition from the induced phenotype to uninduced could not observed.

The mean protein number in the uninduced states appear to be in agreement with experiment. This parameter set exhibits an uninduced state centered at 0 LacY whereas the two-state model has an uninduced state centered at 41. The location of the modes is in agreement with Choi et al.[43] since they showed for mutants unable to produce permease that the LacY distribution was centered at 0 for low inducer concentrations and the LacY distribution for mutants that cannot form DNA loops or produce permease was centered around 75. They did not report copy numbers for the induced state, however we found that adding DNA looping resolved the differences between the induced and uninduced states by moving the induced maximum from 1778 to 2351 and decreasing the width of the distribution by a factor of 2.7.

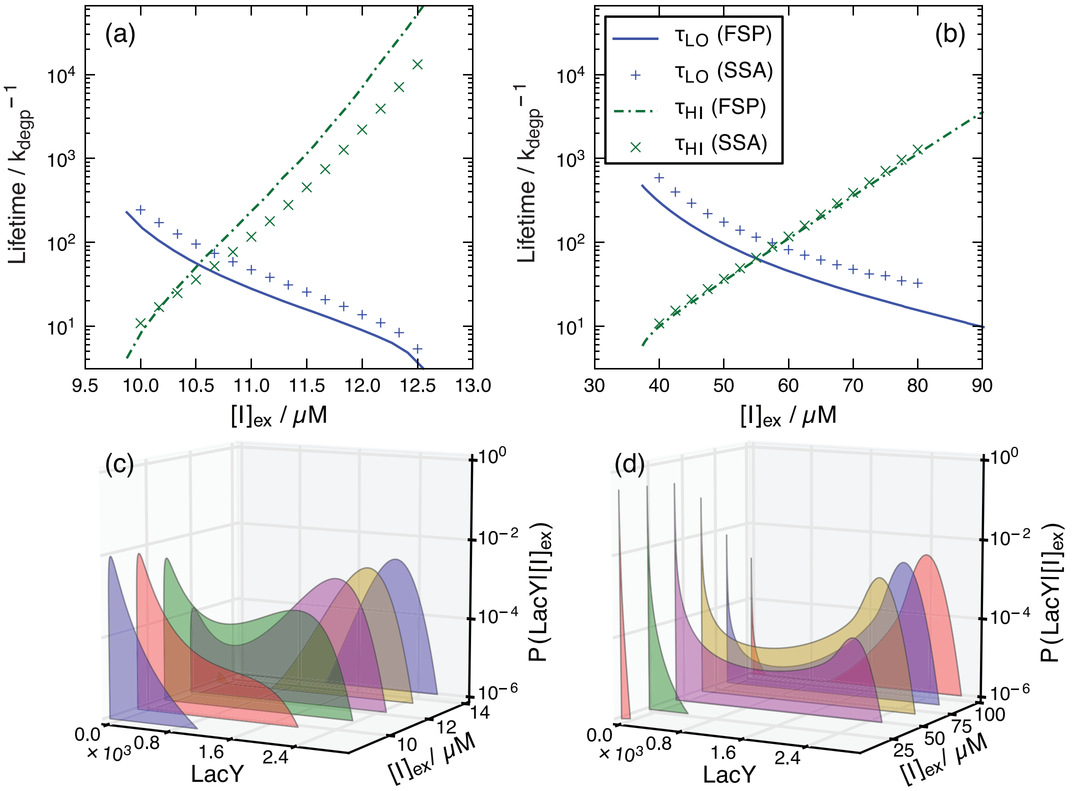


Figure 2.5 Comparison of LO and HI metastable state lifetimes for both (a) the two-state model, (b) the three-state model computed via simulation (SSA) and using the geometric burst approximation (FSP) reported in cell cycles. The deviations from simulation are due to the geometric burst approximation. The point where the two lifetime functions cross is the equal probability concentration, and the lifetime here sets the scale for switching times. Dependence of PDF on external inducer concentration for (c) the two-state model and (d) the three-state model computed using the geometric burst approximation.

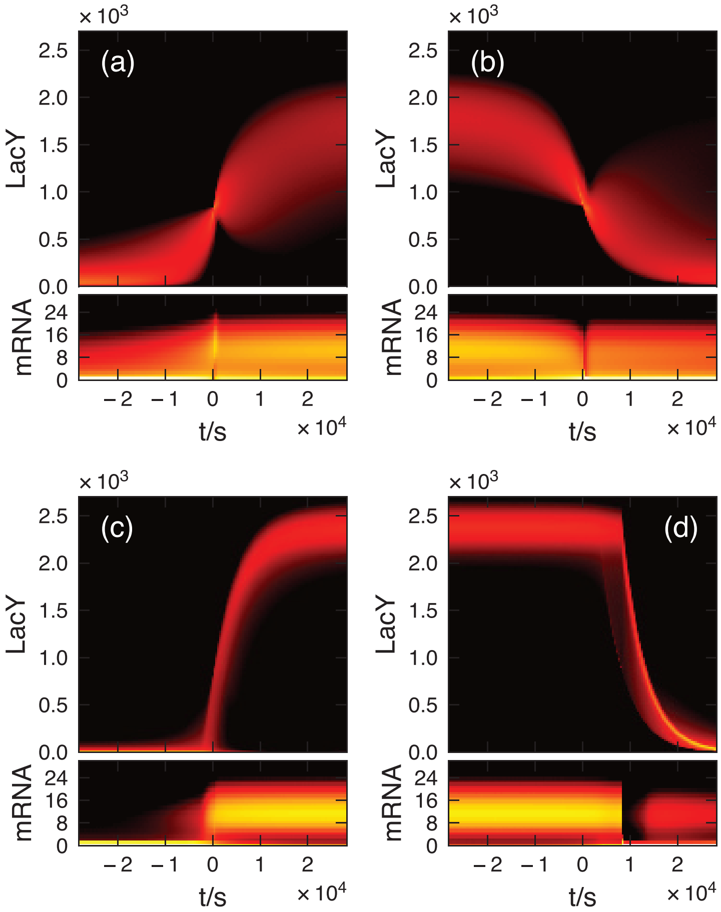


Figure 2.6 Distributions of protein and mRNA abundance as a function of time for (a) the two-state model transitioning LO→HI, (b) transitioning HI→LO, (c) the three-state model transitioning LO→HI, and (d) transitioning HI→LO. Induction and uninduction events in the two-state model are driven by protein number fluctuations, however in the three-state model fluctuations of the transcriptional state drive the switch, leading to quasi-deterministic behavior.

2.4.3 Microscopic dynamics of induction state transitions

To investigate how the system transitions between the two phenotypes, we plotted protein and mRNA abundance histograms at regular time steps during a switching event in Figure 2.6. We computed multiple trajectories using the SSA and searched them for segments in which the system switched between metastable states. All segments were then translated in time such that the first transcriptional state change event before switching is at $t = 0$.

For the LO→HI transition, the three-state model shows quasi-deterministic switching behavior with an average switching time of 2.9 cell cycles. The duration and shapes of the protein trajectory during a switching event are all uniform. The two-state model does not exhibit this behavior. Instead the transitions are stochastic and irregular. The reason for this behavior appears to be that the three-state model stays in the *Loop* state in the uninduced phenotype and very rarely switches into *Off*. Prior to switching the mRNA population increases leading to more frequent bursts of protein, which can carry the system from the *Off* transcriptional state into *On*. Once *On* is reached switching back to *Off* is rare, leading to the deterministic switching behavior.

The HI → LO transition is also quasi-deterministic and is slower than the LO → HI transition (8.3 cell cycles). However the mechanism triggering the transition out of the induced state is different. Here random fluctuations of repressor binding are what drive the transition. The transcriptional state fluctuates into *Loop* by quickly passing through *Off*. If the system stays too long in the *Loop* state, the mRNA number will have decayed back to the distribution found in LO and the protein number will fall deterministically.

2.5 Discussion

Our three-state model attempts to incorporate more of the biological details of *lac* regulation than previous mathematical models. The coarse-grained loop state captures the essence of the DNA looping regulatory element by providing a long-lived state with alternative transcriptional properties. Our model gives good agreement with the limited *lac* bistability data that is available. In order to select better parameters and to determine whether our coarse-grained loop state captures *all* of the relevant dynamics, better experimental characterization of the switching properties of *lac* are required. Specifically, the mean switching times as a function of inducer concentration, for both induction and uninduction are key observables that are currently unknown.

Our model is relatively robust to variations in most parameters; a wide distribution of parameter sets satisfy the experimental criteria. Most fluctuations in the parameters should not disproportionately affect the behavior of the switch. We speculate that this parameter insensitivity would allow the genetic switch to function in conditions such as varying ribosome numbers, mutations that affect the reaction rates between DNA/inducer states, or abnormal environmental conditions.

We applied the results of a thermodynamic model of DNA looping in the *lac* system[91] to compare to our three-state model. This thermodynamic model uses *in vitro* data from Oehler et al.[46] to compute the free energies of operator/repressor binding and DNA looping. From these free energies we are able to estimate the population of our transcriptional states at zero external inducer. We consider the *On* state to be the operon with no operators bound, the *Loop* state to be all looped states and all states with only auxiliary operators bound, and the *Off* state to be the remaining possible DNA/repressor configurations. Using the free energies of these states we found that $P(On) = 4.7\times10^{-6}$, $P(Off) = 2.2\times10^{-2}$ and $P(Loop) = 0.978$ (see Section A.4). Setting external inducer to zero, the FSP calculation for our mean parameter set yields transcriptional state probabilities of $P(On) = 4.6\times10^{-4}$, $P(Off) = 3.7\times10^{-2}$ and $P(Loop) = 0.963$. Our *Off* probability compares well with the probabilities computed from *in vitro* data. It is also possible to estimate the leakage factor ϵ from free energies. When the system is in the *Loop* state, it is transcriptionally active only in a non-looped conformation. The leakage factor then must be the conditional probability to be in an unlooped conformation while in the *Loop* state. Using the data from Oehler et al.[46] we find that $\epsilon = 1.24\times10^{-4}$. This result compares within an order of magnitude of the value estimated from the burst size of the uninduced state experimentally measured in Choi et al.[43] used in our modeling. These values computed from Oehler et al.[46] are for a LacI count of 50. However, in this work we are using ten LacI per volume. Extrapolating these free energies to ten LacI shows better agreement (see Section A.4). This is all suggestive that the parameters selected from the parameter search are reasonable for the *lac* system.

The PDF of a stochastic system fully describes its switching properties and when comparing the shapes of the PDF for the two- and three-state models (see Figure 2.7) one can see significant differences. In the three-state model, the transition between the two stable phenotypes has a sharp change in slope on each side, leading to a shallower transition basin between the two states. Also, the peaks in the three-state model are significantly narrower and further separated than in the two-state model. The net effect of these properties of the PDF is to allow the three-state model to have two well-separated phenotypes while still allowing relatively easy transitions between

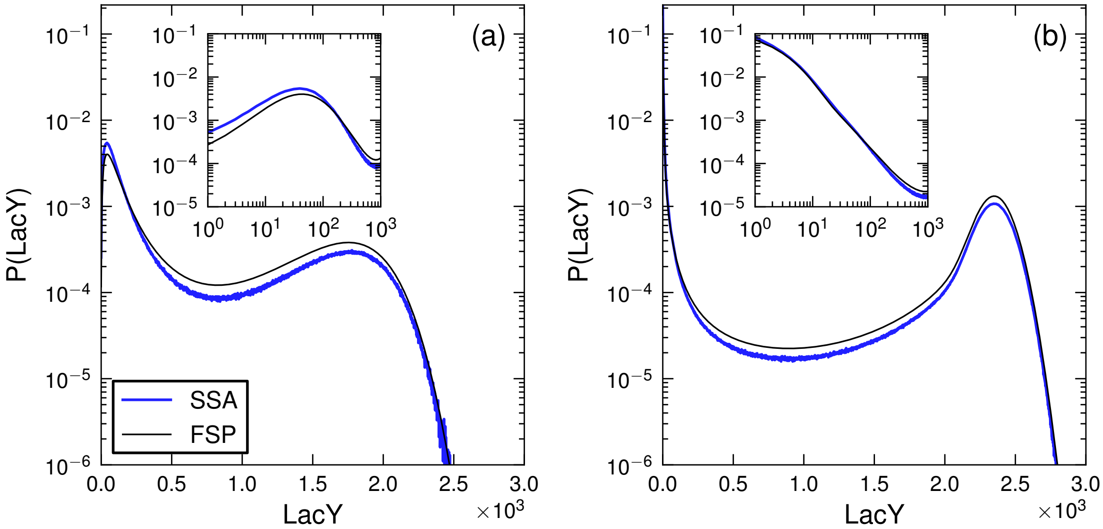


Figure 2.7 Protein number PDF computed from simulation (SSA) and the geometric bursting approximation (FSP) for (a) the two-state model and (b) the three-state model. The inducer concentration was chosen such that the system was equally likely to be in either induction state. The inset shows a closer look at the uninduced state distribution on a log-log scale. Slight deviations from simulation are due to the geometric burst approximation. Computations of the PDF including explicit mRNA dependence (not shown) fit the simulation identically.

them. In the two-state model, as the LO and HI phenotypes become further separated it becomes more difficult for a random fluctuation to move the system between them. The three-state model breaks this dependence by creating a higher-probability connecting basin. These characteristics may be desirable from a biological perspective by allowing a population to be prepared for and to react quickly to changing environmental conditions and are the hallmark of an effective bistable switch.

In our two-state model, the Hill functions describing the regulation appear to prohibit sharply defined phenotypes while maintaining fast switching rates. We wondered if there was another regulatory function that could impart these properties to the two-state model. To check, we calculate the effective *Off* → *On* transition function that yields the most similar probability distribution and switching times when used in a two-state model. The *Off* and *Loop* states are combined into a single slowly transcribing state, *Off*$'$. The necessary leakage factor for this aggregated state can be estimated by considering the *Off* and *Loop* states at zero external inducer. Since no transcription

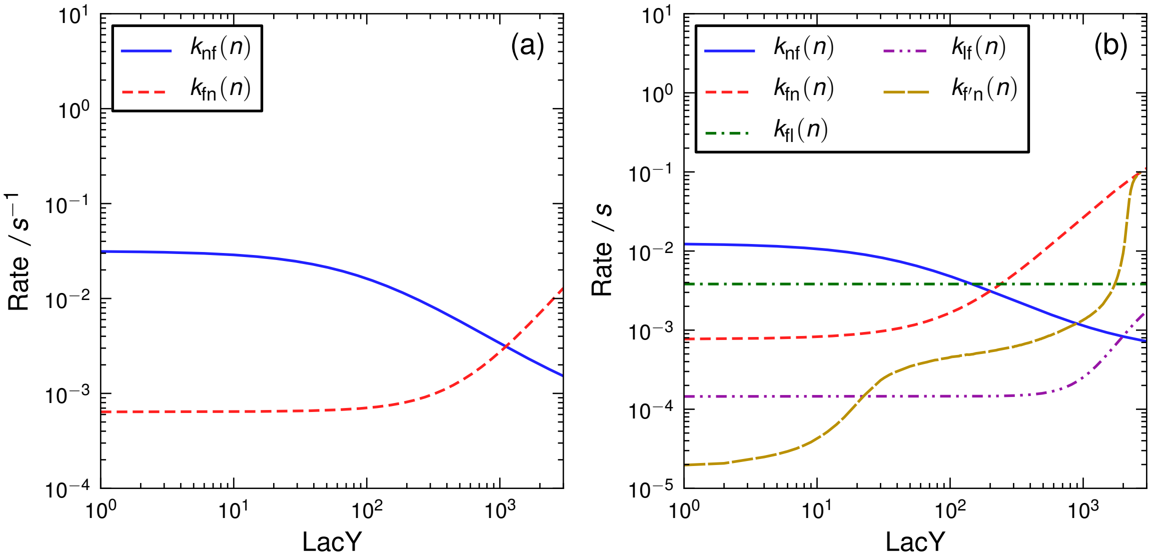


Figure 2.8 Comparison of the transcriptional state switching functions used. (a) Functions used in the two-state model at $[\mathrm{I}]_{ex} = 10.5\,\mu\mathrm{M}$. (b) Functions computed using values from the mean of the parameter search distribution for the three-state model at $[\mathrm{I}]_{ex} = 55.3\,\mu\mathrm{M}$. The function $k_{\mathrm{f'n}}(n)$ is the extracted switching function from an effective two-state model built from the results of the three-state parameter search.

occurs in the *Off* state, the new leakage factor is

$$\epsilon' = \frac{\epsilon P(Loop)}{P(Off) + P(Loop)} \approx 0.96\,\epsilon. \tag{2.28}$$

We can now compute an effective switching rate from *Off*′ to *On* from the FSP results.

The probability to be in *On* as a function of LacY abundance is approximately

$$P(On|\mathrm{LacY}) = \frac{k_{\mathrm{f'n}}(n)}{k_{\mathrm{f'n}}(n) + k_{\mathrm{nf'}}(n)} \tag{2.29}$$

where $k_{\mathrm{nf'}}(n) = k_{\mathrm{nf}}(n)$ since the *On* state is not affected by combining *Loop* and *Off*. From the FSP results we compute $P(On|\mathrm{LacY})$ using Bayes' rule and solve for $k_{\mathrm{f'n}}(n)$. The resulting function is plotted in Figure 2.8b.

When used in the two-state model, the new rate function yields a similar protein number probability distribution to the three-state model with similar switching rates between LO and HI. However the switching dynamics are different, and they must be due to the reliance on the *Loop* state transitions. Switching on is similar to the full three-state model, however switching off is different. The transitions are more rounded and not as sharp.

In order to get a PDF with fast switching between the metastable states without increasing the noise of each state, a switching function like the one computed for the effective two-state model is required. This dependence of switching rate on inducer concentration is not likely to be achieved with a single transcription factor binding. The double step is not realizable in a single reaction and the effective Hill coefficients of these steps are large with the first step having a Hill coefficient of 3 and the second having 18.

The microscopic dynamics of the induction event in the three-state model showed that in order for the switch to induce, the system must switch from *Loop* to *On* through *Off*. This observation supports the argument made in Choi et al.[43], in which a single molecular event triggers the induction of the switch. They claim that dissociation of the DNA/repressor complex causes a large burst of transcription that induces the cell. This phenomenon is seen in the stochastic simulations of the three-state model. Transitions from the *Loop* state are what drives the transition, not fluctuations in protein number as seen in the two-state model.

In Choi et al.[43], they were not able to observe bistability for external TMG concentrations above 20 μM for mutants without the ability to form DNA loops. Our two-state model predicts a small window of bistability between 10.1 and 11.0 μM of external inducer. It might not be possible to observe bistability in this mutant, however a minimum inducer concentration for induction should be measurable. This model could also be tested further through experiments to measure the induced and uninduced state lifetimes as a function of external inducer concentration. We expect an exponential dependence on the TMG concentration.

Adding a looped state to the two-state model vastly increases the range of bistability. The bistability range for the three-state model is on average 43 times larger for the parameter sets that meet our selection criteria, and for any three-state parameter set that is biologically reasonable it can be up to 370 times greater. It should be possible to investigate this enhanced range by increasing the stability of the loop. It has been shown that changing the distance between operators and the sequence of the operators allows one to directly control the loop stability[92]. A way to modify the stability of the loop without changing other parameters could be to change the sequences and locations of the secondary operators since this perturbation should not affect the binding affinity for primary operator and thus change the $On \leftrightarrow Off$ rates. Increasing the distance between the main and auxiliary operators should only affect the rate k_{fl}. Manipulating this rate could allow one to continuously transform the three-state model into the two-state model.

The transition rate k_{fl} is a complicated function of the operator spacings, because it depends on the effective concentration of the operator around the operator binding site on LacI. This effective concentration is called the J factor and it depends on the tension in the DNA molecule required to bend the operator site to the operator binding site, the torsion in the strand due to matching the correct side of the DNA molecule to the operator binding site, and the configurational entropy of bringing the operator binding site and operator together. This dependence highly complicates how the looping rate is affected by changing the operator separation. Instead, one could measure the transition rate and bistability properties simultaneously to verify our results.

The bistability range and phenotypic lifetime at 50% induction probability are plotted in Figure 2.9 as a function of the looping rate. We predict that at faster looping rates, the bistability range widens while the onset of bistability occurs at greater inducer concentrations. The lifetimes of the metastable states increase with looping rate as well. A way to experimentally confirm these results may be to prepare an ensemble of mutants with differing spacing between O_1-O_3 and O_1-O_2 and measure both the bistability range and the looping rate of each mutant. The looping rates could be measured using single molecule FRET measurements between the operators[93].

2.6 Conclusions

The geometric bursting approximation coupled with FSP is a fast way to numerically solve CME involving transcription while accurately accounting for transcriptional noise. Formation of DNA loops in a genetic switch has a profound effect on the sensitivity of the switch to external inducer concentration, allowing it to be bistable over a much larger range of concentrations. Looping also

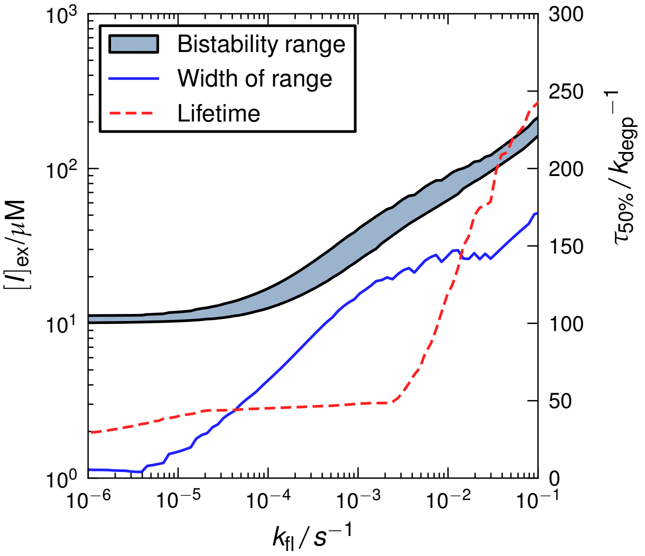


Figure 2.9 Dependence of the bistability range and metastable lifetime $\tau_{50\%}$ at equal induction probability on the looping rate k_{fl}. By increasing the distance between the primary and auxiliary operators, the three-state model can be continuously changed into a two-state model. In general, faster looping rates lead to greater ranges of bistability. The minimum inducer concentration necessary for bistability also increases with the looping rate. The curves appear noisy since the fixed point n_0 changes as k_{fl} changes. See Section A.3 for details.

affects the PDF of protein copy number, allowing for fast switching times between metastable states while maintaining sharply differentiated states of induction, hinting at a possible design methodology in synthetic biology where fast, highly resolved switch states are needed.

Chapter 3

Towards a whole-cell model of ribosome biogenesis: Kinetic modeling of small subunit (SSU) assembly*

3.1 Introduction

Translation is the universal process that synthesizes proteins in all living cells. Sequence (and structural) signatures in the ribosomal RNA (rRNA) were used to classify all living organisms into the three domains of life[95,96]. Ribosomal protein can themselves be signatures of ribosomal evolution and, in the case of bacteria, roughly one third of them are unique with the remaining common to all three domains of life[96,97]. Ribosomes constitute approximately one fourth of a bacterial cell's dry mass, and biogenesis of the ribosome, together with the other cellular processes involved in translation, consume a significant fraction of the cell's energy budget. A whole-cell model of ribosome biogenesis is crucial for our understanding of cell growth, however a comprehensive dynamical description of the biogenesis process is still missing.

In bacteria, the precise synthesis and assembly of a ribosome[98] involves at least four critical

*Chapter 3 contains material reproduced with the permission of Elsevier from Earnest TM, Lai J, Chen K, Hallock MJ, Williamson JR, and Luthey-Schulten Z (2015). Toward a whole-cell model of ribosome biogenesis: Kinetic modeling of SSU assembly. *Biophys. J.*, **109**(6), pp. 1117–1135. doi:10.1016/j.bpj.2015.07.030.

steps: transcription of rRNA from multiple ribosomal operons; synthesis of the r-proteins, which is regulated on the translational level based on organization of the r-protein operons in the genome; post-transcriptional processing and modification of both the rRNA and r-proteins; and highly coordinated assembly of r-proteins and rRNA towards the mature ribosomal subunits. All these events occur constantly and in parallel throughout the cell cycle.

Ribosomal assembly involves the cooperation of many molecular components. The 30S small subunit (SSU), tasked with the initial binding of messenger RNA (mRNA) and its decoding, is composed of the 16S rRNA and 21 r-proteins. The 50S large subunit (LSU), tasked with channeling growth of the nascent polypeptide chain through peptide bond formation, is composed of the 5S and 23S rRNA and 33 r-protein. These 54 proteins must diffuse through the cell to find their rRNA and bind in a well-defined assembly order. These proteins are classified by their order of binding to the rRNA. Primary proteins bind to the bare rRNA, secondary proteins require the presence of certain primary protein in order to bind, and tertiary proteins require the presence of a secondary protein to bind. The r-protein can compose 9 – 22 % of the total protein counts in the cell[99,100]. In addition, approximately 20 assembly cofactors are engaged to facilitate the process at various assembly stages.

The rich complexity of 30S assembly process attracted Nomura et al. [101], who first observed how the binding stability of r-proteins can depend on the prior binding of other r-proteins. Using equilibrium reconstitution experiments at temperatures optimal for the growth of *Escherichia coli* (37 °C), Nomura constructed a hierarchical dependency map of the assembly process (Figure 3.1). Progress in biophysical approaches has increased our understanding of *in vitro* ribosomal self-assembly through the protein assisted dynamics of RNA folding[102–104], and the kinetic cooperativity of protein binding[105–109]. All of the studies suggest that assembly of the *E. coli* 30S subunit proceeds through multiple parallel pathways, starting with the proteins associated with the $5'$ domain of the 16S rRNA binding first, followed by the central domain proteins, and finally the $3'$ domain proteins.

Using the Nomura map of thermodynamic binding dependencies and kinetic data of protein incorporation, we have constructed comprehensive *in vitro* kinetic models that capture the topology of the r-protein/rRNA interaction network and reproduce the protein binding kinetics of assembly, starting from the bare 16S rRNA or from pre-prepared assembly intermediates, at low and high temperatures[107,108]. Both models are consistent with an assembly mechanism inferred from cryo-electron microscopy (cryo-EM) of 30S assembly intermediates. MD simulations of the early intermediates in the *in vitro* assembly model suggest a molecular basis for the two distinct assembly pathways predicted by the low temperature kinetic model. The low temperature model reproduces all of the control and prebinding experimental kinetics[108,109]. Furthermore, both models predict intermediates central to the assembly process that which would be good candidates for further experimental and computational studies.

The *in vivo* biogenesis of the ribosome is further complicated by spatial segregation of the ribosomes from the nucleoid region[33,110–113]. Cryo-electron tomograms and single molecule experiments have indicated that the full 70S ribosomes[33,114] are partitioned such that 80% are found outside of the nucleoid region; however, the 30S and 50S subunits are found uniformly throughout the cell[113]. In slow-growing *E. coli* (grown in minimal media), roughly 3000 ribosomes accumulate at the cell poles and are almost entirely excluded from the nucleoid[33,110]. In living *E. coli* cells, there can be be as little as one copy of the gene coding for an r-protein. Due to the relatively small number of 30S particles in the process of assembly and the large range of possible intermediates, the counts of specific 16S/r-protein configurations can be of the order of one per cell. To describe the effects and fluctuations arising from the spatial segregation of ribosomes and the low copy number of genes and assembly intermediates, a spatially resolved representation accounting for the discreteness of chemical species is essential for a more realistic treatment of the problem[115].

We present a detailed reaction–diffusion master equation representation of the *in vivo* biogenesis of the SSU, incorporating the spatially inhomogeneous environment of the cell and the stochastic nature of chemical reactions. We have adapted our high temperature *in vitro* assembly model—developed from kinetic studies utilizing pulse/chase quantitative mass spectrometry (P/C qMS)—to an *in vivo* model of ribosome biogenesis including transcription of mRNA and rRNA from DNA localized at their genetic loci, translation of r-protein, and loss of species due to

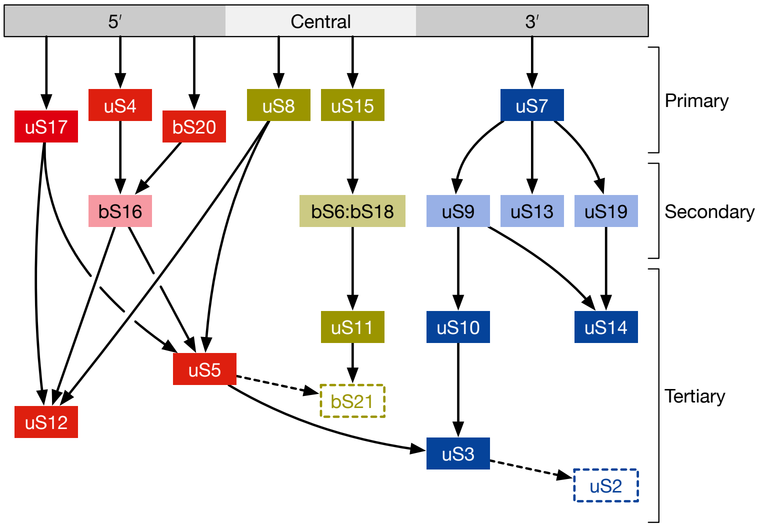


Figure 3.1 Graph of thermodynamic protein binding dependencies to the 16S rRNA[101]. Only the major dependencies used in the *in vitro* model are depicted here. Arrows point from a protein to the protein that is dependent on it. uS2 and bS21, shown in open rectangles, are not included in these models, due to difficulties in acquiring their kinetic data[107].

active degradation of mRNA and dilution arising from cell division. The cell is compartmentalized into cytoplasm and nucleoid regions, which can have different diffusion and intercompartmental transition rates for each chemical species. Our models of *in vivo* 30S biogenesis based on slow-growing *E. coli*[33,114] roughly reproduce the timescale for assembly seen in live cells and predict spatial inhomogeneity in the assembly process.

3.2 Materials and Methods

3.2.1 Generation of assembly networks

The network of r-protein association reactions is constructed programmatically by iteratively adding species and reactions following a rule list. The reaction rule list is a representation of the Nomura map of thermodynamic binding dependencies, in which the binding of a protein to an intermediate is thermodynamically stable only if all of that protein's upstream dependencies are bound. Starting with a stack containing only bare rRNA, an intermediate is removed from the top of the stack and stored in a list of visited species. All possible binding reactions from this species are computed using the reaction rules and their products are only added to the top of the stack if they have not been previously visited. This process is iterated until the stack is empty.

Another rule set is used to assign rate constants to the generated reactions (See Table 3.1). A sequence of rate rules are defined for each r-protein. They consist of additional requirements on the composition of the intermediate independent of the thermodynamic dependencies. To choose the rate parameter for that reaction, each rule is tested in order and the first to succeed is applied to the reaction. These rates are derived from kinetic experiments using pre-prepared intermediates with various proteins bound to the rRNA. For the low temperature model, a rich variety of prebinding experiments are available to derive these rules from. For the high temperature model, no prebinding data is available so only one parameter is used for the binding of a protein to any intermediate. Parameter values are given in Table B.3.

3.2.2 Deterministic modeling and optimization of rate constants at low temperature

The *in vitro* binding process at 15 °C is simulated using the same initial conditions used in the P/C qMS study, which had a 50% excess of r-protein over the 16S rRNA (0.458 μM r-protein versus 0.305 μM 16S rRNA)[107]. The system of ordinary differential equations is solved numerically using using the CVODES package[116] (solver equations derived in Section B.1.) Goodness of fit to the experimental protein binding curves is measured using the objective function

$$\Phi(\{k_i\}) = \frac{1}{\mathcal{N}_{\text{expt.}}\mathcal{N}_{\text{prot.}}(T_1 - T_0)} \sum_{e\in\{\text{expts.}\}} \int_{T_0}^{T_1} \frac{\mathrm{d}t}{t} \sum_{s\in\{\text{r-prot.}\}} \left[\chi(y_{e,s}(t)) - \chi_{e,s}^{\text{expt}}(t)\right]^2 \tag{3.1}$$

which computes the mean-squared error between the experimental and simulated assembly progress curves for the parameters $\{k_i\}$. Here $y_{e,s}(t)$ is the protein concentration s at time t starting from the initial prebinding intermediate e, $\chi_{s,e}^{\text{expt}}(t)$ is a single exponential fit to the actual P/C qMS experiment, and

$$\chi(y) = \frac{p_0}{p_0^* + p_0} + \frac{p_0^*(p_0^* - r_0 + p_0)}{r_0(p_0^* + p_0)} \left(\frac{p_0 - y}{p_0^* + y}\right), \tag{3.2}$$

which converts protein concentrations to an idealized pulse/chase fraction where p_0 is the concentration of labeled protein due to the pulse, p_0^* is the concentration of unlabeled protein due to the chase, and r_0 is the initial rRNA concentration. This assumes that binding is irreversible and all rRNA is converted to intermediates (derived in Section B.2.) The integration is performed over the same time interval as the experiment, with a weighting of $1/t$ in order to treat each decade in time equally. Using the adjoint sensitivity analysis capabilities of the CVODES package, we are able to compute the gradient of Eq. 3.1 with respect to the reaction rates to enable rapid minimization of the objective function using a gradient based optimization algorithm.

The rate constants are derived from single exponential fits to the kinetic data. This exponential rate is converted into a second-order rate constant by assuming that the protein concentration remains constant over the assembly process. At 50% excess, this is a poor approximation and the converted second-order rate constant will not be measuring the binding rate directly for the secondary and tertiary proteins, but instead a composite rate that includes the time for the dependent proteins to bind. We will use local optimization from these initial values using the L-BFGS method[117] informed with true gradient information from CVODES to find proper second-order rate constants for these reactions.

3.2.3 Reduction of kinetic model

To increase the speed of our whole-cell simulations, the assembly network must be pruned of species which do not contribute significantly to the assembly process. This is accomplished by iteratively removing the species, s, which contributes the least to the total amount of 30S assembled. This contribution is quantified as the total reaction flux consuming that species, $\mathscr{F}_s$, which is computed from the integral

$$\mathscr{F}_s = \sum_{r \in \mathscr{R}_s} \int_{T_0}^{T_1} \mathrm{d}t \, k_r [\mathrm{P}_r][\mathrm{I}_s], \tag{3.3}$$

where the summation is over all reactions consuming species, s. The quality of the reduced low temperature model is monitored by computing the root mean square (RMS) error of the protein binding curves between the initial and modified networks. The modified network with the minimal number of intermediates not exceeding the error tolerance of 2×10^{-2} is accepted. Due to limited data available for the high temperature model, we instead monitor the difference in free protein half-lives between the reduced and unmodified models and accept the smallest network which does not exceed an average of 6% $\log_{10}$ difference in half-lives.

3.2.4 Construction of ribosomal biogenesis network *in vivo*

The *in vivo* biogenesis model consists of the assembly network determined from the *in vitro* data at 40 °C as well as transcription, translation, mRNA degradation, and dilution reactions, along with the cellular geometry and diffusion constants for all species. Transcription is modeled as a first-order birth process, where RNA production is localized at points in the cell representing their originating operon in the genome. The rates of the mRNA and rRNA birth processes are tuned to an intended expression level, with no gene regulation included in the model. RNA is produced from nine r-protein and seven rRNA operons placed throughout the cell according to their genomic position. Assembly of the large subunit is not included in this model. Instead, the LSU is introduced into the system as a zeroth-order birth process which creates LSU species uniformly throughout the cell at a rate matching 16S rRNA expression to ensure that the 30S and 50S copy numbers remain balanced.

The rates for translation depend on the operon structure taken from *E. coli* K-12 MG1655 genome (accession number: U00096[118]; genomic data processed using Biopython[119].) Translation elongation is modeled by a series of reactions. Each reaction represents the combination of the formation of a r-protein associated and the advancement of the ribosome along the transcript to the next r-protein gene. The transition rate between positions along the mRNA is simply the translation rate per nucleotide divided by the number of bases between the start of the protein created during this step and the beginning of the next protein to be produced (or the end of the transcript). The lengths of intervening genes which code for proteins not included in the model are included in the genomic distance used to compute the transition rate. Rates of transcription from the operons considered in our model are chosen such that the proteins reach a realistic steady-state concentration. The values of parameters used the *in vivo* model are summarized in Table B.3. All parameter values are reported in Table B.3.

3.2.5 Spatially resolved simulations of *in vivo* biogenesis network

Spatially resolved chemical reaction trajectories are sampled from the solution to the RDME describing the *in vivo* network and cell geometry discretized onto a lattice. The RDME is

$$\begin{aligned}\frac{\mathrm{d}P(\boldsymbol{x},t)}{\mathrm{d}t} =& \sum_{\nu}^{V}\sum_{r}^{R}[-a_r(\boldsymbol{x}_\nu)P(\boldsymbol{x}_\nu,t) + a_r(\boldsymbol{x}_\nu - \boldsymbol{S}_r)P(\boldsymbol{x}_\nu - \boldsymbol{S}_r,t)] \\ &+ \sum_{\nu}^{V}\sum_{\xi}^{\pm\hat{i},\hat{j},\hat{k}}\sum_{\alpha}^{N}[-d_\nu^\alpha x_\nu^\alpha P(\boldsymbol{x},t) + d_{\nu+\xi}^\alpha(x_{\nu+\xi}^\alpha + 1)P(\boldsymbol{x} + 1_{\nu+\xi}^\alpha - 1_\nu^\alpha, t)], \qquad (3.4)\end{aligned}$$

where $P(\boldsymbol{x},t)$ is the probability distribution to find a configuration $\boldsymbol{x}$ at time t. The configuration vector $\boldsymbol{x}$ contains the number of species present at each individual lattice site. The first term in Eq. 3.4 describes the flow of probability between different copy number states at every lattice site. The reaction propensities $a_r(\boldsymbol{x}_\nu)$ give the transition probabilities for reaction r at site ν. The r row of the stoichiometry matrix $\boldsymbol{S}$ is the change in species counts when reaction r occurs. The second term describes the flow of probability due to diffusion between neighboring lattice sites, indexed by ξ. Here d_ν^α is the diffusive propensity for species α in volume ν to leave its lattice site. Lattice Microbes (LM)[29], a software package designed to simulate stochastic reaction-diffusion systems using the multi-particle diffusion RDME (MPD-RDME) algorithm[32,120,121], is used to sample trajectories from the solution to Eq. 3.4. This software is highly optimized to take advantage of general-purpose computing on graphics processing units on NVIDIA hardware allowing for simulation times reaching cell cycle timescales.

Since this is the most complex RDME model simulated by LM to date, modifications to the code base were necessary to increase the performance of models with many chemical species and reactions. The reaction kernel, responsible for selecting the reaction and performing the update of species counts at each time step, was replaced with programmatically generated code with all loops unrolled, and all constant factors to the propensity calculations replaced with immediate values. This leads to a speed up allowing for an hour of simulation time to complete within approximately 3 days.

LM v2.2 simulations were executed on the XK7 nodes of NCSA Blue Waters (AMD 6276 Interlagos / NVIDIA Tesla K20X graphics processing unit (GPU) accelerators using CUDA 6.5) for short trajectories (< 10 minutes) over 64 simultaneous replicates. Replicates covering an entire cell cycle were performed on a local machine (2× Intel Xeon CPU E5-2640 / 4× NVIDIA GeForce GTX 980 GPU using CUDA 6.5) allowing for 4 simultaneous replicates.

3.2.6 Molecular dynamics simulations of early intermediates[†]

Atomic models of the assembly intermediates are built using the crystal structure of the *E. coli* ribosomal SSU (PDB: 2I2P[122]. Proteins and nucleic acids are parameterized with the CHARMM36[123,124] force fields. All systems are prepared using the protocol described in Section B.3. Systems are neutralized with sodium ions. A total of 840 ns of MD simulation on the 16S intermediates are reported.

Production runs are conducted using NAMD 2.10[125] under the NPT ensemble at 1 atm and 300 K. Periodic boundary conditions are applied, and a 1 fs - 2 fs - 4 fs multiple time-stepping approach was used. Long range interactions are calculated using particle mesh Ewald with 10 Å switching/12 Å cutoffs. Each run uses approximately 40 000 node-hours on NCSA Blue Waters's XE6 nodes (2× AMD 6276 Interlagos).

3.3 Results

3.3.1 Modeling *in vitro* small subunit assembly

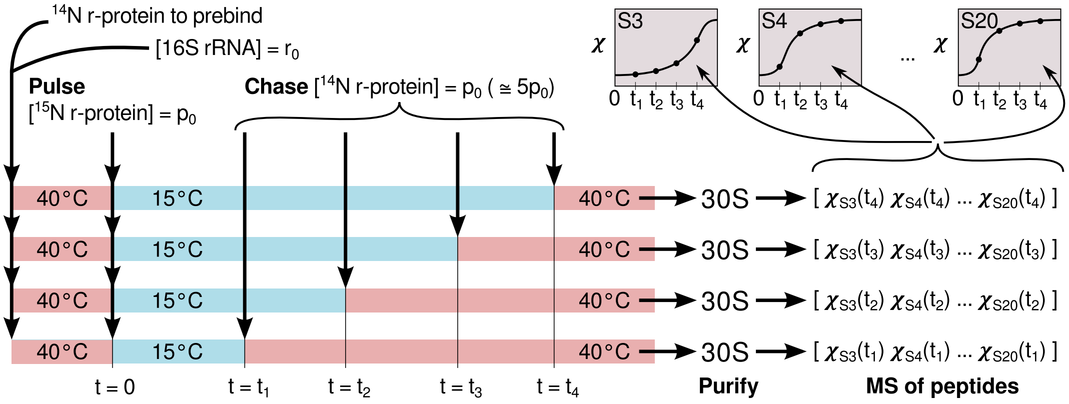
Construction of *in vitro* low temperature kinetic model of SSU assembly

The assembly process of the *E. coli* small subunit can be described by a network of binding reactions of the 21 r-proteins to the 16S rRNA and subsequent assembly products. We are omitting

[†] Molecular Dynamics simulations and their analyses were performed by Jonathan Lai.

bS1 in this model it is not an integral part of the mature 30S particle, and uS2 and bS21 due to the lack of kinetic data, owing to their transient binding nature. We have adopted nomenclature for the r-proteins which emphasizes their homology or lack there of between the three domains of life[97]. Because bS6 and bS18 form a stable heterodimer in solution[126], they are treated singly as the dimer bS6:bS18 in all the binding reactions, and is assumed to have already formed. The naïve assumption is that these proteins can bind in any order. If this is the case, then the network will include 2^{17} (10^5) species and 17! (10^{14}) reactions. To reduce this complexity, the Nomura map of thermodynamic dependencies among r-protein[101] is used to determine under which circumstances a protein can bind to an intermediate. Imposing this requirement leads to 1612 SSU assembly intermediates and 6997 reactions.

Initially, the rate constants are taken from a P/C qMS study of the reconstitution of the SSU *in vitro*[107]. Curves tracking the progress of r-protein binding to assembly intermediates were measured starting with no proteins bound initially (control experiment) and to various r-protein/16S intermediate configurations, i.e. prebinding experiments (Figure 3.2). From single-exponential curves fit to these data, an initial rate constant is approximated by assuming the exponential rate is a pseudo-first-order rate constant and converting to a proper second-order rate constant using the initial protein concentration. The rates are chosen from the prebinding experiments where the protein binds directly without requiring any dependent proteins to be present. This study revealed that the rates for several protein binding reactions are significantly increased for initial intermediates configured with proteins for which the binding protein is not thermodynamically dependent. These situations are referred to as kinetic cooperativity to differentiate the phenomenon from the thermodynamic cooperativity observed by Nomura[101]. For binding reactions exhibiting kinetic cooperativity, an ancillary rate constant is used to take this behavior into account. New rates are only introduced if there is a 2× or greater difference compared to the slowest rate observed for binding of that protein. This criterion ensures that the general character of kinetic cooperativity is represented in the model while minimizing the set of unnecessary parameters. A summary of the fold increases due to this phenomenon is provided Table B.1 for all P/C qMS experiments used in this model.



^{14}N r-protein to prebind
[16S rRNA] = r_0
Pulse
[^{15}N r-protein] = p_0
Chase [^{14}N r-protein] = p_0 (≅ 5p_0)
S3
S4
S20
χ
0 t_1 t_2 t_3 t_4
...
40°C
15°C
30S
[$\chi_{S3}(t_4)$ $\chi_{S4}(t_4)$ ... $\chi_{S20}(t_4)$]
[$\chi_{S3}(t_3)$ $\chi_{S4}(t_3)$ ... $\chi_{S20}(t_3)$]
[$\chi_{S3}(t_2)$ $\chi_{S4}(t_2)$ ... $\chi_{S20}(t_2)$]
[$\chi_{S3}(t_1)$ $\chi_{S4}(t_1)$ ... $\chi_{S20}(t_1)$]
t = 0
t = t_1
t = t_2
t = t_3
t = t_4
Purify
MS of peptides

Figure 3.2 Schematic of P/C qMS experiments. The prebinding intermediate is constructed initially from rRNA and the initial set of unlabeled r-proteins by incubation at 40°C. The labeled proteins are added and incubated at 15°C until the chase of 5× molar excess of unlabeled proteins is added. This is incubated at 40°C again to allow all binding to complete. The 30S particles are purified and mass spectrometry is used to analyze the fraction of labeled proteins, χ, for all r-proteins simultaneously. This process is performed many times to build up the pulse/chase curves.

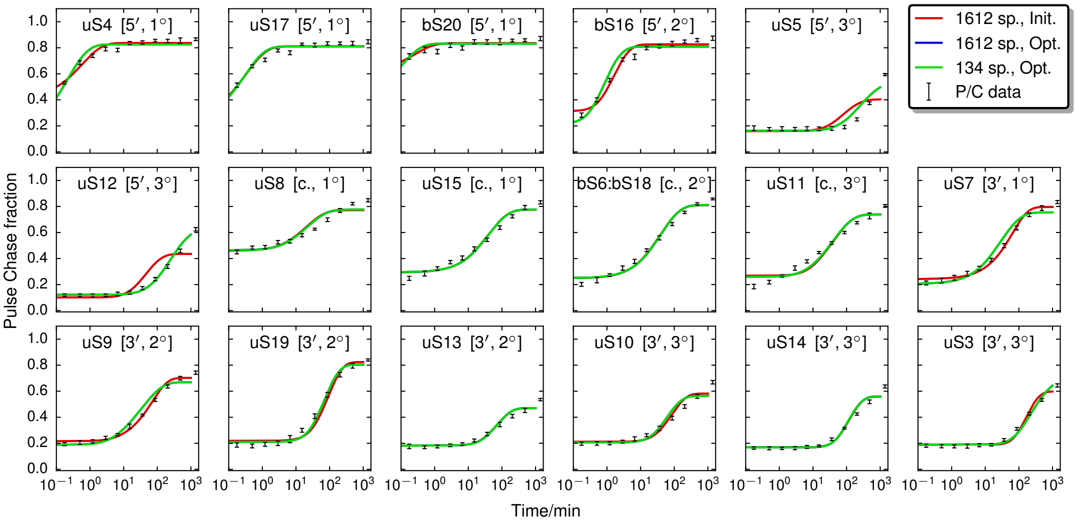


Figure 3.3 Comparison of experimental P/C qMS measurements of ribosome assembly starting with bare 16S rRNA (error bars) to the 15 °C model (solid curves). Raw concentration data from the model is transformed into an idealized pulse/chase curve assuming the same ratios of labeled to unlabeled species used in the experiments[107]. Using the rates estimated by fitting to the experimental curves yields the red curve. Improvement on this curve is made by optimizing the model parameters over P/C qMS experiments starting with nine different initial intermediates (Figure B.2). By reducing the intermediate count from 1612 to 134 by removing the least important intermediates, a simplified model (green curve) is generated which quantitatively matches the full model.

The proteins uS3, uS5, bS6:bS18, uS11, uS12, uS14, and bS16 show no significant kinetic cooperativity. In this model each of these proteins bind to allowed intermediates at a rate independent of the intermediate composition. All other proteins bind using some manner of kinetic cooperativity. The rate rules for assigning parameters to reactions are derived by considering the kinetic data for each protein individually. When all rules fail to apply to a reaction, a default rate is used. This rate is chosen from the prebinding experiment in which the initial intermediate satisfies all of the dependencies with the least total number of protein bound.

The most significant examples of kinetic cooperativity were observed in binding to the 3′ domain. For uS9, its binding rate is increased by over 200× if the intermediate it binds to includes uS19 (and uS7 from Nomura dependencies). The minimum rate was observed for binding to the intermediate with all primary proteins prebound. If uS7 is present alone, the rate is 20× the minimum, however if uS7 and uS13 are present as well, the rate drops to 4×. Finally if all 5′ and central domain proteins and uS7 are prebound, the rate is 5× the minimum rate, implying that some or all of the secondary and tertiary proteins binding to the 5′ and central domains increase the binding efficiency. Assuming that the effect of uS19 is dominant, the rate rule list for uS9 is developed by first testing for the presence of uS19 ignoring any other non-dependent species to uS9, such as the 5′ domain and central domain proteins. Each rule defines a new rate parameter for the model. The value of this parameter is taken from the prebinding study that the rate originates from. Second, the presence of uS13 is tested since this appears to decrease the binding efficiency compared to the case of uS7 bound alone. Third, the presence of all primary and secondary 5′ and central domain proteins is tested for, ignoring the tertiary proteins. Fourth, the presence of all primary binding proteins is tested and finally the default rate is chosen to be from the uS7 prebinding experiment since this prebinding intermediate minimally satisfies the thermodynamic dependencies for uS9. The parameter assignment rules are developed similarly for all other proteins. A summary of the 32 parameters and their rules are provided in Table 3.1. This method gives rise to an enormous reduction of the parameter space dimensionality, leading to 15 parameters describing kinetic cooperativity, and 17 default rates. Since we are fitting to 107 curves which are all parameterized by a single rate constant, overfitting of the model is not a concern.

The initial conditions are chosen to match the experimental conditions used in the P/C qMS experiments: 0.305 μM of 16S rRNA and 0.458 μM of each r-protein. The model is integrated from 6 seconds to 2000 minutes. Figure 3.3 (red curve) compares the protein binding curves from the model to the control prebinding experiment. The experimental pulse/chase curves do not compare directly to the simulated ideal pulse/chase curves since experimentally the reactions are not 100% efficient. To correct for this a linear transformation is applied to the simulated data to match the starting and ending fraction of the experimentally measured curves. To compute the initial second-order rate constants, a single exponential is fit to the experimental assembly progress curves for the proteins and experiments referenced in Table 3.1. The exponential rate from this fit is then used to compute a second-order rate constant assuming pseudo-first-order conditions with constant protein concentration. This is not a necessarily a good approximation in this situation, however it is sufficient to compute an initial parameter set to perform a local optimization.

Optimization of Assembly Parameters and Kinetic Rules

Since there is some variability between rates taken from different experiments and our initial rates were derived using a pseudo-first-order approximation, it is justified to perform optimization on our network to tune the parameters toward a better fit. Biologically reasonable limits on the parameter space were used: 4×10^{-6} $\mu M^{-1} s^{-1}$ for the lower limit which corresponds to a reaction timescale an order of magnitude larger than the duration of the P/C qMS experiments, and 3.5×10^{3} $\mu M^{-1} s^{-1}$ for the upper limit corresponding to the fastest diffusion limited association of r-protein to the 16S rRNA. By minimizing Eq. 3.1, we reduced the MSE between the P/C qMS experiments and our model to 6.5% of the error computed from the initial rates (Figure 3.3, blue curve.) The majority of parameters change within an order of magnitude or less, however significant deviations in the parameters for uS3 and uS5 between the estimated and optimized rates were observed.

Table 3.1 Assembly rate constants for the *in vitro* ribosome biogenesis kinetic model at 15°C.

					Rate (μM^{-1}s^{-1})		Rules	
	Protein	Symbol	# Rxn.	Experiment	Initial	Optimized	Present	Absent
5′ domain	uS4	$k_{4,\text{o7}}$	32	uS7	1.713×10^{-1}	2.918×10^{-1}	uS7	uS9, uS13, or uS19
		$k_{4,\text{def}}$	512	Control	8.383×10^{-2}	2.173×10^{-1}		
	uS17	$k_{17,13\text{o}19}$	120	uS7 & uS19	5.285×10^{-2}	1.152×10^{-1}	uS13 or uS19	uS9
		$k_{17,\text{def}}$	560	Control	1.421×10^{-1}	1.614×10^{-1}		
	bS20	$k_{20,7}$	32	uS7	4.483×10^{-1}	9.325×10^{-1}	uS7	uS9, uS13, or uS19
		$k_{20,\text{def}}$	512	Control	2.005×10^{-1}	4.968×10^{-1}		
	bS16	$k_{16,\text{def}}$	272	1°	5.103×10^{-2}	7.655×10^{-2}		
	uS5	$k_{5,\text{def}}$	136	1° & 2°	7.29×10^{-4}	1.701×10^{-4}		
	uS12	$k_{12,\text{def}}$	160	1° & 2°	1.895×10^{-3}	1.806×10^{-4}		
central domain	uS8	$k_{8,7\text{r}9}$	120	uS7 & uS9	2.223×10^{-2}	3.419×10^{-2}	uS7 or uS9	uS13 or uS19
		$k_{8,13}$	320	uS7 & uS13	6.488×10^{-3}	3.429×10^{-3}	uS13	
		$k_{8,\text{def}}$	240	Control	1.531×10^{-3}	4.52×10^{-4}		
	uS15	$k_{15,13\text{o}19}$	92	uS7 & uS13	5.176×10^{-4}	MIN	uS7, uS13, or uS19	uS9
		$k_{15,\text{def}}$	311	Control	1.276×10^{-3}	1.265×10^{-3}		
	bS6:bS18	$k_{6,\text{def}}$	403	1°	1.257×10^{-1}	2.89×10^{-1}		
	uS11	$k_{11,\text{def}}$	403	1°	1.166×10^{-2}	2.441×10^{-2}		
3′ domain	uS7	$k_{7,5\text{c}}$	1	5′ & cent.	2.333×10^{-3}	5.146×10^{-3}	5′ and cent.	
		$k_{7,\text{def}}$	91	Control	7.654×10^{-4}	1.665×10^{-3}		
	uS9	$k_{9,19}$	184	uS7 & uS19	1.786×10^{-1}	4.456×10^{-1}	uS19	
		$k_{9,13}$	92	uS7 & uS13	2.989×10^{-3}	3.007×10^{-3}	uS13	
		$k_{9,5\text{c}}$	8	5′, cent., & uS7	4.374×10^{-3}	1.027×10^{-3}	1° & 2° of 5′ and cent.	
		$k_{9,\text{pri}}$	7	1°	8.019×10^{-4}	MIN	1°	
		$k_{9,\text{def}}$	77	uS7	1.713×10^{-2}	2.572×10^{-2}		
	uS13	$k_{13,19}$	476	uS7 & uS19	1.13×10^{-1}	1.134×10^{-1}	uS19	
		$k_{13,\text{pri}}$	51	1°	2.187×10^{-3}	MIN	1°	
		$k_{13,\text{def}}$	233	uS7	4.009×10^{-4}	MIN		
	uS19	$k_{19,\text{pri}}$	102	1°	1.713×10^{-3}	1.838×10^{-3}	1°	
		$k_{19,\text{def}}$	466	uS7	1.093×10^{-3}	5.718×10^{-4}		
	uS3	$k_{3,\text{def}}$	48	1° & 2°	1.13×10^{-3}	3.703×10^{-2}		
	uS10	$k_{10,19}$	368	uS7 & uS19	4.592×10^{-2}	6.584×10^{-2}	uS19	
		$k_{10,\text{def}}$	184	uS7 & uS9	4.738×10^{-4}	4.008×10^{-4}		
	uS14	$k_{14,\text{def}}$	384	1° & 2°	1.749×10^{-3}	1.173×10^{-3}		

The 32 parameters in the ribosome assembly kinetic model shown are separated by domains and listed in decreasing rule precedence. The initial reaction rate constants are estimated from Bunner et al.[107] and the final reaction rate from global optimization are shown for each parameter. The parameters are sorted by decreasing rule precedence. If an intermediate does not satisfy the rules for a parameter (presence or absence of certain r-proteins), the next parameter in the list is tested. MIN indicates that the local optimizer has driven this parameter to the lower limit of 4×10^{-6} μM^{-1}s^{-1}.

Analysis of Low Temperature Binding Reaction Network

To gain a better understanding of the core of the binding reaction network, we simplified the full kinetic model by eliminating species with the smallest contribution to the overall integrated flux (Eq. 3.3) through the assembly network. The network was reduced from 1612 species to 134 species. Using a simple mean-square error metric, the protein binding curves of the reduced network match that of the full network with an average error of 1.8×10^{-2} (Figure 3.3, green curve.) With the network thinned out, one can readily visualize the distribution of reaction fluxes by drawing a network diagram (Figure 3.4) where the thickness of each edge from intermediate A to B represents the integrated fluxes or, equivalently, total amount species A converted to B over the entire assembly time (summand of Eq. 3.3.)

To discuss individual assembly intermediates, we must first develop a concise nomenclature to uniquely specify its protein/rRNA configuration. The states are labeled by the symbol $\{xyz\colon s_i, s_j, \ldots, s_k\}$, which consists of two parts. The first part indicates the level of completion of the $5'$ domain (x), central domain (y), and $3'$ domain (z) respectively. The letters here are placeholders for integers that indicate that all not all primary proteins are bound to that domain (0), all primary proteins bound (1), all primary and secondary proteins bound (2), or all proteins for that domain are bound (3). The second term indicates the specific proteins bound in the intermediate which were not included in the first domain label. For example, $\{000\colon 4\}$ describes the 16S rRNA with the only primary $5'$ domain proteins uS4 bound, and $\{100\}$ describes the state with all primary $5'$ domain proteins, uS4, uS17, and bS20, present.

A dominant pathway emerges from the reduced network diagram (Figure 3.4) where the 30S is assembled in the order $5' \to \text{central} \to 3'$. This result confirms the observed $5'$ to $3'$ binding order seen in experiments[105,127–129]. This main pathway contains intermediates seen in cryo-EM maps of *in vitro* SSU assembly at higher temperatures: $\{100\}$; $\{232\}$; $\{232\colon 5, 10, 14\}$; $\{233\colon 5\}$; and $\{332\colon 10, 14\}$[108]. With the exception of $\{100\}$, these intermediates are all found late in the assembly process. An ensemble of binding order sequences can be constructed through random walks over the network using the amount of intermediate converted to weight the transition probabilities. These sequences cluster well into two classes. The first cluster is associated with the dominant

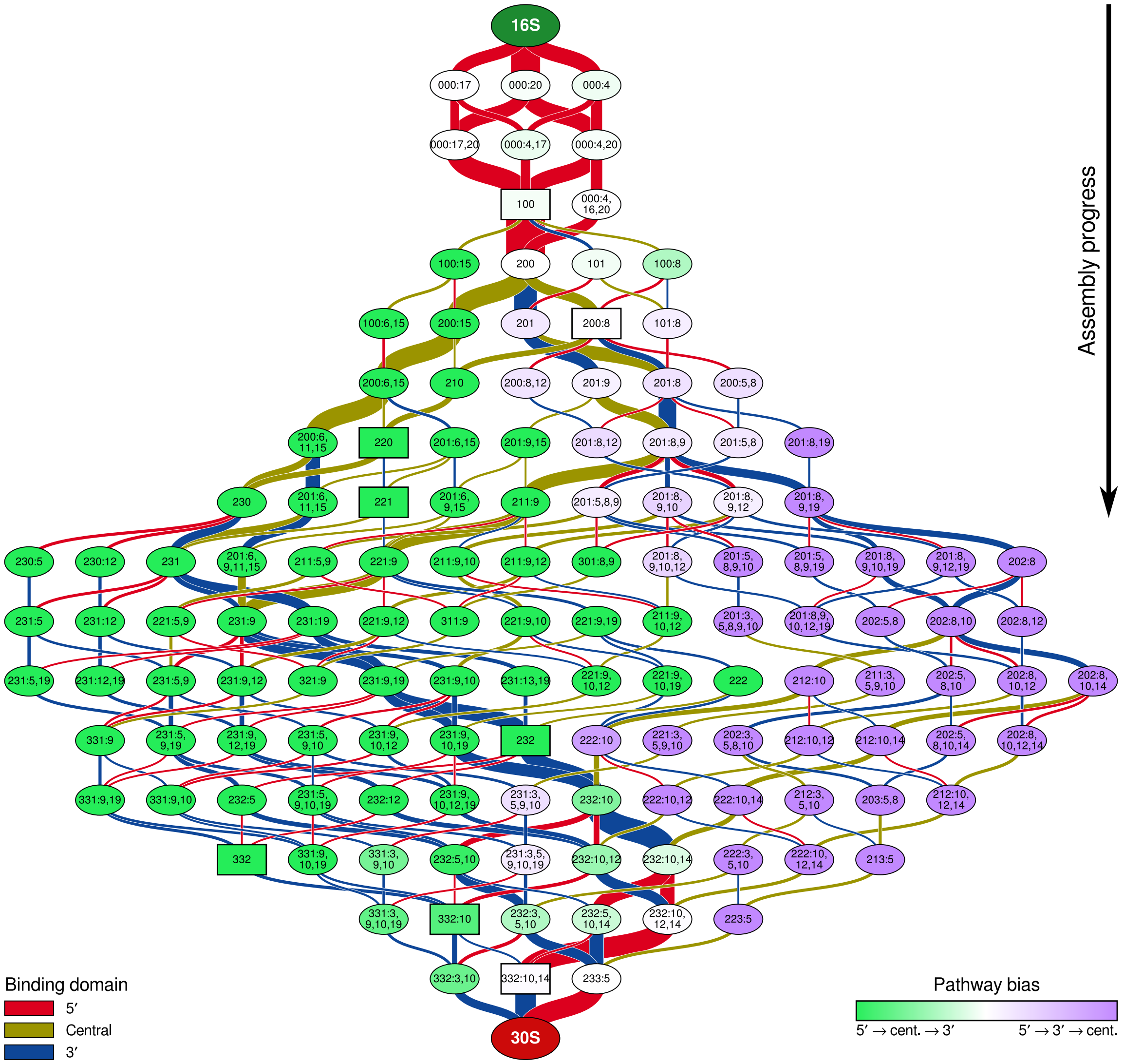


Figure 3.4 Reduced network for 30S assembly at 15 °C. Each node is an assembly intermediate, labeled according to which proteins are bound. A three digit number describes the set of r-proteins bound to each domain (5′-, central-, and 3′- respectively), and all remaining r-proteins are listed after the three digit number. The edges connecting the intermediates represent the r-protein binding reactions. The width represents the total amount of intermediate converted by that reaction, and the color indicates the binding domain of that protein (5′-red, central-yellow, and 3′-blue.) The color of each node indicates its bias toward its use of the two assembly pathways. Green indicates that clustering of protein binding order trajectories have indicated that this species is more likely to take part in the 5′ → central → 3′ pathway. Predicted assembly intermediates from P/C qMS and cryo-EM[108] are represented using rectangles.

$5' \rightarrow$ central $\rightarrow 3'$ ordering and contributes 70% of the total reaction flux. The other appears to assemble in a general $5' \rightarrow 3' \rightarrow$ central binding sequence and contributes the remaining 30%.

Both binding order clusters start out by binding all of the primary and secondary r-protein in the 5′ domain, forming {200}. This intermediate is the bifurcation point in which both assembly pathways begin to diverge The majority of trajectories from the major pathway complete the central domain before starting the 3′ domain, however the minor pathway switches between binding 5′ and central domain proteins until it reaches {201: 8}. This is another branch point in which the minor path can either rejoin the major pathway, or continue finishing the 3′ domain. With the exception of {200: 8}, no intermediates predicted using cryo-EM and P/C qMS are present on the minor pathway. State {200: 8} feeds about half of the reaction flux from that species back into the major pathway. The majority of the remaining flux ends up at {201: 8,9}, from which half of the flux flows back to the major pathway as well. Though the clustering analysis identified {200: 8} as a minor pathway species, it contributes equally to each path. Finally, both pathways converge in the vicinity of {232: 10}, from which the remaining tertiary 5′ and 3′ domain proteins bind to complete the 30S.

MD Simulations to probe network bifurcation and structural barriers at 15 °C

The minor pathway in the kinetic model has not been experimentally observed; however, the proteins bound to the *in vitro* {100} and {200: 8}, appearing before and after the bifurcation point, have been predicted using cryo-EM and P/C qMS[108]. Using MD simulations, we probed the ensemble of conformations of {201}, {200: 8}, and {200: 15} near the bifurcation point at {200} (Table B.2). All states contain the intact 16S rRNA and are prebound with uS4, uS17, bS20, and bS16 while {201}, {200: 8}, and {200: 15} have in addition uS7, uS8, and uS15 bound respectively. To observe the maximum fluctuations in the nucleic acid conformations, we prepared the MD simulations with a neutralizing concentration of sodium ions with no magnesium ions present.

In our previous MD simulations and experiments[104,130,131] on the motions of the 5′ domain under similar conditions, we saw that the dominant role of uS4 in {100} and {200} is to bring together helices h16 and h18, while r-proteins uS17, bS20, and bS16 tighten helices in their

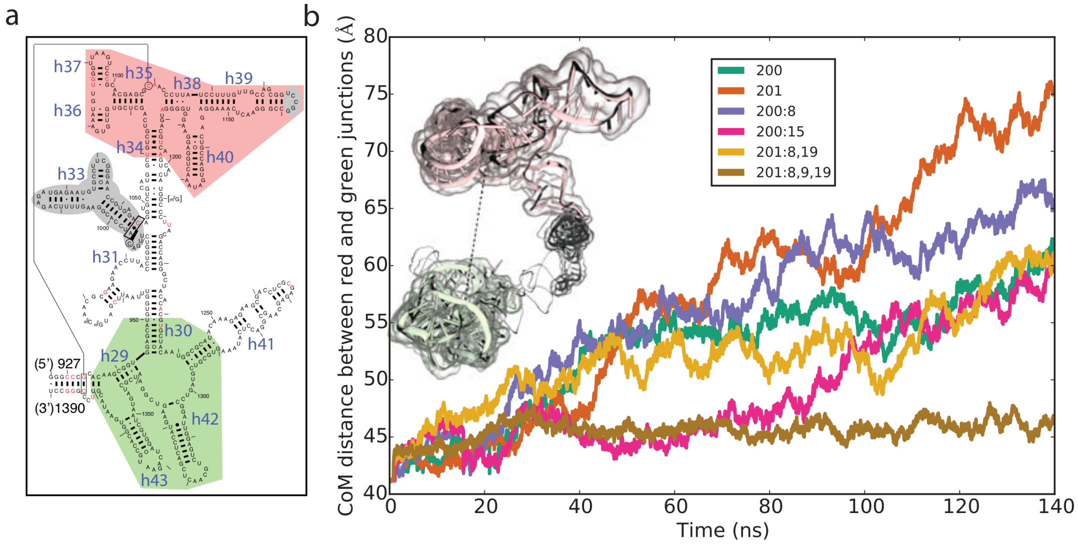


Figure 3.5 (a) Secondary structure diagram of the 3′ domain[132]. Center of masses are computed from the lower four-way junction helices h29, h30, h41–h43 (green region) and the upper three-way junction helices h34–h40 (red region). These centers are separated by the structural signature h33 (gray region)[96]. (b) Time traces of center of mass distances in the 3′ domain. The r-protein binding sites in the folded small subunit for each domain are provided in Figure B.3 and Figure B.3.

binding sites on the 5′ and central domain. Because the central domain is already partially formed in {200}, it is expected that the main role of uS8 and uS15 is to add rigidity to the central domain. uS7 binds to the partially formed 3′ domain while uS8, and uS15 binds to regions in the central domain already formed (see Figure B.3 and Figure B.3.)

In the 3′ domain, all four simulations showed similar motions. These fluctuations are dominated by the partial unfolding of the 3′ domain. Helices in the lower four-way junction (h29, h30, h41–h43) separate from helices in the upper three-way junction (h34–h40) (Figure 3.5a). Time traces of the center of masses for the different junctions in all four MD simulations show that the helices separate from 40 Å to over 60 Å after 140 ns (Figure 3.5b). Simultaneously, the structural signature[96] h33 separates from h31 and h32 and becomes more solvent exposed. This is expected since h33 is connected to these junctions. Similar results are seen in simulations of the *Thermus thermophilus* small subunit (Figure B.1), suggesting that these motions are probably common to all bacterial organisms. The fact that states {200}, {201}, {200: 8}, and {200: 15} all have similar motions suggests that there is no strong bias to binding either uS7, uS8, or uS15 and that the next major assembly barrier, the opening of the $3'$ domain, occurs further along in the assembly pathway.

Because the binding of uS7 and uS8 have a minimal effect globally on the structure of the ribosome assembly intermediates, we probed the effect of adding the $3'$ domain binding r-proteins uS9 and uS19. In the folded SSU, uS9 binds to both the lower four-way and upper three-way junction while uS19 binds to the structural signature h33 (Figure B.3). As the uS19 binding site is more local than uS9, we investigated the binding of uS19 first (Figure B.3). Adding uS19 to the simulations (moving from state $\{200\colon 8\}$ to $\{201\colon 8, 19\}$), tightens the structural signature in h33 and keeps h33 packed against h31–h32 and like the four previous simulations, $\{201\colon 8, 19\}$ also shows similar unfolding of the $3'$ domain (Figure 3.5b). State $\{201\colon 8, 9, 19\}$, on the other hand, does not have the separation in the $3'$ domain (Figure 3.5b). Interestingly, all six MD simulations showed the $3'$ domain rotating away from the five-way junction in the $5'$ domain, suggesting that there is another folding barrier further along in the assembly pathway. This motion might only be arrested upon the addition of uS5.

Construction of *in vitro* high temperature kinetic model of SSU assembly

The previously described model fits the experimental data well over many different initial intermediate configurations and has predictive power, however it is not adequate for use in a *in vivo* model of *E. coli* since it describes the reconstitution of the 30S at a temperatures much lower than that required for optimal *E. coli* growth. Since the rates of binding for each protein will vary independently with temperature in ways that are difficult to predict, it is not sufficient to simply scale the rates of the low temperature model to match the observed assembly time *in vivo*. To prepare a kinetic model of SSU assembly at physiologically optimal temperatures, we constructed a model based on *in vitro* reconstitution experiments performed at 40 °C[108]. These experiments were performed at lower concentrations than the low temperature model: 0.02 μM 16S rRNA and 0.04 μM labeled r-proteins, however the 5× molar excess of the chase unlabeled proteins was the same as before. Since only the control protein binding curves were measured in this work, we are not able to include the effect of cooperative binding. Due to the lack of these reactions, the high

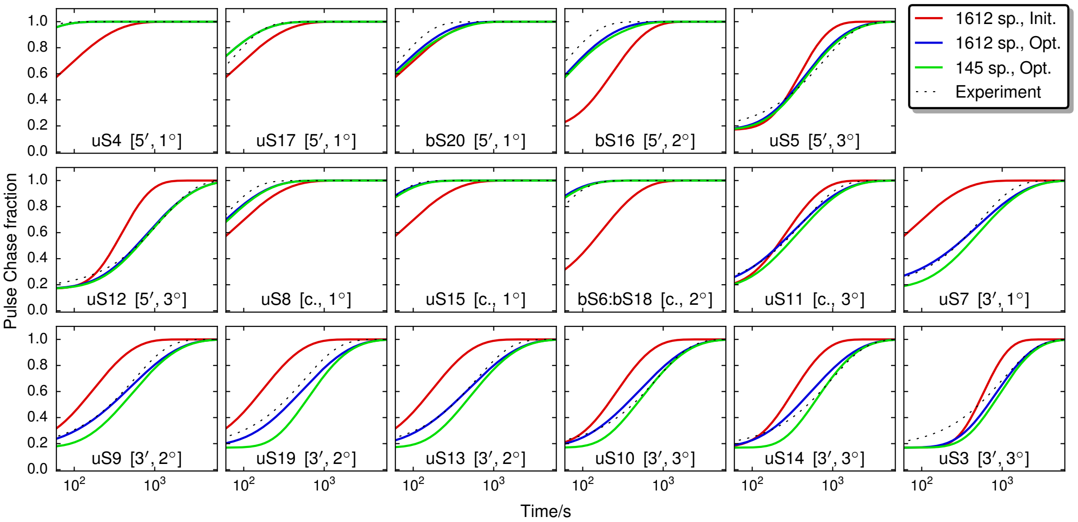


Figure 3.6 Fitting of protein binding curves from the high temperature *in vitro* model to the curves measured from the 40 °C experiment. Deviations of the reduced model with respect to the full model tend to only impact the 3′ domain binding proteins.

temperature model does not fit to the experimental data as well as the low temperature model (Figure 3.6.) However, the correct protein binding order is represented and protein abundance half-lives are reproduced within 6%.

The reduced network assembly model at 40 °C contains 145 unique intermediates and 325 protein binding reactions. The number of intermediates was set to focus on the core binding network and to allow efficient RDME simulations of the *in vivo* model discussed below. While Figure 3.6 shows the reduced set captures the binding kinetics well, we carried out additional simulations to investigate whether important assembly pathways are being removed. Reducing the full high temperature model from 1612 to 638 states, we repeated the previous analysis of the assembly network. It was observed (data not shown) that there is a minor partitioning of protein binding order trajectories into the two pathways seen in the 15 °C data. However, the 5′ → central → 3′ trajectories occur greater than 90% of the time compared to the 70% seen in the low temperature network. The dominance of the 5′ → central → 3′ pathways is likely due to the effects of the higher temperature which increase the rates of binding in the primary proteins and diminishes the differences previously observed between the secondary and tertiary proteins.

Since the rate constants have changed significantly with respect to the low temperature model, the reduced network structure has as well. The problems we experienced with uS3 and uS5 were not repeated here since the experimental binding order of these proteins was consistent with the Nomura map. The assembly pathway is much less directed, i.e. for most states there are many binding reactions that occur at similar reaction rates (Figure 3.7). It is evident that the temperature has had a large effect on the utilization of assembly pathways. The bifurcation into two distinct pathways seen in the low temperature model is absent in the high temperature model (Figure 3.7). Though the binding order is less well-defined at higher temperatures, the assembly still progresses in a $5' \rightarrow$ central $\rightarrow 3'$ directionality, with the $5'$ and central domain proteins binding in parallel, followed by the $3'$ domain proteins, and finally the remaining tertiary proteins from the $5'$ domain.

Binding of the primary proteins uS4 and uS15 to the $5'$ and central domains respectively, dominate the nucleation of the nascent 30S. The most highly traversed intermediates seen at low temperatures, states $\{100\}$ and $\{200\}$, appear less prominent at high temperatures. State $\{100\}$ appears 1 minute into the assembly process in both the proposed mechanism[108] and our kinetic model. The state $\{220\}$ acts as a central hub for most assembly paths in our network and is also predicted as an intermediate in the proposed mechanism. It reaches its peak concentration at 2.2 minutes which is comparable to the time of 3 minutes inferred from P/C qMS and cryo-EM. The following state, $\{221\}$, appears in both our model and the predicted mechanism as well however the timings are different. It was predicted to bind 8 minutes into the assembly process, however we are observing the intermediate $\{221\}$ coming in about 6 seconds after $\{220\}$. The next predicted assembly intermediate is $\{232\}$ which is less prominent in our model than what would be expected from the P/C qMS and cryo-EM data. The maximum concentration of $\{232\}$ is reached much sooner than expected from the proposed mechanism, coming in at 5 minutes instead of 12 minutes as predicted. The latest predicted intermediate, $\{332: 10, 14\}$, which is missing only uS3, comes in at 20 minutes instead of 70 minutes as predicted. The timing discrepancies between the experiments and our results is likely due to the lack of kinetic cooperativity in our model. Though there are differences between these times, the P/C qMS study did not identify exact intermediates experimentally, instead they are inferred from the data. The relative ordering of intermediates is suggestive that this model and the published mechanism are in agreement.

3.3.2 Modeling *in vivo* ribosome biogenesis

Construction of the Ribosome Biogenesis Model

In addition to the hierarchical assembly of the SSU described above, the process of ribosome biogenesis in the cell must also include the transcription of rRNA and mRNA coding for r-proteins, the translation of r-protein, and the degradation of mRNA. The high temperature *in vitro* model of SSU assembly developed from kinetic experiments with well-mixed solutions of rRNA and r-proteins is now applied to biogenesis in the heterogeneous cellular environment. For the full ribosome biogenesis model, we control the birth rate of the LSU to match that of the SSU without explicitly including LSU assembly and include 70S formation and dissociation reactions with rates taken from the literature[133–135].

We present a spatially resolved model of the process in a simulation of a slow growing *E. coli* cell, of dimensions $4.0 \times 0.9 \times 0.9\ \mu\text{m}^3$, and initially containing approximately 3000 ribosomes[33,114]. Using our LM v2.2 we monitor the stochastic changes in the number of species over in a cell over its doubling time of 120 minutes. The capsule shaped cell is discretized onto a lattice with 32 nm spacing between lattice sites, allowing us to neglect excluded volume effects from the 20 nm diameter 70S particles. The nucleoid region of dimensions $3.1 \times 0.45 \times 0.45\ \mu\text{m}^3$ is centered within the cell volume (Figure 3.8a). At each lattice site, we assume the well-stirred approximation to evaluate the reaction time course using the Gillespie algorithm[8].

The protein diffusion constants are estimated based on their mass using a scaling relation between the diffusion constant in water versus cytosol[136] leading to diffusion constants in the range 8 – 20 $\mu\text{m}^2\,\text{s}^{-1}$. The maximum time step Δt that can be used in the MPD-RDME simulation is determined by the fastest diffusing species, which in this case is bS18. To ensure no particles diffuse more than a single lattice site per step, the maximum time step is chosen to ensure that the RMS displacement of a Brownian particle, $\sqrt{6D\Delta t}$, is shorter than the lattice spacing. In order to speed up the simulation, the protein diffusion constants were all scaled by a factor of 0.3 to

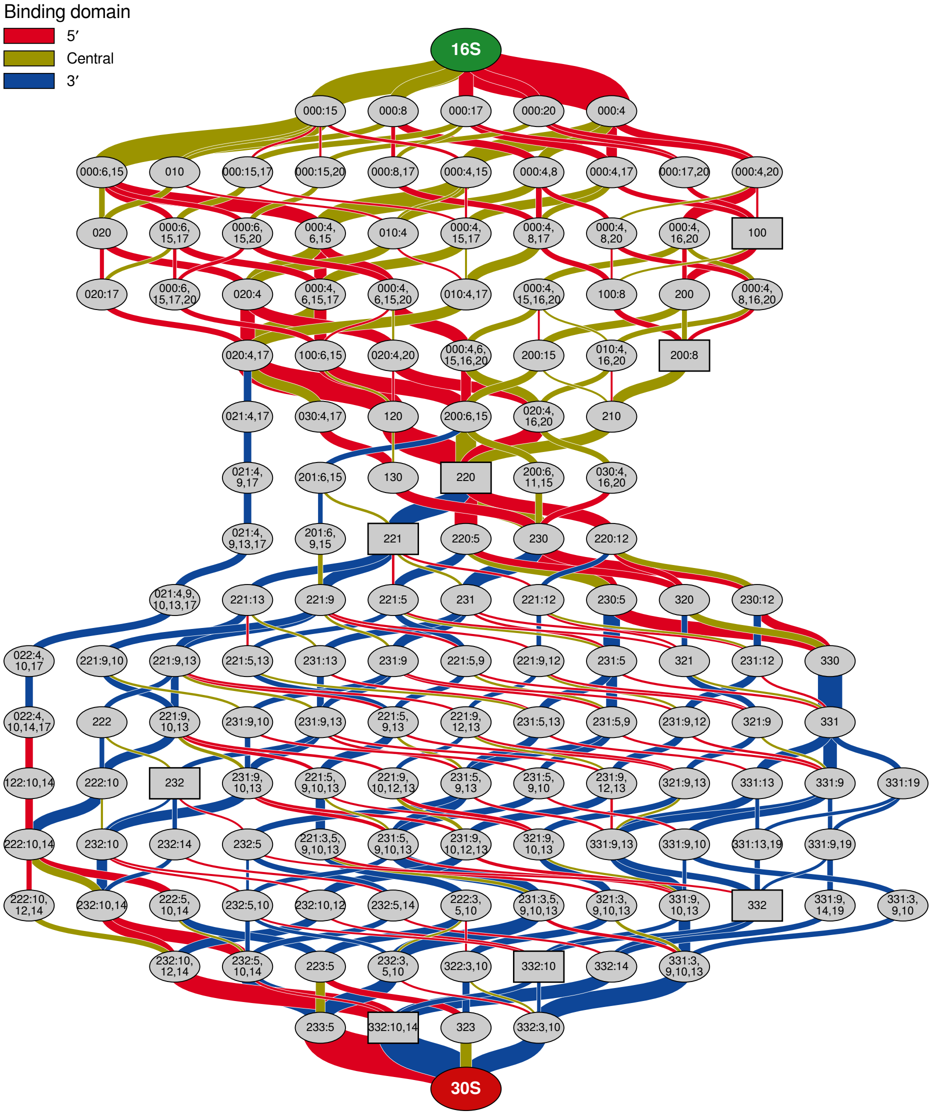


Figure 3.7 Reduced network for 30S assembly at 40 °C. The 5′ and central domain proteins bind simultaneously, leading to {220}. From here two weakly defined paths emerge: either the 5′ and central domains are completed simultaneously followed by the 3′ domain, or vice versa, ending in the formation of the 30S.

allow for longer time steps, resulting in a maximum time step of 25 μs. This should not have a significant effect on the outcome of the simulation since the slowest protein diffuses at a rate nearly an order of magnitude faster than the fastest non-protein species.

Messenger RNA diffuses at 0.3 $\mu m^2 s^{-1}$ as measured in the literature[137]. The diffusion constant for rRNA is computed from the radius of gyration[138] using the same scaling relationship to account for diffusion in cytosol as for r-protein. Assembly intermediate diffusion rates are assigned by counting the number of protein bound and using this number to linearly interpolate between the diffusion constants of 16S and 30S species. Transition rates between compartments are computed from the geometric mean of the diffusion rates for each compartment.

Single particle tracking experiments of individual small and large subunits as well as complete ribosomes have shown that ribosomes are partially excluded from the nucleoid region and diffuse at a rate 10× slower than individual subunits[113]. From this study, we take the rates of 0.4 $\mu m^2 s^{-1}$[113] for both SSU and LSU and 0.055 $\mu m^2 s^{-1}$[111,113] for full 70S ribosomes. We decrease the diffusion constant of ribosomes, ribosomal subunits, and assembly intermediates within the nucleoid region by a factor of 10× to account for the increase in molecular crowding due to the presence of a compacted chromosome. The 70S particles are observed to be partially excluded from the nucleoid region. The reason for this is not well-understood[111,113], however is most likely a result of the excluded volume interactions between the ribosomes and DNA. To account for ribosome exclusion without explicitly simulating the chromosome, we bias the transition rates between the nucleoid and cytoplasm by a factor of 4.0. A summary of the diffusion parameters are given in Table 3.3, and the complete list can be found in Table B.3.

The initial species counts (see Table B.4) are determined from the mean copy numbers at the steady state of a well-stirred stochastic simulation of the *in vivo* network within a volume equal to the cell (2.37 fl) using LM. The freely diffusing species are placed uniformly throughout the cell, the translating ribosomes are placed outside the nucleoid uniformly in the cytoplasm, and the operons are placed based on their genetic loci. These seven rRNA operons and nine r-protein operon species are placed in the nucleoid region at randomly about the central axis. Assuming that the origin of replication is at the center of the cell and the chromosome is linearly organized[139], operons are placed along the cell axis at positions relative to their distance from *oriC* (Figure 3.8b). Subsequent simulations are initialized from random time steps taken from a long running simulation approaching steady state (Figure 3.8c).

The next step towards a spatially resolved model of ribosomal biogenesis is to provide constant and balanced production of rRNA and r-protein through transcription, translation, and degradation in the cell. Transcription is modeled as a simple birth process localized at operon sites within the nucleoid region. Transcription of 16S rRNA occurs from seven ribosomal operons (rrnABCDEGH) at a birth rate resulting in a mean count of 4500 ribosomes at steady state. This number is chosen in order to approximate a cell which initially contains 3000 ribosomes immediately following cell division, and doubles to 6000 ribosomes over its 120-minute cell cycle. Transcription of messenger RNA from the nine r-protein operons is modeled similarly to rRNA. Since mRNA is actively degraded by RNase E at various rates depending on the content of the transcript, we use data from a genome wide microarray study of *E. coli* mRNA half-lives[140] to estimate the decay rate for each messenger species individually. In lieu of explicit gene regulation, we tune the mRNA birth rates such that the steady-state copy numbers are roughly equal for each r-protein species. Since the volume of the cell does not change in our simulations, dilution reactions (modeled as a first-order death process) are added to account for the effect of increasing cell volume as the cell grows. Dilution reactions in addition to the mRNA degradation reactions are added for all species with the exception of the operons. These reactions occur at a rate of $\ln 2/120\,\text{min.}$, approximating a slow growing cell with a doubling time of two hours.

Our model of transcription and translation takes the operon structure in the mRNA transcripts into account and allows for multiple gene products to be produced from a single mRNA molecule. Translation is modeled in three stages. First, initiation occurs by the association of the messenger and small subunit, followed by the association of the large subunit to this complex to form a translating ribosome. Since a model of LSU assembly has yet to be developed, we simply add 50S species to the system at a rate that matches the production rate of 30S small subunits. Second, translation of the ribosome along the mRNA strand is simulated by assuming that once a 50S species associates to the 30S/mRNA complex, the ribosome translates with a constant speed until it dissociates from the end of the transcript. Each SSU r-protein is made sequentially at a rate k_{tl}/N_i where k_{tl} is the translation rate per amino acid (10 aa/sec, estimated from Bremer and Dennis[99]) and N_i is the number of codons between the stop codon of the previous and current SSU r-protein genes, including the length of any intervening genes not represented in the model (e.g. LSU r-protein.) Genomic data is taken from the *E. coli* K-12 MG1655 genome (GenBank accession number: U00096[118].) Finally, termination occurs following translation past any remaining genes not considered in the model, by the simultaneous dissociation of the ribosome into mRNA, 30S, and 50S subunits. An example of the derivation of the translation reactions from genomic data is given in Section B.4 for the *spc* operon. No post-processing is assumed to occur for protein. However, bS6 and bS18 dimerize prior to associating with rRNA at an assumed rate of $1.0\mu\text{M}^{-1}\text{s}^{-1}$[141] and dissociate at a rate $8.7 \times 10^{-3}\ \text{s}^{-1}$ computed from the dissociation constant reported in Recht and Williamson[126]. A summary of the *in vivo* reactions, rate constants and diffusion parameters are presented in Table 3.2 and Table 3.3. All parameters are reported in Table B.3.

Simulation Results of the Ribosome Biogenesis Model

We start with the initial conditions derived from the steady-state well-stirred simulation. Since these initial conditions describe the mean of a growing cell—starting at 3000 ribosomes and ending at 6000 ribosomes—we scale all species counts by 2/3 in order to approximate the initial conditions of a newly divided cell. The initial rRNA and 30S intermediate counts are set to zero so that the birth of new ribosomes over the 120-minute cell cycle can be monitored. The first new 30S begin to appear after 17 seconds (Figure 3.8c), and the cell quickly reaches a stable-state bulk 30S production rate of 27 per minute (from slope of production line), with new SSU appearing uniformly within the cell. The production rate is accelerated with respect to the *in vitro* simulations and is due to the greater r-protein concentration in the *in vivo* simulations. The total ribosome count, using the sum of 30S, 30S:mRNA, and 70S particles, increases from 3000 to 6000 over the 120-minute cell doubling time. The assembly intermediate counts fluctuate significantly over the course of the cell cycle with a mean count of 9.7 and standard deviation of 3.8

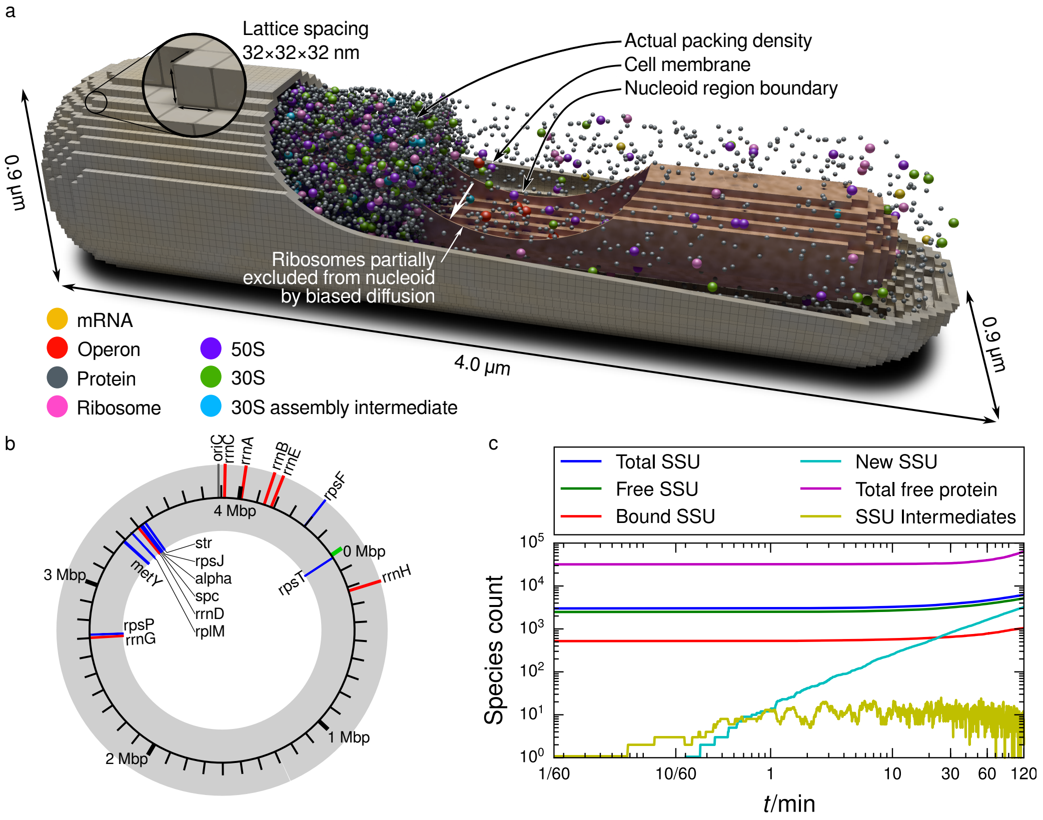


Figure 3.8 (a) Cut away of a representative simulated cell configuration. Operon locations (red) are fixed within the nucleoid region. Messengers (yellow) are transcribed from these sites and diffuse to find 30S particles (green) upon which a 50S subunit (purple) joins the complex forming a translating ribosome (pink). The ribosome emits r-protein (gray) which diffuse away and bind to small subunit intermediates (cyan). Translating 70S particles are excluded from the nucleoid region through a bias in their intercompartmental transition rates. (b) Genome diagram of the operons transcribed in the *in vivo* biogenesis model. (c) Species counts for a single replicate during a full 120-minute cell cycle. The initial species counts are set to their mean values from a well-stirred simulation at steady state. The counts of 16S rRNA and assembly intermediates are set to zero to investigate the formation of new intermediates. Dilution reactions are omitted from this simulation in order to investigate the change in particle count over a cell cycle. The curve "Bound SSU" measures the total count of 30S particles which are not bound to other species in the cell, i.e. all translating ribosomes and 30S/mRNA complexes. "Total SSU" measures all 30S particles in the cell, including both free species and bound.

(coefficient of variation: 0.39). All 145 intermediates appear with non-zero counts at some point during the cell cycle. Intermediate {233: 5} (30S missing uS12) had a maximum copy number of 12 which is greater than that of any other intermediate during the cell cycle. None of the other final intermediates (Figure 3.7) were found in such high quantities.

In order to gather more statistics on the formation times of the intermediates and new subunits, we designed simulations based on the previous cell-cycle long simulation (Figure 3.8c) to measure the delay between the appearance of rRNA and formation of intermediate species. Since the assembly time of the 30S is of the order of a few minutes, we performed 5 minutes of simulation time over 64 replicates to collect sufficient data to compute distributions of assembly times. The initial conditions for each replicate are selected from random time points during the cell cycle simulated previously, and have been modified to remove all assembly intermediates. The rRNA operons are removed and 100 rRNA molecules are distributed uniformly throughout the cell, allowing for the measurement of the time interval between the formation of 16S rRNA and the subsequent intermediates. Since the protein count (Figure 3.8c) is much higher than the initial rRNA count, the results from these simulations will be comparable to the full cell cycle. From these formation time simulations, we measure the birth times of the species of interest from the start of the simulation. The results of this process are equivalent to computing the species birth times by following the fate of each rRNA in the cell-cycle simulation.

To investigate the spatial distribution of assembly intermediates, we perform clustering in time to partition the set of intermediates into classes of species which are correlated in time. We use the data from the formation time simulations to compute mean copy number versus time curves for each intermediate. The curves from each intermediate are scaled to unit amplitude to treat each species equally with respect to its maximum concentration, and are compared using an RMS difference metric. Hierarchical clustering is used to partition the intermediates into 6 classes (T0–T5), where each class contains species which are formed at similar times. The fraction of the total rRNA that contributes to each temporal class (derived from the formation time simulations) is provided in Figure 3.8a, and the membership of all intermediates to each cluster is provided in Figure B.5. To achieve adequate sampling of the spatial distribution of all

Table 3.2 Summary of reactions and rate constants for the *in vivo* ribosome biogenesis model

Type	Reaction	Parameter values	Units	Compartments
Assembly	$\mathrm{I}_i + P_j \longrightarrow \mathrm{I}_{i+1}$ (1° prot.)	0.041 – 1.69	$\mu\mathrm{M}^{-1}\,\mathrm{s}^{-1}$	`cyt.`, `nuc.`
	$\mathrm{I}_i + P_j \longrightarrow \mathrm{I}_{i+1}$ (2° prot.)	0.24 – 31.	$\mu\mathrm{M}^{-1}\,\mathrm{s}^{-1}$	`cyt.`, `nuc.`
	$\mathrm{I}_i + P_j \longrightarrow \mathrm{I}_{i+1}$ (3° prot.)	0.025 – 1.75	$\mu\mathrm{M}^{-1}\,\mathrm{s}^{-1}$	`cyt.`, `nuc.`
Degradation	$\mathrm{mRNA}_i \longrightarrow \varnothing$	$1.0 \times 10^{-3} – 1.4 \times 10^{-3}$	s^{-1}	`cyt.`, `nuc.`
Dilution	$x \longrightarrow \varnothing$	9.6×10^{-5}	s^{-1}	`cyt.`, `nuc.`
Transcription	$\mathrm{DNA}_{\mathrm{rrn}X} \longrightarrow \mathrm{DNA}_{\mathrm{rrn}X} + 16\mathrm{S}$	0.062	s^{-1}	`nuc.`
	$\mathrm{DNA}_x \longrightarrow \mathrm{DNA}_x + \mathrm{mRNA}_x$	$4.9 \times 10^{-3} – 0.012$	s^{-1}	`nuc.`
Translation	$\mathrm{mRNA}_x + 30\mathrm{S} \longrightarrow \mathrm{Rib}^x_{\mathrm{init}}$	1.0×10^{2}	$\mu\mathrm{M}^{-1}\,\mathrm{s}^{-1}$	`cyt.`, `nuc.`
	$\mathrm{Rib}^x_{\mathrm{init}} + 50\mathrm{S} \longrightarrow \mathrm{Rib}^x_0$	3.0	$\mu\mathrm{M}^{-1}\,\mathrm{s}^{-1}$	`cyt.`, `nuc.`
	$\mathrm{Rib}^x_i \longrightarrow \mathrm{Rib}^x_{i+1} + \mathrm{P}_{x_i}$	0.019 – 0.27	s^{-1}	`cyt.`, `nuc.`
	$\mathrm{Rib}^x_{\mathrm{term}} \longrightarrow 30\mathrm{S} + 30\mathrm{S} + \mathrm{mRNA}_x$	0.015	s^{-1}	`cyt.`, `nuc.`
LSU birth	$\varnothing \longrightarrow 50\mathrm{S}$	3.1×10^{-4}	$\mu\mathrm{M}\,\mathrm{s}^{-1}$	`cyt.`, `nuc.`
Dimerization	$bS6 + bS18 \longrightarrow bS6\colon bS18$	1.0	$\mu\mathrm{M}^{-1}\,\mathrm{s}^{-1}$	`cyt.`, `nuc.`
	$bS6\colon bS18 \longrightarrow bS6 + bS18$	8.7×10^{-3}	s^{-1}	`cyt.`, `nuc.`

intermediates, we performed 128 short (5 minute) simulations from multiple starting conditions sampled randomly from the cell-cycle simulation. Using the temporal clustering, we computed mean intermediate distributions over the whole cell volume and projected the distribution onto the xz plane, leading to a qualitatively similar measurement of density as would be performed using an optical microscope.

The first class, designated T0, contains the 16S rRNA and 40 early intermediates and is formed at the sites of the rRNA operons. These intermediates are localized because the timescale of the protein binding reactions of the primary and secondary proteins of the 5′ and central domains are of the same order as the rRNA diffusion time (Figure 3.9b). In the next class, T1, the 3′ primary and secondary proteins uS7 and uS9 bind (Figure 3.9c), and the distribution of intermediates in this class begins to leave the nucleoid region. T2 contains the main bottleneck species 200, and includes intermediates as late as {220: 10}. Because of this, there is a path through the network which can skip over T3 entirely. T3 consists of less common intermediates undergoing the binding of 3′ domain proteins and later binding 5′ domain proteins. This is the last cluster where any spatial heterogeneity is evident. T4 consists of more common late stage intermediates

Table 3.3 Summary of diffusion constants for the *in vivo* ribosome biogenesis model.

Species	Compartment	$D/\mu\mathrm{m}^2\,\mathrm{s}^{-1}$
Ribosome	`cytoplasm`	0.055
	`nucleoid`	0.0055
	`cytoplasm` → `nucleoid`	0.0043
	`nucleoid` → `cytoplasm`	0.0017
Subunit	`cytoplasm`	0.4
	`nucleoid`	0.04
	`cytoplasm` ↔ `nucleoid`	0.126
Protein	`cytoplasm, nucleoid`	2.6 – 6.4
mRNA	`cytoplasm, nucleoid`	0.3
Intermediate	`cytoplasm`	0.15 – 0.39
	`nucleoid`	0.015 – 0.039
	`cytoplasm` ↔ `nucleoid`	0.047 – 0.122

Transition rates between compartments are computed from the geometric mean of their diffusion constants.

undergoing similar binding as T3. The distribution of T4 is effectively uniform over the cell. Finally, T5 contains species missing tertiary proteins and is distributed uniformly. This leads to production of new 30S occurring uniformly throughout the cell. The temporal class membership of all intermediates is given in Figure B.5.

The complex formed from the binding of mRNA to the SSU is found either in the cytoplasm or close to the messenger's originating operon. The mRNA cannot diffuse far from its originating transcription site because of the high concentration of 30S particles throughout the cell. Once the translating complex is formed by binding a 50S particle to the 30S/mRNA complex, the particle will diffuse out of the nucleoid. Its diffusion back into the nucleoid is hampered by the biased intercompartmental transition rates. Once translation completes, the 70S dissociates leaving 30S, 50S and mRNA species free outside the nucleoid region. This leads to a distribution where the 30S/mRNA binding events are localized around their originating operons and in the cytoplasm compartment. The termination of translation appears to occur almost entirely outside of the nucleoid region, since the translation process is slow enough to allow the ribosome to completely diffuse out of the nucleoid (Figure 3.9e.)

The mean assembly time for individual subunits was measured to be 30 seconds. The distribution of assembly times is approximately gamma distributed with a scale parameter of 2.35 seconds and a shape parameter of 0.208 (Figure 3.9d.) This mean assembly time is similar to the experimentally measured *in vivo* maturation time for the 30S of 1.3 – 3.5 minutes at a cell doubling time of 100 minutes[142].

Performance of LM software[‡]

To our knowledge our simplified model, with its 251 unique species and 1336 reaction (676 within the nucleoid region and 660 in the cytoplasm), is the largest time dependent simulation of of *in vivo* ribosome biogenesis to date. The cell model tests the limits of LM v2.2 with regard to its handling of the number of species and reactions. Two major data structures used by LM are the stoichiometric matrix, S, with dimensions of $N_{\text{reactions}} \times N_{\text{species}}$ and the reaction location matrix, RL with dimensions $N_{\text{reactions}} \times N_{\text{compartments}}$, specifying the reactions that can occur in a given compartment. Both of these structures are typically stored in GPU constant memory which is limited to 48 KB in size in most GPU. The size requirements of S and RL are 16 KB and 10 KB respectively, so for the species count required for the ribosome biogenesis model, only 64 reactions could have been supported. In LM v2.2, we added the functionality to relocate S and RL to GPU global memory and access them via the read-only data cache path added to the Kepler-class GPU. Current GPU constant memory usage now only handles the remaining data structures, allowing simulations of 2400 reactions without any additional changes.

The performance of the MPD-RDME simulations is determined by the wall-time required for particle diffusion, reaction evaluations, and handling of input/output and simulation overflows. The scaling of computational time of a single time step is consistent with the previous version developed for the multi-GPU simulations[29], where the evaluation of reactions is a linear time operation in the number of reactions since the reaction list must be traversed for every non-empty lattice site. Because of this, a single time step on Kepler class NVIDIA GPU (K20X; CUDA 6.5) on the NCSA Blue Waters supercomputer takes approximately 18 ms. At a time step of 25 μs, one hour of simulation time requires 21 days of wall time. On Maxwell-class CPUs (GTX 980; CUDA

[‡] Benchmarking and performance tuning of LM was performed by Michael J. Hallock.

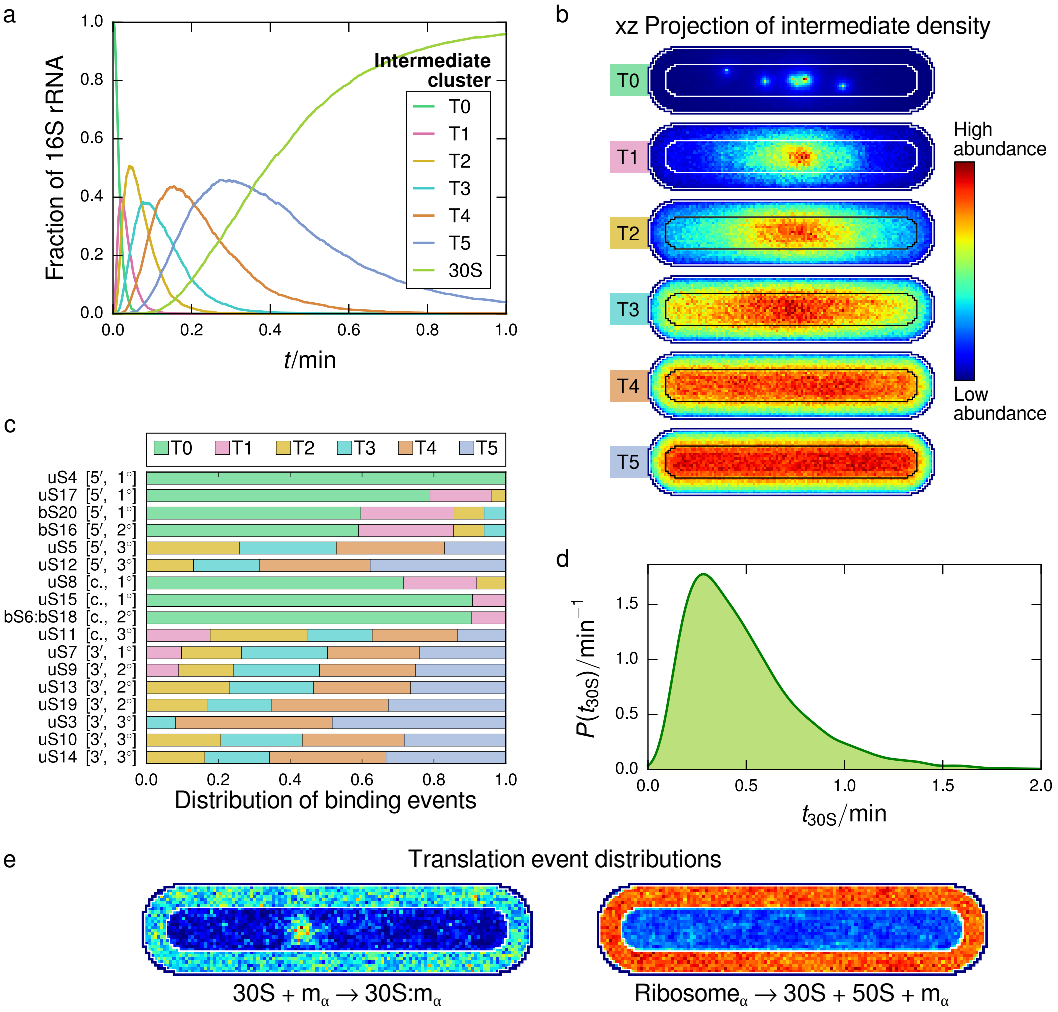


Figure 3.9 The assembly process of the 30S particle is spatially dependent. (a) Fraction of intermediate temporal clusters present as a function of time. Temporal clustering groups the 145 intermediate species into mutually exclusive groups based on their order of appearance in the assembly process. The precise assignment of intermediates to clusters is provided in Figure B.5. (b) Projections of the intermediate spatial probability distributions for the 6 temporal classes (T0–T5) onto the xz axis. The distribution of individual intermediates are reported in Figure B.4. (c) Distribution of protein binding events in each temporal class, providing a timeline of protein binding reactions. For example, all uS4 binding reactions occur in group T0 and all uS15 and bS6:bS18 binding reactions occur in T0 and T1. (d) Distribution of assembly times for the SSU. The birth time distribution, measured as the time from birth of 16S to birth of 30S, is approximately gamma distributed. (e) Translation is spatially dependent. Central y-slices of the 3D probability density of binding events: (left) 30S associating with mRNA from the alpha operon, and (right) dissociation events of ribosomes translating alpha mRNA. Binding of messenger to SSU appears to happen in two locations: outside the nucleoid region, and inside the nucleoid region localized near the originating operon. From the dissociation events, it is clear that the translating ribosomes are correctly excluded from the nucleoid region as intended.

6.5) in a desktop computer, the time step is approximately 6 ms and one hour of simulation time will finish within a week.

To further accelerate the reaction kernel runtime, we investigated specialization and employed code generation techniques to write a reaction kernel to solve the specific model being simulated. This has the benefit of requiring even fewer data structures to be accessed in constant memory as memory references are now replaced with immediate value loads, and loops that could not be unrolled at compile-time are flattened before compilation. Using this technique, runtimes on on GTX 980 GPU and the K20X accelerators was reduced to 1.9 ms and 4.0 ms per time step respectively, allowing one hour of simulation time to be completed in approximately 3 to 6 days. Simulations of the full 120-minute cell cycle would require 6 – 12 days depending on the GPU used. The enormous improvement in performance is achieved by applying algorithms that exploit the newest features in the rapidly developing field of GPU computing. These improvements will allow us to add more species and reactions to a simplified model describing regulation and coupling to the metabolic network.

3.4 Discussion and Outlook

Here we report on the progress to develop a simplified reaction-diffusion master equation description of the transcription, translation, and protein/rRNA association events comprising ribosome biogenesis in whole-cells. We have constructed an assembly model of the SSU, which is to our knowledge the most detailed description to date. Our whole-cell model accurately reproduces the assembly timescales of the SSU, and predicts both the identity of major assembly intermediates and their spatial distributions throughout the cell. By tuning the formation rate of the large subunit to match the formation rate of the SSU, we capture the increase of the ribosome count from 3000 to 6000 over the full 120-minute cell cycle. Nevertheless, there are several important features and reactions that a more complete model of ribosome biogenesis requires.

The low temperature assembly model predicts a heretofore unrecognized assembly pathway, through which the SSU is assembled in a $5' \rightarrow 3' \rightarrow$ central directionality. However, it is unlikely that this assembly pathway is biologically relevant due to the conditions from which it emerges. It appears to be an artifact of the low temperature (15 °C) *in vitro* conditions. This pathway is not seen in the reduced high temperature (40 °C) network, used as the basis of the whole-cell RDME simulations. Additionally, if *in vivo* assembly occurs cotranscriptionally, the proteins will bind in the order $5' \rightarrow$ central $\rightarrow 3'$ as the transcript leaves the polymerase. Though not directly relevant to ribosome biogenesis *in vivo*, this alternate pathway illustrates the sensitivity of coordinated assembly networks to varying conditions such as temperature.

The spatially resolved simulations exhibit strong localization of early SSU intermediates within the nucleoid region, even without explicitly treating cotranscriptional assembly. Our model predicts that 50% of the SSU will be assembled within 42 seconds which is faster than the accepted 30S maturation time of 30 – 90 seconds in rich media or 78 – 150 seconds in minimal media[142]. The two main contributions to the assembly time difference are the lack of uS2 and bS21 in our model and the omission of rRNA processing. These remaining tertiary proteins would be expected to have slow binding rates, on the order of uS3 and uS5, and could add 10 – 15 seconds to the assembly time.

An important additional feature to consider is rRNA processing and maturation reactions. We assume in the simplified model that the 16S is emitted from the ribosomal operons completely processed, however the transcript is actually polycistronic and includes the 16S, 5S, and 23S rRNA, and tRNA as well. Each gene in the transcript has to be processed individually. The processing of the rRNA involves a number of enzymes and is considered to take place primarily in the nucleoid region, although there are suggestions in the literature that some processing may occur at the inner membrane. The maturation processes are still being investigated, but as soon as a consistent understanding emerges these reactions can be included[143–145].

Another feature missing in our model is the action of assembly cofactors. Though the ribosome is capable of being reconstituted *in vitro* from only rRNA and r-protein, in living cells the process is aided by RNA chaperones, RNA helicases, ribosome-dependent GTPases, and other maturation factors[98]. These species act to improve the speed and efficiency of assembly by minimizing the misfolding of nascent subunits into kinetic dead ends. Pulse/chase quantitative mass spectrometry experiments have shown that the assembly cofactors RimM, RimP, and Era significantly increase the binding rates of particular r-protein during the *in vitro* assembly of the 30S[146]. However, kinetic data with varying cofactor concentrations is unavailable, limiting its applicability to our model. KsgA is an assembly cofactor which appears to have its greatest effect during *in vivo* assembly. Inclusion of this cofactor could significantly change the assembly landscape as well, since it functions as a checkpoint which blocks binding sites until the intermediate reaches the correct conformation to continue assembly[147]. However, the kinetics are likely difficult to measure since they must be measured *in vivo*.

The actual distribution of messengers in bacteria and their diffusive behavior is not well-understood and conflicting reports have been published stating that mRNA are freely diffusing throughout the cell[111], mRNA are addressed to certain subcellular areas in a sequence specific way[148], and that mRNA is localized near its originating operon[149]. Though we assume that the mRNA can diffuse freely, we see that the regions with the largest density of 30S/mRNA association reactions are found near the originating operon of the messenger and outside of the nucleoid region. This distribution arises due to two effects. First, the new messenger are created at the location of its operon and cannot diffuse far before association with a SSU. Second, translating ribosomes are excluded from the nucleoid region, which leads to an accumulation of mRNA outside the nucleoid region from the dissociation into 30S, 50S, and mRNA.

In our whole-cell simulations, the ribosomes are distributed such that only 7% are found in the nucleoid region. In fast-growing *E. coli*, 12% are found in the nucleoid region[111]. This is a reasonable result, since we are modeling slow-growing *E. coli* where the chromosome is assumed to be densely packed into a single copy of the genome. It has been proposed that the segregation arises from maximizing the conformational entropy of the chromosome and the translational entropy of the ribosomes[150], however this alone does not explain compaction of the chromosome seen in stationary phase and translationally arrested cells. Our method for imposing a difference in ribosome densities between the two compartments is rather simplistic, however since the exact reason ribosomes are excluded from the nucleoid region is not clear, implementing a more physically realistic segregation mechanism may be premature. In the future we will include the full DNA in our model in the form of a biased random walk as used in our previous work[33].

It is known that in living *E. coli* cells 15% of the ribosomes are not actively engaged in translation[151]. Only approximately 25% of the 30S subunits are found in translating ribosome complexes in our simulations. This seems problematic, however in this model only messengers which code for the SSU r-proteins uS3 – bS20 are transcribed. This leads to overexpression of the r-protein, as well as the underutilization of the available ribosomes. Transcription of mRNA that does not code for the r-proteins used in this model could restore the correct balance of free/transcribing ribosomes, as well as correct the steady-state levels of protein and free messenger.

The number of ribosomes in a bacterial cell is observed to be roughly linearly correlated with the cell's growth rate. Such relationship is captured by the empirical growth law[152,153] that parallels growth rates of bacterial cells with how they allocate resources to protein synthesis and metabolic functions. However, the cell's effort to enforce such balance between metabolism and macromolecular synthesis is yet to be understood. This SSU assembly model can be combined with genome scale models of metabolism and protein expression[154,155]. Through network reduction methods and parameter space searches, these models could be integrated into our RDME simulations to simulate living cells.

The integration of metabolism with the model of ribosomal biogenesis would require the explicit regulation of rRNA and r-protein expression. Currently, we prescribe a constant transcription rate for each operon such that all r-protein is produced at approximately the same rate. Introducing gene regulation would alleviate the necessity of fine-tuning these rates. The two most important modes of regulation to model are the autoregulation of translation of r-protein mRNA and the regulation of transcription by guanosine tetraphosphate (ppGpp)[98]. In the autoregulation mechanism, certain free r-proteins can bind to their own transcripts, though at an affinity lower than that they bind to rRNA, inactivating the mRNA by blocking its translation. Any excess of r-protein will downregulate its own expression, leading to a small free r-protein pool. Most of r-protein operons are regulated this way. The other mode of regulation is transcription deactivation via the global regulator, ppGpp, which is produced through the stringent response, i.e. during amino acid starvation conditions. The molecule binds to RNAP affecting its affinity to specific promoters. This effect depends on the sequence of the promoter: downregulating most of the genes necessary for growth including r-protein and rRNA, and upregulating various stress regulation genes and genes necessary for amino acid synthesis.

In summary we have presented the first steps toward a whole-cell level model of ribosome biogenesis in *E. coli*, starting with the assembly of the SSU. Our low temperature *in vitro* assembly model fits the experimental kinetic data extraordinary well, and predicts previously unobserved assembly pathways. The high temperature model reproduces the same binding timescales for all proteins measured in *in vitro* studies and predicts key assembly intermediates in agreement with the cryo-EM data. The high temperature model was used to construct a spatially resolved, whole-cell model of ribosome biogenesis taking transcription and translation into account. The cellular environment was constructed to approximate slow growing *E. coli* with a densely packed nucleoid region that excludes ribosomes. Although the assembly model was developed from experiments performed *in vitro*, with the increased cellular concentrations of r-protein it yielded 30S assembly times comparable to experiments performed *in vivo*. The RDME model predicted non-uniform spatial distributions of mRNA and early 30S intermediates. Though simplified, this model has real predictive power and will be used as the basis for more complete models of ribosome biogenesis and cellular metabolism. Systems Biology Markup Language versions of the well-stirred simulation and LM v2.2 input files of the whole-cell simulations will be made available on our website: http://www.scs.illinois.edu/schulten/research/ribosome_biogenesis_2015/. A tutorial describing the use of LM is available on our website as well.

Chapter 4

Ribosome biogenesis in replicating cells: integration of experiment and theory[*]

4.1 Introduction

In *Escherichia coli*, ribosomes account for approximately one fourth of the cellular dry mass and the majority of the total RNA[157]. It can be tempting, then, to think of the bacterial cell as a finely tuned machine for building ribosomes. Their ubiquity and high sequence conservation has made them an invaluable window into the process of evolution at the molecular level[96,158–160], and their role in protein synthesis involves them (either directly or indirectly) in essentially every process within the cell.

Ribosome production has evolved to be tightly regulated by the cell. This is no small feat, considering that each 70S ribosome involves the coordinated transcription, translation, folding, and hierarchical assembly of three strands of ribosomal RNA (rRNA) and over four dozen proteins, all within the heterogeneous, crowded intracellular space. Starting as early as 1966, pioneering *in vitro* studies began to unravel some of the mechanistic details of this process[161]. Work on the 30S small subunit (SSU) which is largely responsible for recognizing and decoding messenger

[*] Chapter 4 contains material reproduced with permission from Earnest TM, Cole JA, Peterson JR, Hallock MJ, Kuhlman TE, and Luthey-Schulten Z (2016). Ribosome biogenesis in replicating cells: integration of experiment and theory. *Biopolymers*, **105**(10), pp. 735–751. doi:10.1002/bip.22892.

RNA (mRNA), showed that assembly nucleates with the folding of the so called five-way junction in the 16S rRNA of the SSU (residues 27–45 and 394–554 in *E. coli*), and then proceeds through the hierarchical association of sets of ribosomal protein (r-protein), each progressively folding and stabilizing the rRNA growing tertiary structure[104,107–109,130,131]. Interestingly, a number of *in vitro* studies have observed this process proceeding over timescales on the order of the cell cycle or longer[107–109], while *in vivo* it can take just a few minutes[142]. Moreover, single cell-imaging studies on both slow- and fast-growing cells have also shown that complete ribosomes are not uniformly dispersed throughout the cytoplasm, but rather they tend to aggregate to the cell poles[33,110,111,113,162]. Understanding these phenomena requires a model with both a complete (or nearly complete) kinetic description of the assembly process and fine spatial resolution.

Recently, Earnest et al.[94] reported the first spatially resolved stochastic simulations of ribosome biogenesis for slow-growing *E. coli*. In that work, a model involving 251 different species (including the SSU, large subunit (LSU), rRNA, 18 proteins that bind to it, the genes and mRNA that code for them, and over 140 possible intermediates in the SSU assembly) and approximately 1300 reactions for transcription, translation, and ribosome assembly were developed and parameterized along with diffusion constants for all species. The use of a stochastic simulation methodology was important for a number of reasons. First and foremost, gene expression has been shown to be highly variable from cell-to-cell; this is especially pronounced when the molecules involved are in low copy numbers[65,66,163]. Ribosomal RNA is transcribed from seven operons interspersed throughout the *E. coli* genome, and many of the intermediate structures along the assembly pathways can exist in very few copies due to the rapid binding of additional proteins[94]. Accurately modeling the random diffusive motions and reactions of the individual substrates allowed Earnest et al.[94] not only to investigate the mean behavior of the assembly network, but also the inherent variability in it.

Although unprecedentedly complete, the model did not account for some of the most basic functions of the cell—namely, replication of the genome, cell division, and metabolism. Using mRNA distributions obtained from super-resolution imaging experiments, recent articles by Peterson et al.[1] and Jones et al.[164] showed that mRNA copy numbers exhibit a significant amount of variability simply by virtue of the fact that the genes that encode them are duplicated at some point during the cell cycle (which, in turn, depends on the genes' positions on the chromosome.) To quantitatively describe the replicative dynamics of the chromosome, we have generated a series of *E. coli* strains with gene loci labeled by a fluorescent repressor–operator system (FROS) distributed evenly around the chromosome. High-throughput imaging of these strains and identification and quantification of the gene copy number in each cell allows us to fit simple models of cell growth and genome replication to extract estimates for the timing of replication of each gene as a function of its position on the chromosome. We use these results to extend the ribosome biogenesis model to explicitly include cell growth, gene duplication, and division (henceforth referred to as the ribosome biogenesis model (RBM), for ribosome biogenesis model). Although single-cell rRNA and r-protein mRNA distributions are not available for direct comparison, a number of theoretical models of mRNA statistics—including some that account for gene duplication—have been proposed [1,164], although, importantly, they do not explicitly account for mRNA–ribosome interactions. The transcription and mRNA degradation rates in the RBM differ from those generated by the theoretical model in fitting the simulated mRNA distributions. We ultimately attribute this discrepancy to the fact that the RBM does not account for competition from non-ribosomal gene expression (e.g. genes involved in metabolism, regulation, etc.) We derive a simple statistical model that accounts for messenger production, degradation, and interactions with the ribosomes (henceforth referred to as the semi-analytic model (SAM), for semi-analytical model) which we use to investigate the dependence of mRNA statistics on chromosome duplication as well as the expression of non-ribosomal genes within the cell.

4.2 Results and Discussion

4.2.1 Determining replication initiation timing and progression[†]

To track the progress of replication in living cells, we constructed strains of *E. coli* where an array of 240 specific operators for *tet* repressor (TetR) was inserted chromosomally. The position of

[†] All experimental work was performed by Thomas E. Kuhlman. Image analysis was performed by TME.

the *tetO* array was varied to evenly sample loci over the full genome (Figure 4.1b) at 14 positions. Expression of TetR–EYFP *in trans* from the plasmid pBH74 allows for the direct visualization of genomic loci and observation of operon counts to be gathered from populations of cells. These statistics can be combined with a model of cell replication to determine initiation time, replication time and quiescent phase time.

The strains were grown to exponential steady-state, doubling every 120 min. Approximately 1000 epifluorescence and phase-contrast images were taken of each of the 14 strains. The data processing procedure was automated such that the detection of cells in a frame, the measurement of length and width of each cell, and the counting of fluorescent peaks were all handled without human intervention (Figure 4.1a). This yielded ~7600 total cells with an average length of 3.2 μm and width of 0.7 μm.

To extract the cell cycle parameters from these data, we have developed a probabilistic model linking cell growth with DNA replication. We assume the following about the nature of cell growth and DNA replication. Cell volume is proportional to length since the width of cells do not vary significantly over their cell cycle[165]. Individual cells show variability in widths, however not more than 10% (see Figure C.3 for the distribution of cell widths.) Cell lengths immediately prior to cell division, ℓ_0, are distributed log-normally

$$P_{\text{len0}}(\ell_0) = \frac{1}{\sqrt{2\pi}\sigma_{\text{len0}}\ell_0} \mathrm{e}^{-\frac{1}{2\sigma_{\text{len0}}{}^2}[\ln(\ell_0/\mu_{\text{len0}})]^2} \tag{4.1}$$

with location parameter μ_{len0} and shape parameter σ_{len0}. We base this assumption on experimental histograms of cell division lengths showing positive skewness[165] and recent theoretical analysis showing that under the influence of Gaussian random noise in the cell division time, the cell division length distribution is log-normal.[166]. Since we are modeling *E. coli* with a mass doubling time of 120 min, we assume that only one round of replication occurs per cell cycle. We assume the duration of DNA replication, T_{rep}, (i.e. the C period, Figure 4.2) is constant. Experimental measurements of the distribution of replication initiation times (i.e. duration of B period) over single cells is limited in the literature, however one study reports a broad distribution that could

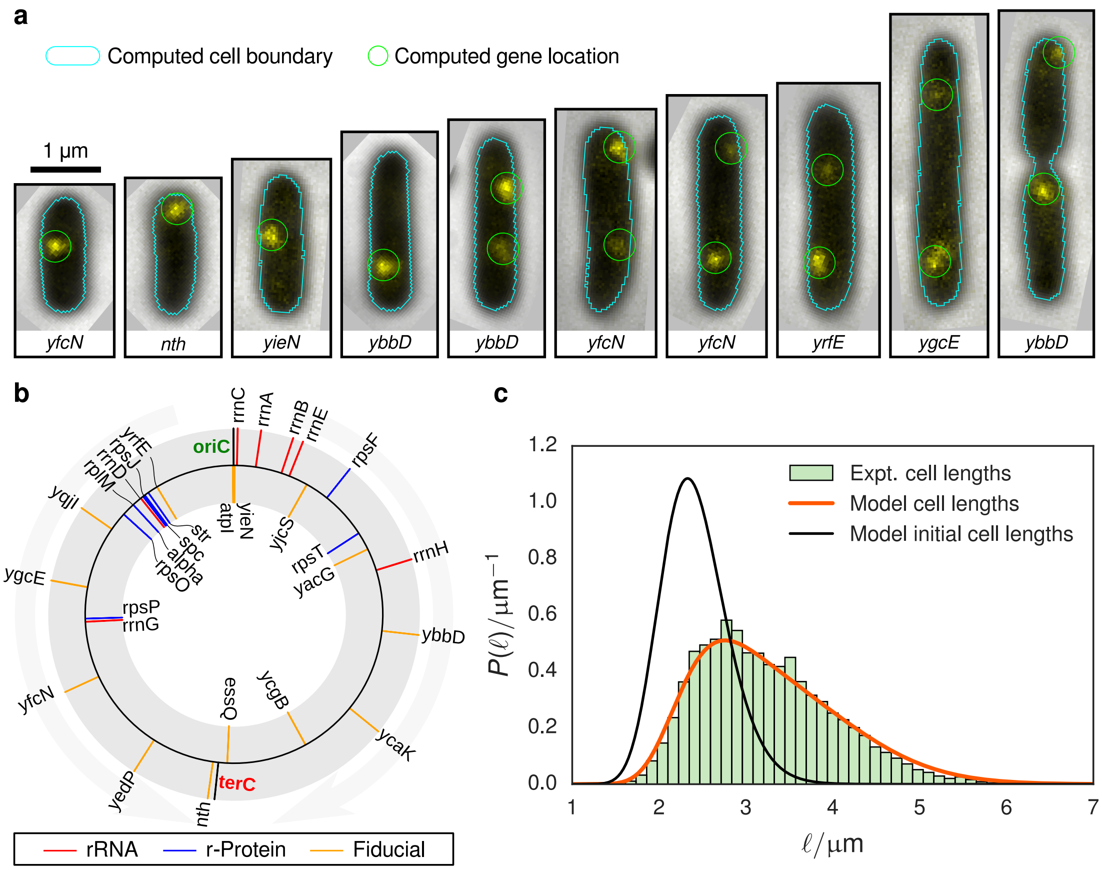


Figure 4.1 (a) Composite phase-contrast and epifluorescence images of representative imaging data used to determine cell length and operon positioning. The cell boundaries (cyan) and operon locations (green) are determined computationally. Examples of rejected cells are presented in Figure C.1. (b) Diagram showing the position of labeled genes used to track DNA replication (fiducial, orange) and those involved in ribosome biogenesis (rRNA, red; r-protein, blue). The black lines indicate the origin of replication (*oriC*) and the replication terminus (*terC*) (c) Abundance of cell lengths (green histogram) from imaging experiments are fitted to a simple exponential growth model (orange line, Eq. 4.6) to estimate the average and variance of cell lengths after division (black line). Approximately 7600 cells are included in this histogram.

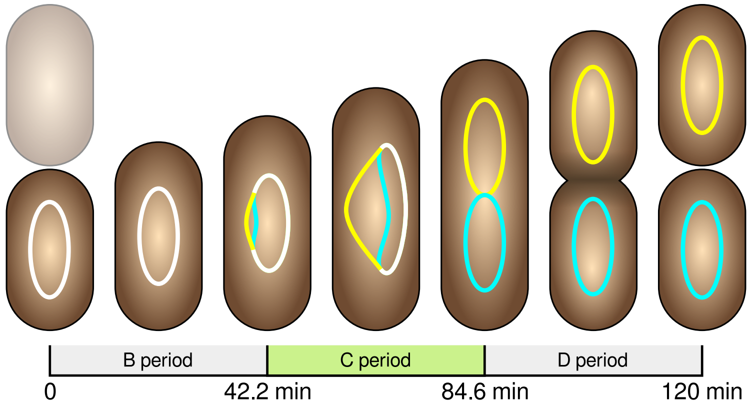


Figure 4.2 A schematic of the replication time parameters extracted from experimental images in the context of a 120 minute doubling cell.

be approximated by a truncated normal distribution[167]. For the sake of simplicity and to allow for some variability we have assumed the DNA replication initiation times, t_{rep}, are distributed via a normal distribution truncated at zero:

$$P_{\text{trep}}(t_{\text{rep}}) = \frac{\mathcal{N}_{\text{trep}}}{\sqrt{2\pi}\sigma_{\text{trep}}} \mathrm{e}^{-\frac{1}{2\sigma_{\text{trep}}{}^2}(t_{\text{rep}}-\mu_{\text{trep}})^2}, \tag{4.2}$$

where the normalization is

$$\mathcal{N}_{\text{trep}}{}^{-1} = \frac{1}{2} + \frac{1}{2}\,\mathrm{erf}\,\frac{\mu_{\text{trep}}}{\sqrt{2}\sigma_{\text{trep}}}. \tag{4.3}$$

We assume that the cells expand in length exponentially following the growth law

$$\ell(t) = \ell_0 2^{t/\mu_{\text{tdiv}}-1}, \tag{4.4}$$

where μ_{tdiv} is the mean division time. This assumption is supported by a great body of experimental evidence[165,168–172]. Finally, we assume that the cell length at birth and the replication initiation times are uncorrelated. There is evidence that the initiation time is correlated with the cell length at birth[167], however including this effect would make analysis of the model significantly more difficult. Using an analytical form for the distribution of cell mass, m, of exponentially growing bacteria[173,174],

$$P_{\mathrm{mass}}(m) = \frac{\mathcal{N}}{2m^2}\left(\int_0^{2m} \mathrm{d}m_0\, P_{\mathrm{mass0}}(m_0) - \int_0^{m} \mathrm{d}m_0\, P_{\mathrm{mass0}}(m_0)\right), \tag{4.5}$$

and assuming that $m \propto \ell$, we derived the distribution of cell lengths,

$$P_{\mathrm{len}}(\ell) = \frac{\mu_{\mathrm{len}}}{2\ell^2}\mathrm{e}^{-\sigma_{\mathrm{len0}}{}^2/2}\left(\mathrm{erf}\,\frac{\ln(2\ell/\mu_{\mathrm{len0}})}{\sqrt{2}\sigma_{\mathrm{len0}}} - \mathrm{erf}\,\frac{\ln(\ell/\mu_{\mathrm{len0}})}{\sqrt{2}\sigma_{\mathrm{len0}}}\right), \tag{4.6}$$

by substituting Eq. 4.1 into Eq. 4.5 and normalizing the distribution over positive lengths.

In order for our model to describe the relationship between the data we measure for each cell—its length, the identity of the labeled gene, and the number of copies of that gene—we must somehow theoretically connect the length of a cell with its gene copy number. To do this we use the cell age, t_{age}—a latent variable of our model. We must first compute the distribution of cell ages conditioned on cell length. By performing a change of variables on Eq. 4.1 using Eq. 4.4, we are left with a normal distribution of cell ages, where the mean age is a function of the cell length,

$$\mu_{\mathrm{tage}}(\ell) = \mu_{\mathrm{tdiv}} \log_2 \frac{2\ell}{\mu_{\mathrm{len0}}} \tag{4.7}$$

and the standard deviation of the age is

$$\sigma_{\mathrm{tage}} = \frac{\mu_{\mathrm{tdiv}}\sigma_{\mathrm{len0}}}{\ln 2}. \tag{4.8}$$

To prevent negative ages, we truncate the distribution and renormalize:

$$P_{\mathrm{tage}}(t_{\mathrm{age}}|\ell) = \frac{\mathcal{N}_{\mathrm{tage|len}}(\ell)}{\sqrt{2\pi}\sigma_{\mathrm{tage}}}\mathrm{e}^{-\frac{1}{2\sigma_{\mathrm{tage}}{}^2}(t_{\mathrm{age}}-\mu_{\mathrm{tage}}(\ell))^2}, \tag{4.9}$$

where the normalization is

$$[\mathcal{N}_{\mathrm{tage|len}}(\ell)]^{-1} = \frac{1}{2} + \frac{1}{2}\,\mathrm{erf}\,\frac{\mu_{\mathrm{tage}}(\ell)}{\sqrt{2}\sigma_{\mathrm{tage}}}. \tag{4.10}$$

The joint–conditional distribution function of cell ages and replication times given length is

$$P_{\text{tage,trep|len}}(t_{\text{age}}, t_{\text{rep}}|\ell) = P_{\text{trep}}(t_{\text{rep}})P_{\text{tage}}(t_{\text{age}}|\ell). \tag{4.11}$$

We consider a gene i to be copied if the age of the cell, t_{age}, is greater than the DNA replication initiation time, t_{rep}, plus the time required to copy up to and including gene i. Written in terms of the relative replication fork position $\hat{\chi}$, we have that

$$\hat{\chi} = \frac{t_{\text{rep}} - t_{\text{age}}}{T_{\text{rep}}} > \frac{\text{dist}(i, \textit{oriC})}{\text{dist}(\textit{terC}, \textit{oriC})} = \chi_i \tag{4.12}$$

when a cell has two copies of gene i. Here $\text{dist}(x, y)$ refers to the distance between two genes along its replichore. Using the growth law, the distribution of lengths at cell division, the distribution of all cell lengths, and the replication time distribution, we can derive the probability to find a cell with length, ℓ, whose replication progress is further than χ_i, $P(\hat{\chi} > \chi_i, \ell)$.

To compute the probability that a gene, i, has been replicated, we change variables to $\hat{\chi}$ in Eq. 4.11 and integrate over all $\hat{\chi}$ less than χ_i

$$\begin{aligned} C_{\text{rprg}}(\chi_i|\ell) &= \int_0^{\infty} \mathrm{d}t_{\text{rep}}\, P_{\text{trep}}(t_{\text{rep}}) \int_0^{t_{\text{rep}}+\chi_i T_{\text{rep}}} \mathrm{d}t_{\text{age}}\, P_{\text{tage}}(t_{\text{age}}|\ell) \\ &= 1 - \frac{1}{2}\mathcal{N}_{\text{tage|len}}[1 - \mathcal{N}_{\text{trep}} f(\mu_{\text{rprg}}, \sigma_{\text{rprg}})]. \end{aligned} \tag{4.13}$$

where

$$f(\mu_{\text{rprg}}, \sigma_{\text{rprg}}) = \frac{1}{\sqrt{2\pi}\sigma_{\chi}} \int_0^{\infty} \mathrm{d}x\, \mathrm{e}^{-\frac{1}{2\sigma_{\text{rprg}}{}^2}(x-\mu_{\text{rprg}})^2} \operatorname{erf} x, \tag{4.14}$$

$$\mu_{\text{rprg}} = \frac{\mu_{\text{rep}} + \chi_i T_{\text{rep}} - \mu_{\text{tage}}}{\sqrt{2}\sigma_{\text{tage}}}, \tag{4.15}$$

and

$$\sigma_{\text{rprg}} = \frac{\sigma_{\text{trep}}}{\sqrt{2}\sigma_{\text{tage}}}. \tag{4.16}$$

The probability to find a cell with length ℓ and n copies of gene i is then

$$P(\ell, n; i) = \begin{cases} C_{\mathrm{rprg}}(\chi_i|\ell) P_{\mathrm{len}}(\ell), & n = 1 \\ [1 - C_{\mathrm{rprg}}(\chi_i|\ell)] P_{\mathrm{len}}(\ell), & n = 2 \\ 0, & \text{otherwise} \end{cases} . \tag{4.17}$$

Thus the likelihood function is

$$\mathscr{L}(\theta|\{\mathrm{data}\}) = \prod_{(\ell,n,i)\in\{\mathrm{data}\}} P(\ell, n; i; \theta) \tag{4.18}$$

with

$$\theta = (\mu_{\mathrm{len0}}, \sigma_{\mathrm{len0}}, \mu_{\mathrm{trep}}, \sigma_{\mathrm{trep}}, T_{\mathrm{rep}}). \tag{4.19}$$

and the data for each cell is its length, ℓ, observed from the phase-contrast images, the copy number of the labeled gene, n, observed from the fluorescence data, and the identity of the labeled gene, i. Fitting Eq. 4.18 to the data simultaneously determines the mean cell division length and its variance, the mean DNA replication initiation time and its variance, and the time necessary to replicate the full genome.

The model parameters were determined by maximizing the logarithm of Eq. 4.18 over 7600 observed cells. To ensure that each operon contributed equally to the likelihood, we used the sum of the mean log-likelihood computed for each for each gene. We used a bounded, global optimization scheme, differential evolution[175], to maximize the objective function. The lower bounds were set to 10^{-6} to prevent numerical divergence and the upper bounds were set to 10 μm and 4 for the cell division length location and scale parameters, and 240 min for the replication time parameters. Uncertainties in the parameters were computed via bootstrap, using 15000 resamplings of the data. A summary of the fitting parameters and their uncertainties are provided in Table 4.1.

The cell length distribution is well-described by the model (Figure 4.1c). The location parameter of the log-normal distribution describing the cell lengths prior to cell division is

Table 4.1 Cell cycle parameters inferred from probabilistic model

Symbol	Description	Value from fit (mean ± std)
μ_{len0}	Location parameter of cell lengths immediately prior to division	4.772 ± 0.021 μm
σ_{len0}	Scale parameter of cell lengths immediately prior to division	0.1560 ± 0.0050
μ_{trep}	Mean replication initiation time	42.2 ± 3.0 min
σ_{trep}	Standard deviation of replication initiation time	22.1 ± 1.9 min
T_{rep}	Replication duration (C period)	42.4 ± 5.0 min
μ_{tdiv}	Mean time between divisions	120 min[a]
σ_{tdiv}	Standard deviation of time between divisions	12 min[b]

[a] From experiment
[b] Assumed

Parameters derived from model fitting along with uncertainties computed from bootstrapping.

4.772 ± 0.021 μm and the shape parameter of the distribution was 0.1560 ± 0.0050. These parameters converted to the arithmetic mean and standard deviation are 4.830 μm and 0.575 μm respectively, implying that new born cells are 2.415 μm long on average. These measurements can be compared to the division length reported for *E. coli* at a doubling time of 51 min reported by Taheri-Araghi et al.[165] of 4.40 ± 0.54 μm. The model predicts a mean replication initiation time of 42.2 ± 3.0 min (duration of B period) with a standard deviation of 22.1 ± 1.9 min, and a replication duration of 42.4 ± 5.0 min. These results are reasonable in light of the experiments of Skarstad et al.[176] who measured a B period of 34 min from *E. coli* B/r A doubling at 113 min and Adiciptaningrum et al.[167] measured the B period distribution for *E. coli* with at 130 min doubling time and reported a broad distribution with a mean of 30 min and a standard deviation of 21 minutes. Michelsen[177] reports a B period of 32 min and a C period of 52 min in *E. coli* K-12 MG1655 doubling at 137 min and shows that the C and D periods increases linearly with generation time when the doubling time is greater than 70 min, however these measurements tend to vary depending on the particular strain and the method of analysis.

Figure 4.3a shows the agreement of the experimental data to our model; fitting plots for all data are provided in Figure C.4 and Figure C.5. The model tends to underestimate the number of cells with two copies for genes near the origin and overestimates for genes near the terminus. However the peaks and dispersion in the model distributions reflects the experimental data well.

We also computed the probability of finding a cell with one copy of a gene. This was accom-

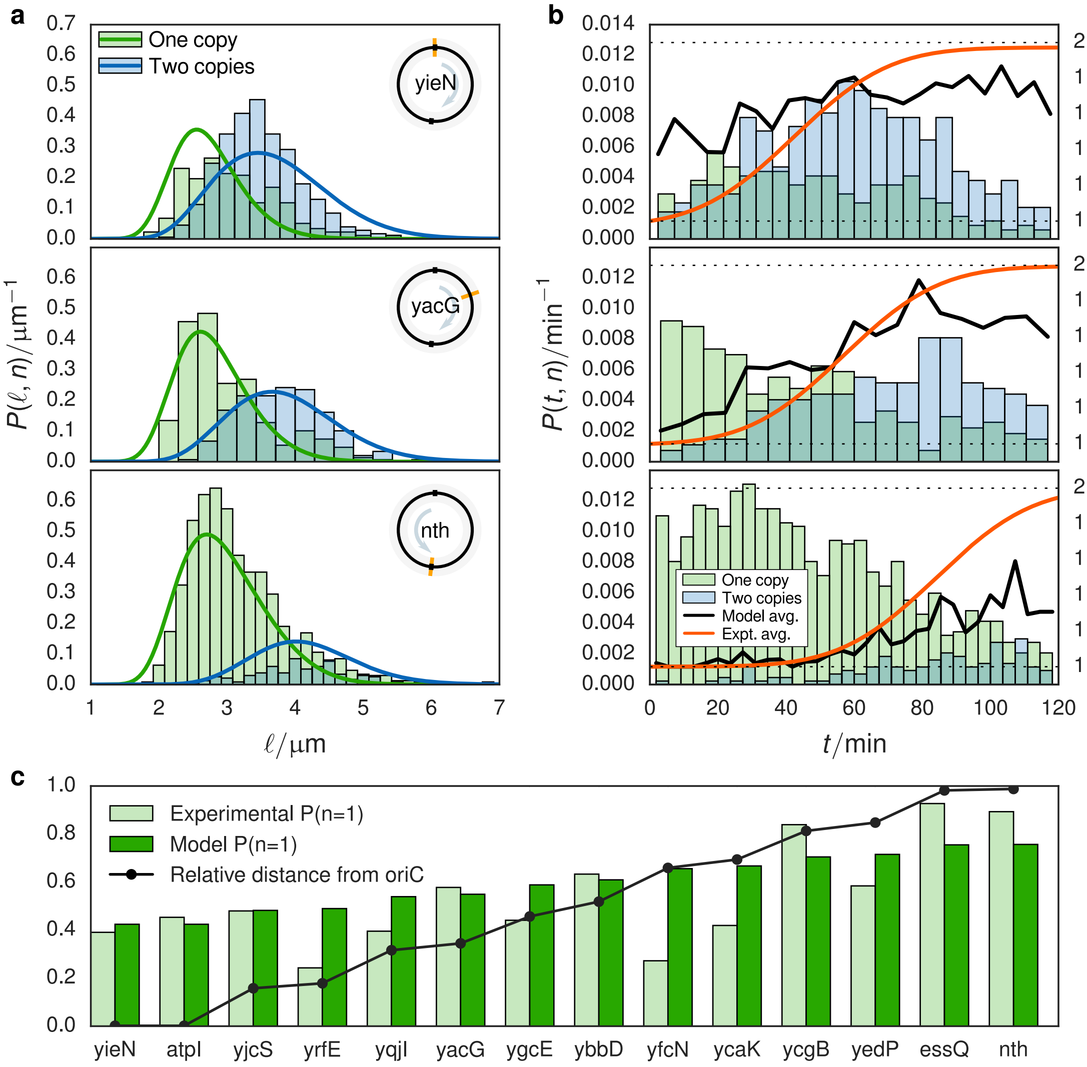


Figure 4.3 (a) Fraction of cells found with one gene copy (green) and fraction predicted by the model (blue). The distance from the origin of replication to the gene, relative to the distance between *oriC* and *terC* along its arm of the chromosome is plotted in black. (b) Abundance of cells with length ℓ and either one (green histogram) or two (blue histogram) gene copies. The probability densities associated with these histograms predicted from the model Eq. 4.17 are plotted as lines. (c) Abundance of cells with age $t \approx \mu_{\text{div}} \log_2 \frac{2\ell}{\mu_{\text{len0}}}$ and either one (green histogram) or two (blue histogram) gene copies for the same genes as shown in (a). The cumulative distribution function of gene replication times is plotted with a black line. Plots for all operons are available in Appendix C.

plished by considering an expression for the distribution of cell ages[178],

$$P_{\text{tage}}(t_{\text{age}}) = 2\mathrm{e}^{-\nu t_{\text{age}}} \int_{t_{\text{age}}}^{\infty} \mathrm{d}t\, P_{\text{tdiv}}(t), \tag{4.20}$$

and assuming that the individual cell division times are distributed normally with a standard deviation of 10% ($\sigma_{\text{tdiv}} = 12\,\text{min}$). Assuming that the bulk mass doubling rate, ν, is approximately equal to the mean cell division rate, μ_{tdiv}, and evaluating the integral we have

$$P_{\text{tage}}(t_{\text{age}}) = \mathcal{N}_{\text{tage}} 2^{-t_{\text{age}}/\mu_{\text{tdiv}}} \operatorname{erfc} \frac{t_{\text{age}} - \mu_{\text{tdiv}}}{\sqrt{2}\sigma_{\text{tdiv}}} \tag{4.21}$$

with

$$\mathcal{N}_{\text{tage}}{}^{-1} = \frac{\mu_{\text{tdiv}}}{\ln 2} \left(1 + \operatorname{erf} \frac{\mu_{\text{tdiv}}}{\sqrt{2}\sigma_{\text{tdiv}}} - \frac{1}{2} \exp \frac{(\ln 2)^2 \sigma_{\text{tdiv}}{}^2}{2\mu_{\text{tdiv}}^2} \operatorname{erfc} \frac{(\ln 2)^2 \sigma_{\text{tdiv}}{}^2 - \mu_{\text{tdiv}}^2}{\sqrt{2}\sigma_{\text{tdiv}}\mu_{\text{tdiv}}} \right). \tag{4.22}$$

The probability to find a cell that has not replicated its labeled gene is

$$P(n_i = 1) = \int_0^{\infty} \mathrm{d}t_{\text{age}}\, P_{\text{tage}}(t_{\text{age}}) P(n_i = 1 | t_{\text{age}}) \tag{4.23}$$

where

$$P(n_i = 1 | t_{\text{age}}) = \frac{1}{2} \operatorname{erfc} \frac{t_{\text{age}} - (\mu_{\text{trep}} + T_{\text{rep}} \chi_i)}{\sqrt{2}\sigma_{\text{trep}}} \tag{4.24}$$

follows directly from Eq. 4.2. Rewritten in a form amenable to numerical integration,

$$P(n_i = 1) = \frac{\mu_{\text{tdiv}} \mathcal{N}_{\text{tage}}}{\ln 2} \int_0^{\infty} \mathrm{d}u\, \mathrm{e}^{-u} \operatorname{erfc}(m_1 u + b_1) \operatorname{erfc}(m_2 u + b_2) \tag{4.25}$$

with the constants

$$m_1 = \frac{\mu_{\text{tdiv}}}{\sqrt{2}(\ln 2)\sigma_{\text{tdiv}}} \tag{4.26a}$$

$$b_1 = \frac{-\mu_{\text{tdiv}}}{\sqrt{2}\sigma_{\text{tdiv}}} \tag{4.26b}$$

$$m_2 = \frac{\mu_{\text{tdiv}}}{\sqrt{2}(\ln 2)\sigma_{\text{trep}}} \tag{4.26c}$$

$$b_2 = \frac{-\mu_{\text{trep}} - T_{\text{rep}}\chi_i}{\sqrt{2}\sigma_{\text{trep}}}. \tag{4.26d}$$

The probability to find a cell with one copy of each of the 14 genes is shown in Figure 4.3c, using the previously computed fitting parameters (Table 4.1). Genes *yrfE*, *yfcN*, and *ycaK* show the worst agreement, neither following the model predictions nor the trend of the other experimental data, however the single copy fraction data for the remaining 11 genes follow the expected trend and are well-described by the model parameters.

As a final test of the fitting of our model parameters, our expression for the probability of finding a cell with a single gene copy was used to independently estimate the replication initiation timing, and the replication duration (see Section C.2). This somewhat less-sophisticated treatment yielded values of $\mu_{\text{trep}} = 34.4$ minutes, and $T_{\text{rep}} = 45.9$ minutes. Importantly, although different in their approaches, both methods estimate similar C and D periods of around 40 minutes each.

It is remarkable that a reasonable measurement of the growth parameters can be made indirectly without monitoring individual cell lineages and labeling the replisome. Cell cycle control in bacteria is highly complex and not completely understood[165,166,172,174,179–182]. There have been at least three classes of cell growth models described in the literature: size-dependent division ("sizer")[174], time-dependent division ("timer")[181,182], and constant extension ("adder")[172,180] models, as well as more complicated mixed models[165,166] have all been proposed. A major result of many of these works is the fact that the size of a cell before and after division is correlated. We are unable to account for this in our model since our experiments do not track individual cell lineages. Thus we use a simple model which ignores the correlations between generations.

4.2.2 Modeling the effects of DNA replication on ribosome biogenesis

We built upon our previous kinetic model of ribosome biogenesis in *E. coli*[94] to construct the RBM, in order to investigate the effect of both gene duplication and changing volume due to cell growth. This model is simulated using Lattice Microbes (LM)[28,29], a software package designed to simulate stochastic reaction-diffusion systems through sampling of the underlying reaction–diffusion master equation (RDME). The spatial domain of the problem is discretized onto a lattice, with each lattice site containing discrete particles. Particles diffuse between lattice sites according to diffusion constants that are local to each cellular region and specific for each species. A Gillespie type kinetic Monte Carlo simulation determines which reaction occurs at each lattice site and which particles diffuse to neighboring sites. Since this technique is highly parallelizable; it is implemented in CUDA to take advantage of NVIDIA GPU, allowing for a complete cell cycle to be simulated in a single day.

The previous ribosome biogenesis model[94] has between modified such that new r-protein and rRNA operons (see Figure 4.1b for their loci) are added to the simulation at times reflecting their position in the genome using the parameters derived in Section 4.2.1, while dynamically growing the cell volume as the simulation progresses. Psuedocode describing the generation of the reduced assembly model (Algorithm C.1), the dynamic construction of cell geometry (Algorithm C.3), and the simulation procedure (Algorithm C.2) are provided in Appendix C. The kinetic model of ribosome biogenesis includes seven ribosomal RNA operons which code for the 16S rRNA and nine operons coding for the 18 r-protein, which along with the 16S rRNA, compose the 30S SSU of the ribosome. Transcription of these operons is explicit in this model—the particles representing the operons are placed in the cell nucleoid region based on their genomic position and emit messenger RNA species at a constant rate, i.e. unregulated, constitutive expression. Translation of r-protein is explicit as well—the mRNA engage in a diffusive search in order to bind to the SSU. The resulting complex associates with the LSU to form a translating ribosome. R-protein are emitted from the translating ribosome in the order in which the genes appear in the transcript. Upon completion, the complex dissociates into free mRNA, LSU, and SSU species, allowing the cycle to begin anew. Newly translated r-protein diffuse away and associate to SSU

Table 4.2 Summary of reactions in the whole-cell model

Type	Reaction	Parameter values	Units	Compartments
Assembly	$\mathrm{I}_i + P_j \longrightarrow \mathrm{I}_{i+1}$ (1° prot.)	$0.041 - 1.69$	$\mu\mathrm{M}^{-1}\,\mathrm{s}^{-1}$	`cytoplasm`, `nucleoid`
	$\mathrm{I}_i + P_j \longrightarrow \mathrm{I}_{i+1}$ (2° prot.)	$0.24 - 31.$	$\mu\mathrm{M}^{-1}\,\mathrm{s}^{-1}$	`cytoplasm`, `nucleoid`
	$\mathrm{I}_i + P_j \longrightarrow \mathrm{I}_{i+1}$ (3° prot.)	$0.025 - 1.75$	$\mu\mathrm{M}^{-1}\,\mathrm{s}^{-1}$	`cytoplasm`, `nucleoid`
Degradation	$\mathrm{mRNA}_i \longrightarrow \varnothing$	$1.0 \times 10^{-3} - 1.4 \times 10^{-3}$	s^{-1}	`cytoplasm`, `nucleoid`
Transcription	$\mathrm{DNA}_{\mathrm{rrn}X} \longrightarrow \mathrm{DNA}_{\mathrm{rrn}X} + 16\mathrm{S}$	0.037 (0.062)	s^{-1}	`nucleoid`
	$\mathrm{DNA}_x \longrightarrow \mathrm{DNA}_x + \mathrm{mRNA}_x$	$3.2 \times 10^{-3} - 7.8 \times 10^{-3}$	s^{-1}	`nucleoid`
		$(4.9 \times 10^{-3} - 0.012)$	s^{-1}	
Translation	$\mathrm{mRNA}_x + \mathrm{SSU} \longrightarrow \mathrm{Rib}^x_{\mathrm{init}}$	1.0×10^2	$\mu\mathrm{M}^{-1}\,\mathrm{s}^{-1}$	`cytoplasm`, `nucleoid`
	$\mathrm{Rib}^x_{\mathrm{init}} + \mathrm{LSU} \longrightarrow \mathrm{Rib}^x_0$	3.0	$\mu\mathrm{M}^{-1}\,\mathrm{s}^{-1}$	`cytoplasm`, `nucleoid`
	$\mathrm{Rib}^x_i \longrightarrow \mathrm{Rib}^x_{i+1} + \mathrm{P}_{x_i}$	$0.019 - 0.27$	s^{-1}	`cytoplasm`, `nucleoid`
	$\mathrm{Rib}^x_{\mathrm{term}} \longrightarrow \mathrm{SSU} + \mathrm{SSU} + \mathrm{mRNA}_x$	0.015	s^{-1}	`cytoplasm`, `nucleoid`
LSU birth	$\varnothing \longrightarrow \mathrm{LSU}$	6.5×10^{-4} (3.1×10^{-4})	3.1×10^{-4}	`cytoplasm`, `nucleoid`
Dimerization	$bS6 + bS18 \longrightarrow \mathrm{bS6{:}bS18}$	1.0	$\mu\mathrm{M}^{-1}\,\mathrm{s}^{-1}$	`cytoplasm`, `nucleoid`
	$\mathrm{bS6{:}bS18} \longrightarrow bS6 + bS18$	8.7×10^{-3}	s^{-1}	`cytoplasm`, `nucleoid`

Parameters which differ between the RBM and RBMfv are provided for each with the RBMfv parameter in parenthesis. The complete assembly network is provided in Figure C.8 of Appendix C, the complete list of all 1300 reactions is available Appendix B.

assembly intermediates following the assembly network described in Earnest et al.[94]. A diagram of the assembly network is shown in Figure C.8 of Appendix C. DNA replication is implemented by choosing a replication initiation time t_{rep}, from a normal distribution with mean μ_{rep} and variance $\sigma_{\mathrm{rep}}{}^2$. New operon copies are added to the simulation at times $t_i = t_{\mathrm{rep}} + \chi_i T_{\mathrm{rep}}$ which are taken directly from the experimental analysis in Section 4.2.1. The operon species are not subject to diffusion in our model, rather they are moved along the long axis of the cell such that they will be found in the same position in the daughter cells as in the mother cell at the start of the cell cycle (Figure 4.4b). This is a vast simplification of the dynamics of the chromosome, however it is the simplest approach given the lack of detailed time-dependent gene localization information available in the literature.

We use $\frac{1}{2}\mu_{\mathrm{len0}}$=2.4 µm from the modeling of the experimental data (Section 4.2.1) as the initial length of the cell and the mean cell width, 0.7 µm computed from the raw cell data, as the simulated cell's width. The cell grows to μ_{len0}=4.7 µm over the course of its 120 minute cell cycle following the growth law, Eq. 4.4. The new cell geometry, which includes the membrane, cytoplasm, and nucleoid cellular compartments, is computed using constructive solid geometry

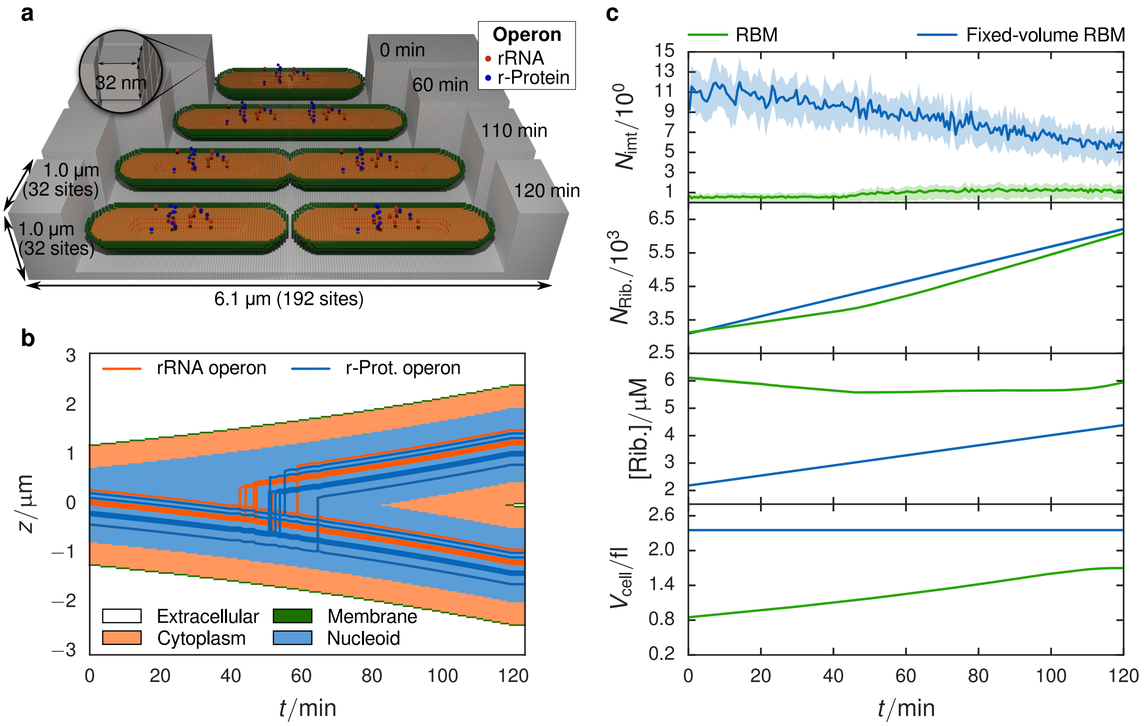


Figure 4.4 (a) Schematic of geometry used in RBM simulations. The lattice is 32 × 32 × 192 sites in the x, y, and z directions respectively, with a lattice spacing of 32 nm. The simulation volume consists of 4 regions: (1) extracellular space (gray), (2) membrane (green), (3) cytoplasm (orange), and (4) nucleoid (not colored, found in center of cytoplasm). The initial length, 2.4 µm, and the width of the cell, 0.7 µm, were chosen from the previous experimental analysis (Section 4.2.1). The proportion of the nucleoid region to the cytoplasm is based on measurements of cryo-electron tomograms of slow-growing *E. coli*[33]. Operon species are placed within the nucleoid region based on their genomic loci and replicated at times computed from their genomic distance to the origin of replication. The position of the operon species is evolved in time such that the operons in the daughter cell are found in the same position as the operons in the mother cell. The cell volume grows constantly throughout the cell cycle at an exponential rate, where upon it divides into two daughter cells of length 2.4 µm. (b) Kymograph showing the evolution of spatial compartments and operon locations over one cell cycle. The jagged steps arise from the discreteness imposed by the 32 nm lattice. (c) Comparison between RBM (green) and RBMfv (blue) models using 16 replicates. Means are represented by solid lines and the interquartile range is given by the shaded area. There is an significantly lower average SSU intermediate count seen in the RBM compared to the RBMfv (top panel), which is a result of the changing cell volume. In the last three panels are plotted the absolute count of ribosomes (translating as well as dissociated), the absolute ribosome concentration, and the cell volume. The RBM produces ribosomes at approximately the same pace as volume expansion, leading to a constant ribosome concentration over the cell cycle.

directly into the lattice data structure. The nucleoid compartment dimensions are chosen to match the proportions of nucleoid to cytoplasm observed in cryo-electron tomograms of slow-growing *E. coli*[33] (available in Figure C.7). When the lattice changes, sites where particles were once forbidden are now allowed and the chemical species rapidly undergo diffusive relaxation to fill the empty space. During the constriction of the cell during division, particles in sites which were once cytoplasm can end up outside of the cell. This problem is mitigated by using the membrane compartment to direct outlying particles back into the cytoplasm. For all particles in the simulation, their transition rate from the membrane to the cytoplasm site type is set to the maximum diffusion rate, $a^2/4\Delta t$, where a is the lattice constant, and Δt is the time step. Transitions from the cytoplasm into membrane sites are all set to zero. By changing the lattice slowly as well as using the membrane sites to redirect straying species, no particles are lost into the extracellular compartment.

Since the volume of the cell and number of gene copies change throughout this simulation, the original parameters used in Earnest et al.[94] were slightly modified. The mRNA and rRNA transcription rates were scaled by a factor of 0.65 and 0.60, respectively, and the zeroth-order LSU birth rate was scaled by 2.1. The change in the transcription rates reflect the changing copy numbers, where as the change in the LSU birth rate is a consequence of the changing volume. These changes were executed in order to ensure the same particle copy numbers at the end of the cell cycle were reached as in the original simulations[94] (RBMfv) to allow for a direct comparison which investigates the effect of cell growth and gene duplication. In order to compare the RBM on even footing with the RBMfv, the RBMfv was simulated again using the current development version of LM (version 2.3a) over 16 replicates.

Comparing the two models, the initial and final species counts are practically identical for all classes of particles with the exception of the SSU intermediates (see Figure 4.4c and Table 4.3.) Here we see that the final intermediate count in the RBMfv is approximately a factor of five larger than the count seen in the RBM. The origin of this effect is due in part to the increased protein concentration at the start of the cell cycle in the RBM. Though the absolute protein numbers are approximately equal, the RBM volume is smaller than the constant cell volume over the full cell

Table 4.3 Particle counts in RDME simulations

	RBM counts		RBMfv counts	
Particle Class	Initial	Final	Initial	Final
All ribosomes	3125 ± 54	6191 ± 58	3094 ± 49	6208 ± 73
Translating ribosomes	545 ± 15	1088 ± 23	528 ± 18	1045 ± 26
Dissociated ribosomes	2580 ± 57	5103 ± 60	2566 ± 53	5163 ± 71
SSU intermediates	1.1 ± 1.4	1.2 ± 1.2	11.4 ± 3.5	5.8 ± 2.7
Ribosomal protein	34 000 ± 3500	69 800 ± 4800	33 200 ± 2700	66 400 ± 4200

Initial and final particle counts from the RBM and RBMfv trajectories (mean±std).

Table 4.4 Particle concentrations in RDME simulations

	RBM concentrations [μM]		RBMfv concentrations [μM]	
Particle Class	Initial	Final	Initial	Final
All ribosomes	6.11 ± 0.10	6.055 ± 0.056	2.184 ± 0.034	4.382 ± 0.052
Translating ribosomes	1.065 ± 0.030	1.064 ± 0.023	0.373 ± 0.013	0.738 ± 0.018
Dissociated ribosomes	5.04 ± 0.110	4.991 ± 0.059	1.811 ± 0.038	3.644 ± 0.050
SSU intermediates	0.0022 ± 0.0028	0.0012 ± 0.0011	0.0081 ± 0.0024	0.0041 ± 0.0019
Ribosomal protein	66.5 ± 6.9	68.2 ± 4.6	23.5 ± 1.9	46.9 ± 2.9

Initial and final concentrations from the RBM and RBMfv trajectories (mean±std).

cycle. The RBMfv cell geometry is significantly greater than the RBM geometry since we had used dimensions of 4 μm × 0.9 μm in the original study[94]. However there appear to be other effects at play since the volume difference of 1.4× is not enough to account for the total difference.

The changing volume due to cell growth causes particle concentrations to remain relatively constant throughout the cell cycle (Table 4.4). For example the ribosome concentration in the RBM spans 5.5–5.9 μM over the cell cycle, where as in the RBMfv the concentration spans 2.4–4.16 μM. However in the RBM the concentration tends to peak before and after cell division (Figure 4.4c). In the bottom panel of Figure 4.4c, the increase in volume slows down near the end of the cell cycle when the cell begins dividing. The ribosome number increases linearly over the whole cell cycle, however the growth of the cell volume can no longer keep the pace with ribosome production during this slowing, leading to an increase in ribosome concentration at the end of the cell cycle. When the cell finally divides, the protein concentration can now relax to the steady state concentration.

Though the majority of the chemical species in the RDME simulations show no spatial hetero-

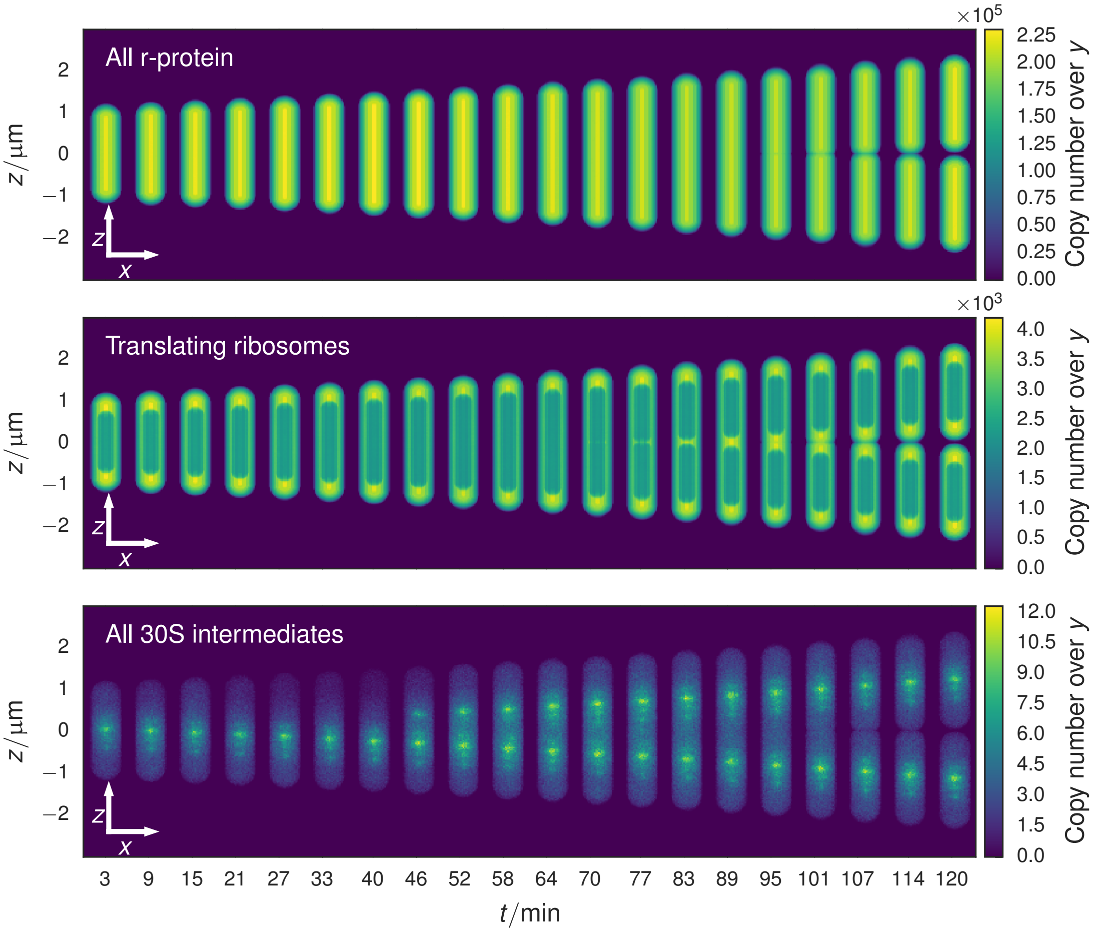


Figure 4.5 xz copy number projections of cells at evenly spaced times throughout the cell cycle. The time evolution of the cell geometry is evident in this series of projections. Constriction begins approximately 100 minutes into the cell cycle through the constriction of the cell membrane. Ribosomal protein (top) diffuses rapidly through all compartments, leading to a distribution which mirrors the thickness of the cell at each (x, z) coordinate. The transition rates of translating ribosomes (middle) between the nucleoid and cytoplasm regions is biased to limit the number of ribosomes in the nucleoid, leading to localization of ribosomes to the cell poles and membrane. The most pronounced spacial heterogeneity is due to the SSU intermediates (bottom), where the earliest intermediates which result from the binding of primary proteins are found near the rRNA operon from which the 16S rRNA was transcribed.

geneity, e.g. r-protein (Figure 4.5, top), two classes of particles exhibit nonuniform distributions throughout the cell. Translating ribosomes, composed of an SSU particle, an LSU particle, and an mRNA, are partially excluded from the nucleoid region by imposing a bias in the transition rates between the nucleoid and cytoplasm compartments. The transition rate from the nucleoid region to the cytoplasm is four times greater than the reverse transition. These biased transition rates model the excluded volume effects arising from the folded chromosome which is not included in the simulation due to a restriction of the number of species allowed in the present version of LM. Heterogeneous distributions of ribosomes have been observed in single particle tracking experiments, which showed that the fully associated ribosome is partially excluded from the nucleoid[111] region while the individual subunits are not[113], as well as in cryo-electron tomography of slow-growing *E. coli*[33] (see Figure C.7.) Since the fate of particles in these RDME simulations are determined by reaction and diffusion processes alone, biased transition rates are necessary to implement excluded-volume effects which arise due to intermolecular forces between particles. Though this is a simplistic approach, it is sufficient for the needs of this study.

The other particle class exhibiting a nonuniform spatial distribution are the SSU assembly intermediates (Figure 4.5 bottom). Ribosomes assemble in a well-defined binding order, where some proteins can only bind once other proteins are associated with the nascent subunit (see Figure C.8.) The earliest SSU intermediates, consisting of the primary and secondary binding proteins associated with the $5'$ and central domains of the 16S rRNA[94], are short lived and are found only within a few hundred nanometers of the site from which the rRNA was transcribed. Due to their short lifetime, their density tracks the position of the rRNA operon tightly. Later intermediates which are beginning to include tertiary binding proteins diffuse farther away from the originating rRNA operon until all memory of their birthplace is washed out.

4.2.3 Effects of DNA Replication and Translation on mRNA Statistics[‡]

As there are no experimental distributions available, computed distributions of the rRNA and r-protein operon mRNAs obtained from our simulations were compared to theoretical results from

[‡] Semi-analytic model developed and tested by John A. Cole and Joseph R. Peterson.

Peterson et al.[1]. The theory derived in Peterson et al.[1] considers a constitutively expressed gene that is replicated during the cell cycle and includes the time-dependent messenger degradation. It was found that modeling the time-dependence was critical to capturing the correct shape and statistical features of the messenger distribution for highly expressed genes or genes with long half-life, both criteria which are met by the r-protein operon genes. We found that the simulated RNA exhibited significantly higher expression and greater variability than the theory of Peterson et al.[1] predicted. Attempts to fit the messenger distributions to theoretical distributions (see Figure 4.6 green lines; described in Section C.3) yielded estimates of $k_{t,\mathrm{eff}}$ and $k_{d,\mathrm{eff}}$ (the messenger transcription and degradation rates, respectively) that differed systematically from the rates used in the RBM simulations—fit $k_{t,\mathrm{eff}}$ values were approximately four times larger than those used in the RBM simulations while fit $k_{d,\mathrm{eff}}$ values are about four times smaller (see Table 4.5 and Figure C.9). We note, however, that the distributions based on the results of the theory[1] do show better agreement than those of an earlier model of mRNA production that accounted for gene duplication, but neglected mRNA decay[164]. This is due to the high expression value of the mRNA and the long half-life of the messengers (8–12 min) both of which require the mRNA relaxation to be explicitly accounted for to capture the correct statistics[1].

An important omission in the RBM, and the underlying reason for the disagreement we see with the results of Peterson et al.[1], is that the simulated cells express only the genes involved in ribosome biogenesis. In reality, cells express a multitude of other mRNA and proteins in order to perform other cellular functions (e.g. metabolism and gene regulation, etc.) In order to investigate how these “missing” mRNA may affect our r-protein mRNA statistics, we constructed a simple model of messenger production that accounts for both gene duplication and interactions with the cell’s ribosomes (denoted SAM) consisting of the reactions

$$D(t) \xrightarrow{k_t} D(t) + m \tag{4.27a}$$

$$m \underset{k_u}{\overset{k_b}{\rightleftharpoons}} n \tag{4.27b}$$

$$m \xrightarrow{k_d} \varnothing \tag{4.27c}$$

Here $D(t)$ represents the gene copy number on the DNA; its time-dependence signifies that at some time t_r (the replication time) it will double from one copy to two copies, and k_t and k_d are the transcription and degradation rate of the mRNA, respectively. Importantly, this model includes transitions of the messenger, m, into and out of a second state, n, which represents the ribosome-bound mRNA. The ribosome binding and unbinding rates are denoted k_b and k_u, respectively, and the binding rate is understood to be a function of the free ribosome concentration. The binding rate is $k_b = 4.2 \times 10^8 M^{-1} s^{-1}$ while the unbinding constants can be estimated as described in Section C.1.2 (also found in Table 4.5). This model assumes (as do our simulations) that ribosome-bound messengers are protected from degradation.

A chemical master equation (CME) corresponding to Eq. 4.27a (see Eq. C.1) was used to derive a set of ODEs and boundary conditions that describe the mean and variance of m and n (see Eq. C.2 and Eq. C.3). By assuming some number, c, of other genes whose mRNA compete for the available ribosomes, we estimated the equilibrium concentration of free ribosomes by solving the system numerically. Subsequent time-averaging over the cell cycle[1] yielded values for the mean and variance of the modeled mRNA. We computed the mean and Fano factor for each of the r-protein operons based on their respective rate parameters and gene doubling times (see Table 4.5). When $c = 8$, which approximates the case of the RBM simulations (there are a total of 9 r-protein operons in *E. coli*; messengers from 8 operons actively compete with the messengers from the operon of interest), we found that the resulting means from the SAM showed very good agreement with simulated RBM values, although the resulting Fano factors tended to be slightly overestimated (see Figure C.10 a, red and blue dots). However, when the value of c in the SAM was set closer to a biologically realistic value (on the order of 1000, assuming roughly 25% of the *E. coli* genome is actively expressed[183]), the resulting means and Fano factors essentially matched those predicted by Peterson et al.[1] (see Section C.4 and Figure C.10 a, green triangles and black "+" signs). These results underscore the need for including the expression of other non-r-protein messengers in future RBM simulations.

Our analysis of the SAM indicates that when in the biologically realistic regime ($c \approx 1000$), messengers are generally not bound by ribosomes and their statistics can be described by the theory of Peterson et al.[1]. The question then arises: What is the expected mean count of messengers for each of the r-protein operons and how should the RBM be modified when competing mRNA are modeled? Using values from the CyberCell Database, which tabulates statistics describing an average *E. coli* cell[184], we estimate that the total count of mRNA to be between 3800 and 10 000 in cells with our measured average length and width (3.2 μm and 0.715 μm, respectively; see Figure 4.1c and Figure C.3). Using relative gene expression values from high-throughput sequencing data for *E. coli*[185] we then estimate the mean mRNA counts for the ribosomal operons are between 20 and 120, which are in good agreement with the RBM values (55 and 145). In the biologically realistic regime for c the transcription and degradation kinetics used in the RBM give mean and noise values that are much lower than these estimates (Figure C.10 a green dots). This indicates that future applications of the RBM which include competing mRNA will require transcription rate parameters that are about four times higher and degradation rate parameters that are about four times lower than in the current RBM to achieve mean mRNA counts that match experiments (as indicated by linear regression between fit and RBM rate parameters; see Figure C.9). Using the theory of Peterson et al.[1] we have estimated that the k_t and k_d values required for the r-protein operons necessary to capture the correct mean messenger counts when modeling all competing mRNA (see Table 4.5).

4.3 Conclusions

In this article we performed fluorescence imaging studies at the single-cell level in order to estimate the timing and duration of DNA replication in slow-growing *E. coli* (doubling time of approximately 120 minutes). We described a simple analytical model describing growth and DNA replication in slow-growing *E. coli* (only one replication process per cell cycle) which does not require the explicit tracking of cell lineages and applied it to our single-cell studies. The B and C parameters determined by the model, 42.2 min and 42.4 min respectively, are reasonable when compared more direct measurements in bulk[176,177] or in single cells[165,167]. These parameters were used to improve a recent spatially resolved, whole-cell model of ribosome biogenesis[94]

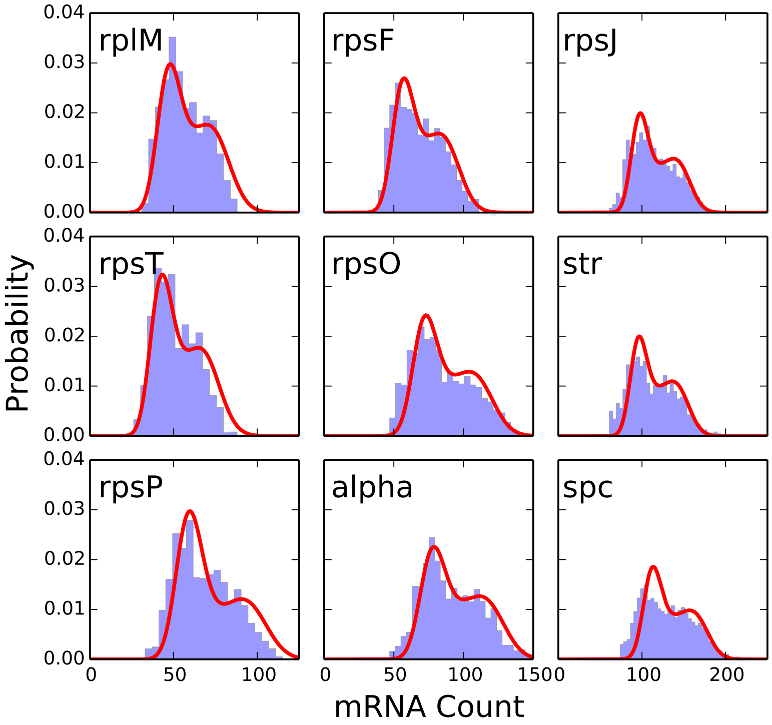


Figure 4.6 Messenger distributions simulated in the ribosome biogenesis model (RBM; blue histogram) with fits from the theory of Peterson et al. [1] (red curve.) Fit parameters for the theory with mRNA relaxation can be found in Table 4.5.

Table 4.5 Rate parameters for the r-protein operon mRNA in the RBM

Operon	k_t (s^{-1})	k_d (s^{-1})	k_u (s^{-1})	$k_{t,\mathrm{eff}}$ (s^{-1})	$k_{d,\mathrm{eff}}$ (s^{-1})
alpha	0.0047	8.363×10^{-4}	0.0079	0.01813	0.00023
rplM	0.0030	1.197×10^{-3}	0.0119	0.01431	0.00030
rpsF	0.0036	8.955×10^{-4}	0.0103	0.01513	0.00027
rpsJ	0.0060	1.029×10^{-3}	0.0059	0.02091	0.00022
rpsO	0.0045	1.238×10^{-3}	0.0082	0.01810	0.00025
rpsP	0.0038	9.785×10^{-4}	0.0092	0.02220	0.00037
rpsT	0.0027	1.144×10^{-3}	0.0139	0.01519	0.00036
spc	0.0069	9.206×10^{-4}	0.0055	0.02225	0.00020
str	0.0058	8.062×10^{-4}	0.0063	0.02065	0.00022
Mean			0.0042	9.8359×10^{-4}	0.0080

Transcription (k_t), degradation (k_d), and messenger unbinding (k_u) rate parameters for the r-protein operon mRNA in the RBM (scaled from those in Earnest et al.[94] as discussed in Section 4.2.2). Each value for the unbinding rates k_u was estimated according to Eq. C.9. The last line gives the harmonic mean over all individual operon rate parameters. These mean values were used to make Figure C.10 b. Effective rate parameters ($k_{t,\mathrm{eff}}$, $k_{d,\mathrm{eff}}$) are from fitting the simulated messenger distributions with the theory of Peterson et al.[1].

that involved the transcription and translation of the rRNA and r-protein operons involved in production of the ribosomal 30S small subunit, as well as its assembly. This model was augmented through the use of the experimentally measured parameters to include the effects of cell growth and gene replication, the latter of which has been shown to significantly impact the copy number statistics of mRNA in models of gene regulation[1,164]. We found that the r-protein operon messenger counts that emerged from our ribosome biogenesis model without regulation did not appear to be well-described by published theoretical models[1,164]. Specifically, the simulated messengers were expressed in greater numbers and with greater variability than the theory of Peterson et al.[1] predicted. We found that this was associated with the low number of non-ribosomal genes in the RBM. By constructing a simple semi-analytical model SAM that accounts for varying numbers of non-ribosomal genes to be expressed, we showed that the mRNA statistics of a cell expressing realistic numbers of non-ribosomal genes should be close to those predicted by Peterson et al.[1]. This means that in order to recover the proper ribosomal messenger counts, future versions of the RBM that include other cellular networks like metabolism and regulation will also require adjustments to the r-protein operon transcription and mRNA degradation parameters.

4.4 Materials and Methods

4.4.1 *E. coli* operon quantification

Strains and plasmids All strains used in this study are derivatives of *E. coli* K-12 MG1655 Δ*lac*[186–188], in which the entire *lac* operon has been deleted from the N-terminus of *lacI* to the C-terminus of *lacA* using the method of Datsenko and Wanner[189]. Gene locations and numbers were determined using the FROS, where of the integration of an array of 240 operators for TetR, was performed at each of 14 loci at evenly spaced intervals around the chromosome using Landing Pad technology[186–188,190,191].

Fluorescent repressor operator system Gene locations were determined using the FROS performed as described[192]. Integrations were made at each site consisting of an array of 240 operators for TetR using Landing Pad technology. After growth to steady state as described below, 0.01% L-arabinose was added to each culture 1 hour before fixation to induce expression of TetR tagged with fluorescent EYFP *in trans* from the plasmid pBH74[192]. Cells were then fixed and processed as above.

Media and growth conditions At the start of an experiment, a seed culture of each strain was inoculated from a glycerol stock into 2 ml lysogeny broth with appropriate antibiotics in 14 ml polypropylene round bottom tubes (Falcon) and allowed to grow to saturation in a 37 °C shaking water bath. This seed culture was then diluted 1000× into 3 ml of M63 minimal medium (100 mM KH_2PO_4, 15 mM $(NH_4)_2SO_4$, 1.7 µM $FeSO_4$, 1 mM $MgSO_4$) + 0.5% glycerol in 20 mm diameter glass test tubes and allowed to grow with extremely vigorous shaking in a 37 °C water bath (New

Brunswick Scientific model G76) until OD_{600} of the culture reached 0.5–0.6 as measured with a spectrophotometer (Bio-Rad SmartSpec 3000). These cultures were then used to inoculate another 25 ml baffled Erlenmeyer flasks of identical fresh medium pre-warmed to 37 °C at an initial density of OD_{600} = 0.005 and again grown with vigorous shaking in a 37 °C water bath. Samples were taken and the OD_{600} of the culture was measured at regular intervals to determine the doubling time of the culture. When the density of the culture reached OD_{600} = 0.2–0.4, the culture was harvested and fixed by the direct addition of an equal volume of freshly prepared and filtered 5% paraformaldehyde in phosphate buffered saline (PBS). The resulting solution was allowed to continue shaking at 37 °C for 10 minutes and was then placed on ice for 30 min. Cells were washed three times via centrifugation and resuspension in 1 ml filtered, ice-cold PBS. At the time of harvest, we estimate that the cultures had been growing in exponential steady-state for ~ 10 generations.

Microscopy After preparation, samples were mounted on glass slides using 40% glycerol. Imaging was performed using a Nikon Eclipse TE2000U microscope with an Applied Scientific Instruments PZM-2000 automated stage utilizing Metamorph automation software. 1000 images per strain were collected using epifluorescent illumination with a 100× phase-contrast objective combined with a 4× telescope attachment using a Roper Scientific Cascade:512 camera.

4.4.2 Data analysis

Image analysis All image analysis was performed in performed in the Jupyter environment[193] using the SciPy Stack[194] and scikit-image[195]. Following background subtraction of all phase-contrast images, a binary mask was computed from each frame using adaptive thresholding to identify potential cells. The potential cell regions from the phase-contrast images were then normalized to [0,1], where by a second binary mask of the cell was constructed from pixels with a normalized intensity less than 0.37. Cell lengths were measured from the arc length of a 5th degree polynomial fit to the cell mask in order to prevent measurement error due to cell curvature.

Regions of EYFP fluorescence were evaluated for suitability by computing the intensity histogram and only accepting regions with a skewness greater than 1. Locations of labeled operons were determined by finding the local maxima of the Gaussian filtered fluorescence image and accepting only peaks with values 1.45× greater than the median signal over the cell mask.

Model fitting The gene replication model was fit to the experimental data by maximizing the objective function Eq. 4.17, which was implemented in Cython[196] for fast numerical optimization using differential evolution[175] implemented in the SciPy[194]. Uncertainty calculations using bootstrapping were performed on NCSA Blue Waters.

4.4.3 Simulations

All simulations were performed using LM v2.3a on a local cluster consisting of three Cirrascale GB5600 Multi-GPU nodes, two equipped with 8 NVIDIA GeForce GTX TITAN X GPU, and one equipped with 4 NVIDIA Tesla K80 GPU. Analysis of simulation data was performed in the Jupyter environment[193] using the SciPy stack. LM v2.3a expands the capability of the GPU-based multi-particle diffusion RDME (MPD-RDME) algorithm[28] by adding support for extended capacity lattices where sixteen different particles may occupy each lattice site. Previous versions allowed up to eight particles per site. When more particles occupy a lattice site than capacity allows, the extra particles are said to have “overflowed” and special handling is required to rectify the situation. A procedure on the CPU locates candidate neighboring lattice sites and moves the excess particles into them. This is costly, as the lattice must be copied to host memory and then back to the GPU after overflows are corrected. Additionally, a higher capacity lattice incurs a cost as well, as the diffusion and reaction operators must access a larger amount of memory to account for the greater number of particles. However, simulations that experience overflows on a frequent basis benefit from the greater capacity, as the cost of accessing more memory is offset by the savings gained from not needing to perform overflow handling.

RDME Running a single replicate per GPU, the simulations completed 120 minutes of simulated time in 29 hours on the TITAN X nodes at a time step of 25 μs[39]. Cell growth and DNA replication was implemented using a custom RDME solver using pyLM[38], whereby the lattice of site types was modified *in situ* every 2000 time steps. Initial species counts were taken from a chemical master equation based simulation of the growth ribosome biogenesis model at steady state. Operon placement was performed by estimating the position of a locus along the cell axis assuming that the chromosome is organized linearly. The operon position in the cross-sectional plane of the cells is distributed uniformly within the nucleoid region.

CME Cell growth and DNA replication was implemented using a custom chemical master equation solver implemented using pyLM[38]. CME simulations of the SAM,

$$D_i(t) \xrightarrow{k_t} D_i(t) + m_i \tag{4.28a}$$

$$m_i \underset{k_u}{\overset{k_b}{\rightleftharpoons}} n_i \tag{4.28b}$$

$$m_i \xrightarrow{k_d} \varnothing \tag{4.28c}$$

were used to validate the semi-analytic theory derived in this paper by varying the number of genes $c = ||\boldsymbol{D}||$ from 10 to 500 and the position of the gene between the *oriC* and *terC* in increments of 10%. In each simulation, c identical genes were produced. Each gene is associated with three species in the simulation, a gene (D_i) that is transcribed to produce unbound messengers (m_i) which can bind and unbind to a ribosome to become sequestered messengers (n_i). The rates transcription, degradation, ribosome binding and unbinding rates (k_t, k_d, k_b, and k_u, respectively) were taken to be identical for each gene. Simulations of 100 replicate cells growing for 11 generations were performed to acquire convergent statistics. Each cell was seeded with identical initial conditions; therefore, the first generation was excluded when computing statistics (therefore, each average was over 100 replicate cells each growing for 10 generations). Each gene (D_i) was replicated according to the fitted replication start (t_s) and replication (T_r) times, and its position along the genome. Cell division was performed every $t_D = 120$ minutes with cell components binomially distributed between daughter cells. Only a single daughter cell was followed after each cell division event. To allow comparison with theory a single set of rate were used for all genes, namely $k_t = 0.0042\mathrm{s}^{-1}$, $k_b = 0.079\mathrm{s}^{-1}$, $k_u = 0.008\mathrm{s}^{-1}$ and $k_d = 9.84 \times 10^{-4}\mathrm{s}^{-1}$.

Chapter 5

Conclusions

5.1 Summary

The three-state stochastic model of the *lac* switch (Chapter 2) has shown the necessity of a third transcriptional state of the operon in producing bistable behavior. Our novel computational technique, the geometric bursting approximation, coupled with the finite state projection (FSP) has allowed for a sufficiently fast model solution such that an exhaustive search of the model parameter space could be made, while maintaining high accuracy in the solution to the chemical master equation (CME). Our results appears to have inspired further analytical work on the three state *lac* switch. Choudhary et al. [197] have computed analytical protein distributions using constant switching rates between the transcriptional states.

Using kinetic data for the binding of ribosomal protein (r-protein) to partially assembled intermediates, we were able to construct a kinetic model which accounts for the kinetic cooperativity of r-protein binding. We were able to reproduce the well-known $5' \rightarrow \text{central} \rightarrow 3'$ order of assembly, as well as a predict a secondary assembly pathway progressing via $5' \rightarrow 3' \rightarrow \text{central}$. Since this data was acquired at 15 °C, it was unsuitable for an *in vivo* model of *Escherichia coli*. Instead, we used kinetic data taken at 40 °C to construct an assembly model appropriate to include in the whole cell simulation, which reproduced the same binding timescales for all proteins measured in *in vitro* studies and predicted assembly intermediates observed through cryo-electron microscopy (cryo-EM). These *in vitro* models of the assembly of the small subunit (SSU) are the first of their kind. The procedural way that the reaction networks are constructed is easily adaptable to other systems, all that is needed is a description of an assembly hierarchy and kinetic data which can be compared to the solution to the deterministic rate equations defined by the generated reaction network. This will allow for the rapid construction of a kinetic model of the assembly of the large subunit (LSU), should the necessary data become available.

Using the 40 °C assembly model and a simplified model of transcription and translation, we constructed the largest and most detailed computational model of a whole-cell to date. The cellular environment was constructed using data from cryo-electron tomography and single particle tracking experiments to approximate slow growing (120 min doubling time) *E. coli* with a densely packed nucleoid region that excludes ribosomes. Although the assembly model was developed from experiments performed *in vitro*, with the increased cellular concentrations of r-protein it yielded SSU assembly times comparable to experiments performed *in vivo*. Using this model, we predicted non-uniform spatial distributions of messenger RNA (mRNA) and early 30S intermediates.

The whole-cell ribosome biogenesis model was then improved further to include cell growth and gene replication. We developed a simple analytical model describing growth and DNA replication in slow-growing *E. coli* which does not require the explicit tracking of cell lineages, and applied it to experimental data to estimate the cell cycle parameters. These parameters were then used to determine the cell geometry immediately after division, replication initiation time, and duration of replication, which are the parameters which describe the growing cell. With the addition of gene replication, the whole-cell model did not show the expected copy number statistics predicted from analytical mRNA expression models in the presence of gene replication[1,164]. The origin of this discrepancy was identified as a lack of competition between the messengers for the ribosomes.

5.2 Outlook

Though the ribosome biogenesis model is unprecedentedly complete, it is still lacking in many ways. Before the model can be used as a viable platform to construct genome-scale models, these deficiencies must be addressed. In order to hook ribosome production into other cellular systems, such as core metabolism, we must include gene regulation of the ribosomal operons. There are three mechanisms of regulation which will be necessary to include. The most trivial regulation to add is translational inhibition arising from r-protein binding to its own mRNA, which shuts down production if there are not enough intermediates present for the r-protein to associate with[98,198]. This will lift the artificial tuning of r-protein transcription rates. Second is transcriptional inhibition of ribosomal RNA (rRNA) operons through guanosine tetraphosphate (ppGpp) during amino acid starvation[98,199]. ppGpp is produced by RelA, which is associated with approximately 0.5% of the ribosomes in the cell. When an uncharged transfer RNA enters the A site of the ribosome, the ribosome-bound RelA is triggered and produces ppGpp. This will be complicated to implement, since it would require accounting of the available amino acid pool in the model. Finally, rRNA transcription is upregulated through the nucleoid-associated protein, Fis, whose production is upregulated in nutrient-rich conditions and strongly downregulated in stationary-phase cells[199]. Though regulation of *fis* expression is complicated as well[200], it appears to be an important link between ribosome production and metabolism.

The problem of insufficient mRNA to fully utilize the ribosomes should be alleviated automatically once metabolism is integrated, however in the meantime it can be corrected through the addition of "silent" mRNA. These species will be placeholders for the messengers in the cell which produce proteins that are not included in the model. By using transcriptomics data, we can determine the necessary number of silent mRNA to include as well as their translation rate which would be used to determine how long the ribosome will be unavailable for translating messengers described by the ribosome biogenesis model.

Finally, the largest issue with this model is the lack of non-reactive intermolecular interactions. Currently, the diffusive behavior of species is determined only through their diffusion constants, which can vary across cellular compartments. However the presence of molecular crowders can impact the spatial distribution of biomolecules. For instance, it is currently possible for multiple ribosomes to occupy the same lattice site in spite of the fact that a ribosome is ~20 nm in diameter compared to the ~32 nm lattice spacing—effectively an ideal gas of ribosomes. This means that the spatial heterogeneity of ribosomes are not being realistically modeled in these simulations. Other spatially resolved stochastic simulation software such as Smoldyn[25] use Brownian dynamics to treat diffusion, which automatically includes excluded volume interactions. However these methods are not fast enough to reach cell-cycle timescales, and genome-scale system sizes. To go forward, we must augment our multi-particle diffusion RDME (MPD-RDME) algorithm to somehow include these effects. One possibility is to keep track of the occupied volume in each lattice site due to molecules, and compute the diffusive propensity based on the occupied volume fraction of the originating and target lattice sites.[201–203] However in using such a coarse-grained treatment, one must rigorously derive the transition probabilities and compare to Brownian dynamics simulations to ensure that the approximation is valid.

With the addition excluded volume interactions, a realistic physical model of the conformation of the chromosome can be added. Currently in development is a GPU-based Brownian dynamics code to fold realistic conformations of the full, 4.64 Mbp *E. coli* genome. The DNA is represented as a beads-on-a-string model, coarse-graining to 10 base pairs per bead. The beads interact through bond potentials and an Lennard-Jones potential. The forces involved in the bonded interactions are stretching, bending, and torsion. The force parameters are chosen to reproduce the linear (~50 nm) and twist (~100 nm) persistence lengths of DNA. The integrator and force field are similar to that used by Chirico and Langowski[204] and Klenin et al.[205]. Integration is performed using an Euler-Maruyama scheme,

$$\boldsymbol{x}(t+\tau) = \boldsymbol{x}(t) + \frac{1}{\zeta}\boldsymbol{F}(\boldsymbol{x}(t))\tau + \sqrt{\frac{2k_\mathrm{B}T}{\zeta}}\boldsymbol{\eta}(t)\sqrt{\tau} \tag{5.1}$$

where ζ is the friction constant and η_i is a Gaussian random variable with zero mean and unit

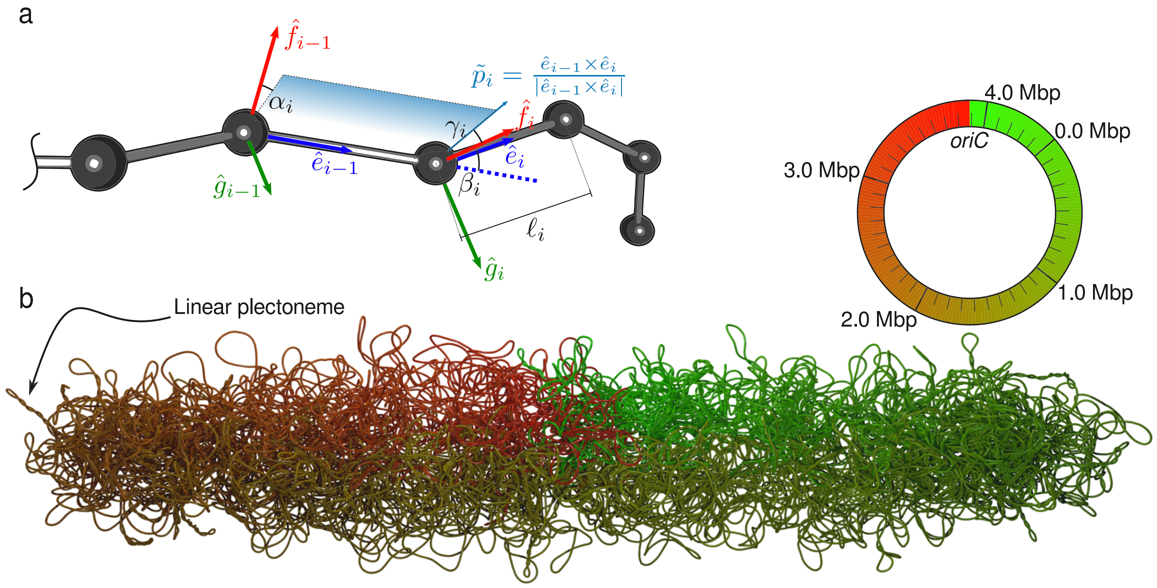


Figure 5.1 (a) Description of force field used in Brownian dynamics simulations used to fold the *E. coli* chromosome. (b) Folded *E. coli* chromosome. The model consists of 464 000 beads, with each bead representing 10 base pairs. The first (green) and last (beads) are part of the origin of replication (*oriC*), which is placed at the midpoint of the cell. Linear plectonemes form in the structure due to the torsion interaction and negative supercoiling present in the model.

variance. The potentials from which the force $\boldsymbol{F}(\boldsymbol{x}(t))$ is computed are

$$U_i^{(\mathrm{st})} = \frac{1}{2}k_{\mathrm{st}}(\ell_i - \bar{\ell})^2 \qquad \text{(bond stretching)} \qquad (5.2\mathrm{a})$$

$$U_i^{(\mathrm{bn})} = \frac{1}{2}k_{\mathrm{bn}}{\beta_i}^2 \qquad \text{(bond bending)} \qquad (5.2\mathrm{b})$$

$$U_i^{(\mathrm{tr})} = \frac{1}{2}k_{\mathrm{tr}}(\alpha_i + \gamma_i)^2 \qquad \text{(bond torsion)} \qquad (5.2\mathrm{c})$$

$$U_{ij}^{(\mathrm{ev})} = 4\epsilon\left[\left(\frac{\sigma}{r_{ij}}\right)^{12} - \left(\frac{\sigma}{r_{ij}}\right)^{6}\right] \qquad \text{(excluded volume)} \qquad (5.2\mathrm{d})$$

where α_i, β_i, and γ_i are the Euler angles transforming the coordinate system $(\hat{\boldsymbol{f}}_{i-1}, \hat{\boldsymbol{g}}_{i-1}, \hat{\boldsymbol{e}}_{i-1})$ centered on bead $i-1$ to the coordinate system $(\hat{\boldsymbol{f}}_i, \hat{\boldsymbol{g}}_i, \hat{\boldsymbol{e}}_i)$ centered on bead i (see Figure 5.1a for a depiction of the geometry).

We can currently construct structures of the folded genome by slowly inserting beads at random locations in the ring polymer until the necessary polymer size has been reached. After a prescribed number of time steps, $\lceil \chi n(t) \rceil$ are inserted into random bond locations, and the bond topology is recomputed. By using a fraction χ of the total number of beads at that time $n(t)$ the growth can proceed exponentially. This prevents problems that arise due to adding a constant number, such as instability (too many beads added at once) or slow growth times (too few). Figure 5.1b shows an example of the structure that is generated using this technique. The chromosome shows the expected linear organization[139] and forms plectonemes. Currently, the force constants and friction coefficients have not been calibrated to produce realistic DNA dynamics, however this will not be difficult to accomplish once the simulation code has been effectively optimized. We will later augment this model with chromosome capture data in order to generate structures which reproduce the loci–loci contact probabilities measured experimentally[206].

In spite of these issues, the whole-cell model is predictive and represents the amalgamation of theoretical biological knowledge and data from many disparate experiments into a cohesive whole. Through the combination of this information, we have a framework from which further questions can be answered through the augmentation of the model and perturbation of the parameters. I hope that it will find use in the future as the computational simulation of living matter comes of age.

References

1. Peterson JR, Cole JA, Fei J, Ha T, and Luthey-Schulten ZA (2015). Effects of DNA replication on mRNA noise. *Proc. Natl. Acad. Sci. USA*, **112**(52), pp. 15886–15891. doi:10.1073/pnas.1516246112.

2. Liu C, McKinney MC, Chen YH, Earnest TM, Shi X, Lin LJ, Ishino Y, Dahmen K, Cann IKO, and Ha T (2011). Reverse-chaperoning activity of an AAA+ protein. *Biophys. J.*, **100**(5), pp. 1344–1352. doi:10.1016/j.bpj.2011.01.057.

3. Indiani C and O'Donnell M (2006). The replication clamp-loading machine at work in the three domains of life. *Nat. Rev. Mol. Cell. Biol.*, **7**(10), pp. 751–761. doi:10.1038/nrm2022.

4. Brown KS and Sethna JP (2003). Statistical mechanical approaches to models with many poorly known parameters. *Phys. Rev. E*, **68**, p. 021904. doi:10.1103/PhysRevE.68.021904.

5. Kirkpatrick S, Gelatt CD, and Vecchi MP (1983). Optimization by simulated annealing. *Science*, **220**(4598), pp. 671–680. doi:10.1126/science.220.4598.671.

6. Nelder JA and Mead R (1965). A simplex method for function minimization. *Comput. J.*, **7**(4), pp. 308–313. doi:10.1093/comjnl/7.4.308.

7. Gillespie DT (1992). A rigorous derivation of the chemical master equation. *Physica A*, **188**(1-3), pp. 404–425. doi:10.1016/0378-4371(92)90283-v.

8. Gillespie DT (1977). Exact stochastic simulation of coupled chemical reactions. *J. Phys. Chem.*, **81**(25), pp. 2340–2361. doi:10.1021/j100540a008.

9. Gillespie DT (1976). A general method for numerically simulating the stochastic time evolution of coupled chemical reactions. *J. Comp. Phys.*, **22**(4), pp. 403–434. doi:10.1016/0021-9991(76)90041-3.

10. Gibson MA and Bruck J (2000). Efficient exact stochastic simulation of chemical systems with many species and many channels. *J. Phys. Chem. A*, **104**(9), pp. 1876–1889. doi:10.1021/jp993732q.

11. Ramaswamy R, Gonzaĺez-Segredo N, and Sbalzarini IF (2009). A new class of highly efficient exact stochastic simulation algorithms for chemical reaction networks. *J. Chem. Phys.*, **130**(24), p. 244104. doi:10.1063/1.3154624.

12. Indurkhya S and Beal J (2010). Reaction factoring and bipartite update graphs accelerate the gillespie algorithm for large-scale biochemical systems. *PLoS ONE*, **5**(1), p. e8125. doi:10.1371/journal.pone.0008125.

13. Ramaswamy R and Sbalzarini IF (2010). A partial-propensity variant of the composition-rejection stochastic simulation algorithm for chemical reaction networks. *J. Chem. Phys.*, **132**(4), p. 044102. doi:10.1063/1.3297948.

14. Cao Y, Li H, and Petzold L (2004). Efficient formulation of the stochastic simulation algorithm for chemically reacting systems. *J. Chem. Phys.*, **121**(9), p. 4059. doi:10.1063/1.1778376.

15. McCollum JM, Peterson GD, Cox CD, Simpson ML, and Samatova NF (2006). The sorting direct method for stochastic simulation of biochemical systems with varying reaction execution behavior. *Comput. Biol. & Chem.*, **30**(1), pp. 39–49. doi:10.1016/j.compbiolchem.2005.10.007.

16. Gillespie DT (2001). Approximate accelerated stochastic simulation of chemically reacting systems. *J. Chem. Phys.*, **115**(4), p. 1716. doi:10.1063/1.1378322.

17. Rathinam M, Petzold LR, Cao Y, and Gillespie DT (2003). Stiffness in stochastic chemically reacting systems: The implicit tau-leaping method. *J. Chem. Phys.*, **119**(24), p. 12784. doi:10.1063/1.1627296.

18. Cao Y, Gillespie DT, and Petzold LR (2005). The slow-scale stochastic simulation algorithm. *J. Chem. Phys.*, **122**(1), p. 014116. doi:10.1063/1.1824902.

19. Murray JD, *Mathematical Biology (Biomathematics)* (Springer-Verlag, 1993), 2nd ed. ISBN 9783540572046.

20. Vafabakhsh R, Kondabagil K, Earnest T, Lee KS, Zhang Z, Dai L, Dahmen KA, Rao VB, and Ha T (2014). Single-molecule packaging initiation in real time by a viral DNA packaging machine from bacteriophage T4. *Proc. Natl. Acad. Sci. USA*, **111**(42), pp. 15096–15101. doi:10.1073/pnas.1407235111.

21. Sun S, Kondabagil K, Draper B, Alam TI, Bowman VD, Zhang Z, Hegde S, Fokine A, Rossmann1 MG, and Rao VB (2008). The structure of the phage T4 DNA packaging motor suggests a mechanism dependent on electrostatic forces. *Cell*, **135**(7), pp. 1251–1262. doi:10.1016/j.cell.2008.11.015.

22. Kottadiel VI, Rao VB, and Chemla YR (2012). The dynamic pause-unpackaging state, an off-translocation recovery state of a DNA packaging motor from bacteriophage T4. *Proc. Natl. Acad. Sci. USA*, **109**(49), pp. 20000–20005. doi:10.1073/pnas.1209214109.

23. Munsky B and Khammash M (2006). The finite state projection algorithm for the solution of the chemical master equation. *J. Chem. Phys.*, **124**(4), 044104. doi:10.1063/1.2145882.

24. MacNamara S, Burrage K, and Sidje R (2008). Multiscale modeling of chemical kinetics via the master equation. *Multiscale Model. Simul.*, **6**(4), pp. 1146–1168. doi:10.1137/060678154.

25. Andrews SS, Addy NJ, Brent R, and Arkin AP (2010). Detailed simulations of cell biology with Smoldyn 2.1. *PLoS Comput. Biol.*, **6**(3), p. e1000705. doi:10.1371/journal.pcbi.1000705.

26. Kerr RA, Bartol TM, Kaminsky B, Dittrich M, Chang JCJ, Baden SB, Sejnowski TJ, and Stiles JR (2008). Fast monte carlo simulation methods for biological reaction–diffusion systems in solution and on surfaces. *SIAM J. Sci. Comput.*, **30**(6), pp. 3126–3149. doi:10.1137/070692017.

27. Schöneberg J and Noé F (2013). ReaDDy - a software for particle-based reaction–diffusion dynamics in crowded cellular environments. *PLoS ONE*, **8**(9), p. e74261. doi:10.1371/journal.pone.0074261.

28. Roberts E, Stone JE, and Luthey-Schulten Z (2013). Lattice Microbes: high-performance stochastic simulation method for the reaction–diffusion master equation. *J. Comp. Chem.*, **34**(3), pp. 245–255. doi:10.1002/jcc.23130.

29. Hallock MJ, Stone JE, Roberts E, Fry C, and Luthey-Schulten Z (2014). Simulation of reaction diffusion processes over biologically relevant size and time scales using multi-GPU workstations. *Parall. Comp.*, **40**(5-6), pp. 86–99. doi:10.1016/j.parco.2014.03.009.

30. Hattne J, Fange D, and Elf J (2005). Stochastic reaction–diffusion simulation with MesoRD. *Bioinformatics*, **21**(12), pp. 2923–2924. doi:10.1093/bioinformatics/bti431.

31. Drawert B, Engblom S, and Hellander A (2012). URDME: a modular framework for stochastic simulation of reaction-transport processes in complex geometries. *BMC Syst. Biol.*, **6**(1), p. 76. doi:10.1186/1752-0509-6-76.

32. Roberts E, Stone JE, Sepulveda L, Hwu WMW, and Luthey-Schulten Z, Long time-scale simulations of in vivo diffusion using GPU hardware. In *Parallel Distributed Processing, 2009. IPDPS 2009. IEEE International Symposium on*, pp. 1–8 (2009). doi:10.1109/IPDPS.2009.5160930.

33. Roberts E, Magis A, Ortiz JO, Baumeister W, and Luthey-Schulten Z (2011). Noise contributions in an inducible genetic switch: A whole-cell simulation study. *PLoS Comput. Biol.*, **7**(3), p. e1002010. doi:10.1371/journal.pcbi.1002010.

34. Isaacson SA (2009). The reaction–diffusion master equation as an asymptotic approximation of diffusion to a small target. *SIAM J. Appl. Math.*, **70**(1), pp. 77–111. doi:10.1137/070705039.

35. Isaacson SA and Isaacson D (2009). Reaction–diffusion master equation, diffusion-limited reactions, and singular potentials. *Phys. Rev. E*, **80**(6). doi:10.1103/physreve.80.066106.

36. Erban R and Chapman SJ (2009). Stochastic modelling of reaction–diffusion processes: algorithms for bimolecular reactions. *Phys. Biol.*, **6**(4), p. 046001. doi:10.1088/1478-3975/6/4/046001.

37. Elf J and Ehrenberg M (2004). Spontaneous separation of bi-stable biochemical systems into spatial domains of opposite phases. *Syst. Biol.*, **1**(2), pp. 230–236. doi:10.1049/sb:20045021.

38. Peterson JR, Hallock MJ, Cole JA, and Luthey-Schulten ZA, A problem solving environment for stochastic biological simulations. In *PyHPC 2013*, Supercomputing 2013 (2013).

39. Hallock MJ and Luthey-Schulten Z, Improving reaction kernel performance in Lattice Microbes: particle-wise propensities and run-time generated code. In *Parallel and Distributed Processing Symposium Workshop (IPDPSW), 2016 IEEE International* (2016). Accepted.

40. Earnest TM, Roberts E, Assaf M, Dahmen K, and Luthey-Schulten Z (2013). DNA looping increases the range of bistability in a stochastic model of the lac genetic switch. *Phys. Biol.*, **10**(2), p. 026002. doi:10.1088/1478-3975/10/2/026002.

41. Novick A and Weiner M (1957). Enzyme induction as an all-or-none phenomenon. *Proc. Natl. Acad. Sci. USA*, **43**(7), pp. 553–566. doi:10.1073/pnas.43.7.553.

42. Jacob F and Monod J (1961). Genetic regulatory mechanisms in the synthesis of proteins. *J. Mol. Biol.*, **3**(3), pp. 318–356. doi:10.1016/b978-0-12-460482-7.50042-7.

43. Choi PJ, Cai L, Frieda K, and Xie XS (2008). A stochastic single-molecule event triggers phenotype switching of a bacterial cell. *Science*, **322**(5900), pp. 442–446. doi:10.1126/science.1161427.

44. Oehler S, Alberti S, and Müller-Hill B (2006). Induction of the lac promoter in the absence of DNA loops and the stoichiometry of induction. *Nucl. Acids Res.*, **34**(2), pp. 606–12. doi:10.1093/nar/gkj453.

45. Vilar JMG and Saiz L (2005). DNA looping in gene regulation: from the assembly of macromolecular complexes to the control of transcriptional noise. *Curr. Op. Genet. Dev.*, **15**(2), pp. 136–44. doi:10.1016/j.gde.2005.02.005.

46. Oehler S, Amouyal M, Kolkhof P, von Wilcken-Bergmann B, and Müller-Hill B (1994). Quality and position of the three lac operators of E. coli define efficiency of repression. *EMBO J.*, **13**(14), pp. 3348–3355.

47. Acar M, Mettetal JT, and van Oudenaarden A (2008). Stochastic switching as a survival strategy in fluctuating environments. *Nat. Genet.*, **40**(4), pp. 471–5. doi:10.1038/ng.110.

48. Dekel E and Alon U (2005). Optimality and evolutionary tuning of the expression level of a protein. *Nature*, **436**(7050), pp. 588–92. doi:10.1038/nature03842.

49. Thattai M (2004). Stochastic gene expression in fluctuating environments. *Genetics*, **167**(1), pp. 523–530. doi:10.1534/genetics.167.1.523.

50. Ozbudak EM, Thattai M, Lim HN, Shraiman BI, and van Oudenaarden A (2004). Multistability in the lactose utilization network of Escherichia coli. *Nature*, **427**(6976), pp. 737–40. doi:10.1038/nature02298.

51. Matthews KS (1992). DNA looping. *Microbiol. Rev.*, **56**(1), pp. 123–136.

52. Schleif R (1992). DNA looping. *Ann. Rev. Biochem.*, **61**(1), pp. 199–223. doi:10.1126/science.3353710.

53. Choi PJ, Xie XS, and Shakhnovich EI (2010). Stochastic switching in gene networks can occur by a single-molecule event or many molecular steps. *J. Mol. Biol.*, **396**(1), pp. 230–244. doi:10.1016/j.jmb.2009.11.035.

54. Stamatakis M and Mantzaris NV (2009). Comparison of deterministic and stochastic models of the lac operon genetic network. *Biophys. J.*, **96**(3), pp. 887–906. doi:10.1016/j.bpj.2008.10.028.

55. Mettetal JT, Muzzey D, Pedraza JM, Ozbudak EM, and van Oudenaarden A (2006). Predicting stochastic gene expression dynamics in single cells. *Proc. Natl. Acad. Sci. USA*, **103**(19), pp. 7304–7309. doi:10.1073/pnas.0509874103.

56. Vilar JMG, Guet CC, and Leibler S (2003). Modeling network dynamics: the lac operon, a case study. *J. Cell Biol.*, **161**(3), pp. 471–6. doi:10.1083/jcb.200301125.

57. Yildirim N and Mackey MC (2003). Feedback regulation in the lactose operon: a mathematical modeling study and comparison with experimental data. *Biophys. J.*, **84**(5), pp. 2841–51. doi:10.1016/s0006-3495(03)70013-7.

58. Wong P, Gladney S, and Keasling JD (1997). Mathematical model of the lac operon: inducer exclusion, catabolite repression, and diauxic growth on glucose and lactose. *Biotechnol. Prog.*, **13**(2), pp. 132–43. doi:10.1021/bp970003o.

59. Pressé S, Ghosh K, and Dill KA (2011). Modeling stochastic dynamics in biochemical systems with feedback using Maximum Caliber. *J. Phys. Chem. B*, **115**(19), pp. 6202–6212. doi:10.1021/jp111112s.

60. Pressé S, Ghosh K, Phillips R, and Dill KA (2010). Dynamical fluctuations in biochemical reactions and cycles. *Phys. Rev. E*, **82**, p. 031905. doi:10.1103/physreve.82.031905.

61. Raser JM and O'Shea EK (2005). Noise in gene expression: Origins, consequences, and control. *Science*, **309**(5743), pp. 2010–2013. doi:10.1126/science.1105891.

62. Elowitz MB, Levine AJ, Siggia ED, and Swain PS (2002). Stochastic gene expression in a single cell. *Science*, **297**(5584), pp. 1183–1186. doi:10.1126/science.1070919.

63. McAdams HH and Arkin A (1997). Stochastic mechanisms in gene expression. *Proc. Natl. Acad. Sci. USA*, **94**(3), pp. 814–9. doi:10.1073/pnas.94.3.814.

64. Assaf M, Roberts E, and Luthey-Schulten Z (2011). Determining the stability of genetic switches: Explicitly accounting for mRNA noise. *Phys. Rev. Lett.*, **106**, p. 248102. doi:10.1103/physrevlett.106.248102.

65. Shahrezaei V and Swain PS (2008). Analytical distributions for stochastic gene expression. *Proc. Natl. Acad. Sci. USA*, **105**(45), pp. 17256–17261. doi:10.1073/pnas.0803850105.

66. Friedman N, Cai L, and Xie XS (2006). Linking stochastic dynamics to population distribution: An analytical framework of gene expression. *Phys. Rev. Lett.*, **97**(16). doi:10.1103/physrevlett.97.168302.

67. Hornos JEM, Schultz D, Innocentini GCP, Wang J, Walczak AM, Onuchic JN, and Wolynes PG (2005). Self-regulating gene: An exact solution. *Phys. Rev. E*, **72**(5), pp. 051907–. doi:10.1103/physreve.72.051907.

68. Thattai M and van Oudenaarden A (2001). Intrinsic noise in gene regulatory networks. *Proc. Natl. Acad. Sci. USA*, **98**(15), pp. 8614–8619. doi:10.1073/pnas.151588598.

69. Kepler TB and Elston TC (2001). Stochasticity in transcriptional regulation: Origins, consequences, and mathematical representations. *Biophys. J.*, **81**(6), pp. 3116–3136. doi:10.1016/s0006-3495(01)75949-8.

70. Peccoud J and Ycart B (1995). Markovian modeling of gene-product synthesis. *Theor. Pop. Biol.*, **48**(2), pp. 222–234. doi:10.1006/tpbi.1995.1027.

71. Berg OG (1978). A model for the statistical fluctuations of protein numbers in a microbial population. *J. Theor. Biol.*, **71**(4), pp. 587–603. doi:10.1016/0022-5193(78)90326-0.

72. Newby JM (2012). Isolating intrinsic noise sources in a stochastic genetic switch. *Phys. Biol.*, **9**(2), p. 026002. doi:10.1088/1478-3975/9/2/026002.

73. van Kampen NG, *Stochastic Processes in Physics and Chemistry, Third Edition (North-Holland Personal Library)* (North Holland, 2007).

74. Bender CM and Orszag SA, *Advanced Mathematical Methods for Scientists and Engineers: Asymptotic Methods and Perturbation Theory (v. 1)* (Springer, 1999). ISBN 0-387-98931-5. doi:10.1007/978-1-4757-3069-2.

75. Dykman MI, Mori E, Ross J, and Hunt PM (1994). Large fluctuations and optimal paths in chemical kinetics. *J. Chem. Phys.*, **100**(8), pp. 5735–5750. doi:10.1063/1.467139.

76. Assaf M and Meerson B (2010). Extinction of metastable stochastic populations. *Phys. Rev. E*, **81**(2), pp. 021116–. doi:10.1103/physreve.81.021116.

77. Escudero C and Kamenev A (2009). Switching rates of multistep reactions. *Phys. Rev. E*, **79**(4), pp. 041149–. doi:10.1103/physreve.79.041149.

78. Mehta P, Mukhopadhyay R, and Wingreen NS (2008). Exponential sensitivity of noise-driven switching in genetic networks. *Phys. Biol.*, **5**(2), p. 026005. doi:10.1088/1478-3975/5/2/026005.

79. Morelli MJ, Allen RJ, Tanase-Nicola S, and ten Wolde PR (2008). Eliminating fast reactions in stochastic simulations of biochemical networks: A bistable genetic switch. *J. Chem. Phys.*, **128**(4), p. 045105. doi:10.1063/1.2821957.

80. MacNamara S, Bersani AM, Burrage K, and Sidje RB (2008). Stochastic chemical kinetics and the total quasi-steady-state assumption: Application to the stochastic simulation algorithm and chemical master equation. *J. Chem. Phys.*, **129**(9), p. 095105. doi:10.1063/1.2971036.

81. Paulsson J and Ehrenberg M (2000). Random signal fluctuations can reduce random fluctuations in regulated components of chemical regulatory networks. *Phys. Rev. Lett.*, **84**, pp. 5447–5450. doi:10.1103/physrevlett.84.5447.

82. Sánchez A and Kondev J (2008). Transcriptional control of noise in gene expression. *Proc. Natl. Acad. Sci. USA*, **105**(13), pp. 5081–5086. doi:10.1073/pnas.0707904105.

83. Dandach SH and Khammash M (2010). Analysis of stochastic strategies in bacterial competence: A master equation approach. *PLoS Comput. Biol.*, **6**(11), p. e1000985. doi:10.1371/journal.pcbi.1000985.

84. Munsky B and Khammash M (2007). A multiple time interval finite state projection algorithm for the solution to the chemical master equation. *J. Comp. Phys.*, **226**(1), pp. 818–835. doi:10.1016/j.jcp.2007.05.016.

85. Sidje RB (1998). Expokit. A software package for computing matrix exponentials. *ACM Trans. Math. Softw.*, **24**(1), pp. 130–156. doi:10.1145/285861.285868.

86. Micheelsen MA, Mitarai N, Sneppen K, and Dodd IB (2010). Theory for the stability and regulation of epigenetic landscapes. *Phys. Biol.*, **7**(2), p. 026010. doi:10.1088/1478-3975/7/2/026010.

87. Walczak AM, Onuchic JN, and Wolynes PG (2005). Absolute rate theories of epigenetic stability. *Proc. Natl. Acad. Sci. USA*, **102**(52), pp. 18926–18931. doi:10.1073/pnas.0509547102.

88. Aurell E and Sneppen K (2002). Epigenetics as a first exit problem. *Phys. Rev. Lett.*, **88**(4), pp. 048101–. doi:10.1103/physrevlett.88.048101.

89. Munsky B and Khammash M (2008). Transient analysis of stochastic switches and trajectories with applications to gene regulatory networks. *IET Syst. Biol.*, **2**(5), pp. 323–333. doi:10.1049/iet-syb:20070082.

90. Eismann ER and Müller-Hill B (1990). lac repressor forms stable loops in vitro with supercoiled wild-type lac DNA containing all three natural lac operators. *J. Mol. Biol.*, **213**(4), pp. 763–775. doi:10.1016/s0022-2836(05)80262-1.

91. Vilar J and Leibler S (2003). DNA looping and physical constraints on transcription regulation. *J. Mol. Biol.*, **331**(5), pp. 981–989. doi:10.1016/s0022-2836(03)00764-2.

92. Johnson S, Lindén M, and Phillips R (2012). Sequence dependence of transcription factor-mediated DNA looping. *Nucl. Acids Res.*, **40**(16), pp. 7728–38. doi:10.1093/nar/gks473.

93. Haeusler AR, Goodson KA, Lillian TD, Wang X, Goyal S, Perkins NC, and Kahn JD (2012). FRET studies of a landscape of lac repressor-mediated DNA loops. *Nucl. Acids Res.*, **40**(10), pp. 4432–4445. doi:10.1093/nar/gks019.

94. Earnest TM, Lai J, Chen K, Hallock MJ, Williamson JR, and Luthey-Schulten Z (2015). Toward a whole-cell model of ribosome biogenesis: Kinetic modeling of SSU assembly. *Biophys. J.*, **109**(6), pp. 1117–1135. doi:10.1016/j.bpj.2015.07.030.

95. Fox GE, Magrum LJ, Balch WE, Wolfe RS, and Woese CR (1977). Classification of methanogenic bacteria by 16S ribosomal RNA characterization. *Proc. Natl. Acad. Sci. USA*, **74**(10), pp. 4537–4541. doi:10.1073/pnas.74.10.4537.

96. Roberts E, Sethi A, Montoya J, Woese CR, and Luthey-Schulten Z (2008). Molecular signatures of ribosomal evolution. *Proc. Natl. Acad. Sci. USA*, **105**(37), pp. 13953–13958. doi:10.1073/pnas.0804861105.

97. Ban N, Beckmann R, Cate JH, Dinman JD, Dragon F, Ellis SR, Lafontaine DL, Lindahl L, Liljas A, Lipton JM, McAlear MA, Moore PB, Noller HF, Ortega J, Panse VG, Ramakrishnan V, Spahn CM, Steitz TA, Tchorzewski M, Tollervey D, Warren AJ, Williamson JR, Wilson D, Yonath A, and Yusupov M (2014). A new system for naming ribosomal proteins. *Curr. Op. Struct. Biol.*, **24**, pp. 165–169. doi:10.1016/j.sbi.2014.01.002.

98. Kaczanowska M and Rydén-Aulin M (2007). Ribosome biogenesis and the translation process in Escherichia coli. *Microbiol. Mol. Biol. Rev.*, **71**(3), pp. 477–494. doi:10.1128/mmbr.00013-07.

99. Bremer H and Dennis PP, Modulation of chemical composition and other parameters of the cell by growth rate. In FC Neidhardt, R Curtiss III, JL Ingraham, ECC Lin, KB Low, B Magasanik, WS Reznikoff, M Riley, M Schaechter, and HE Umbarger, eds., *Escherichia coli and Salmonella typhimurium: Cellular and Molecular Biology*, pp. 1553–1569 (ASM Press, Washington, DC, 1996), 2nd ed. doi:10.1128/ecosal.5.2.3.

100. Liebermeister W, Noor E, Flamholz A, Davidi D, Bernhardt J, and Milo R (2014). Visual account of protein investment in cellular functions. *Proc. Natl. Acad. Sci. USA*, **111**(23), pp. 8488–8493. doi:10.1073/pnas.1314810111.

101. Held WA, Ballou B, Mizushima S, and Nomura M (1974). Assembly mapping of 30S ribosomal proteins from Escherichia coli: Further studies. *J. Biol. Chem.*, **249**(10), pp. 3103–3111.

102. Adilakshmi T, Ramaswamy P, and Woodson SA (2005). Protein-independent folding pathway of the 16S rRNA 5′ domain. *J. Mol. Biol.*, **351**(3), pp. 508–519. doi:10.1016/j.jmb.2005.06.020.

103. Adilakshmi T, Bellur DL, and a Woodson S (2008). Concurrent nucleation of 16S folding and induced fit in 30S ribosome assembly. *Nature*, **455**(7217), pp. 1268–72. doi:10.1038/nature07298.

104. Kim H, Abeysirigunawarden SC, Chen K, Mayerle M, Ragunathan K, Luthey-Schulten Z, Ha T, and Woodson SA (2014). Protein-guided RNA dynamics during early ribosome assembly. *Nature*, **506**(7488), pp. 334–338. doi:10.1038/nature13039.

105. Talkington MWT, Siuzdak G, and Williamson JR (2005). An assembly landscape for the 30S ribosomal subunit. *Nature*, **438**(7068), pp. 628–632. doi:10.1038/nature04261.

106. Sykes MT and Williamson JR (2009). A complex assembly landscape for the 30S ribosomal subunit. *Ann. Rev. Biochem.*, **38**, pp. 197–215. doi:10.1146/annurev.biophys.050708.133615.

107. Bunner AE, Beck AH, and Williamson JR (2010). Kinetic cooperativity in Escherichia coli 30S ribosomal subunit reconstitution reveals additional complexity in the assembly landscape. *Proc. Natl. Acad. Sci. USA*, **107**(12), pp. 5417–5422. doi:10.1073/pnas.0912007107.

108. Mulder AM, Yoshioka C, Beck AH, Bunner AE, Milligan RA, Potter CS, Carragher B, and Williamson JR (2010). Visualizing ribosome biogenesis: Parallel assembly pathways for the 30S subunit. *Science*, **330**(6004), pp. 673–677. doi:10.1126/science.1193220.

109. Sashital DG, Greeman CA, Lyumkis D, Potter CS, Carragher B, and Williamson JR (2014). A combined quantitative mass spectrometry and electron microscopy analysis of ribosomal 30S subunit assembly in E. coli. *eLife*, **3**. doi:10.7554/elife.04491.

110. Wang W, Li GW, Chen C, Xie XS, and Zhuang X (2011). Chromosome organization by a nucleoid-associated protein in live bacteria. *Science*, **333**(6048), pp. 1445–1449. doi:10.1126/science.1204697.

111. Bakshi S, Siryaporn A, Goulian M, and Weisshaar JC (2012). Superresolution imaging of ribosomes and RNA polymerase in live Escherichia coli cells. *Mol. Microbiol.*, **85**(1), pp. 21–38. doi:10.1111/j.1365-2958.2012.08081.x.

112. Bakshi S, Choi H, Mondal J, and Weisshaar JC (2014). Time-dependent effects of transcription- and translation-halting drugs on the spatial distributions of the Escherichia coli chromosome and ribosomes. *Mol. Microbiol.*, **94**(4), pp. 871–887. doi:10.1111/mmi.12805.

113. Sanamrad A, Persson F, Lundius EG, Fange D, Gynna AH, and Elf J (2014). Single-particle tracking reveals that free ribosomal subunits are not excluded from the Escherichia coli nucleoid. *Proc. Natl. Acad. Sci. USA*, **111**(31), pp. 11413–11418. doi:10.1073/pnas.1411558111.

114. Taniguchi Y, Choi PJ, Li GW, Chen H, Babu M, Hearn J, Emili A, and Xie XS (2010). Quantifying E. coli proteome and transcriptome with single-molecule sensitivity in single cells. *Science*, **329**(5991), pp. 533–538. doi:10.1126/science.1188308.

115. Mahmutovic A, Fange D, Berg OG, and Elf J (2012). Lost in presumption: stochastic reactions in spatial models. *Nat. Methods*, **9**(12), pp. 1163–1166. doi:10.1038/nmeth.2253.

116. Serban R and Hindmarsh AC, CVODES: The sensitivity-enabled ODE solver in SUNDIALS. In *Volume 6: 5th International Conference on Multibody Systems, Nonlinear Dynamics, and Control, Parts A, B, and C* (ASME, 2005). doi:10.1115/detc2005-85597.

117. Liu DC and Nocedal J (1989). On the limited memory BFGS method for large scale optimization. *Math. Program.*, **45**(1-3), pp. 503–528. doi:10.1007/bf01589116.

118. Benson DA, Cavanaugh M, Clark K, Karsch-Mizrachi I, Lipman DJ, Ostell J, and Sayers EW (2012). GenBank. *Nucl. Acids Res.*, **41**(D1), pp. D36–D42. doi:10.1093/nar/gks1195.

119. Cock PJA, Antao T, Chang JT, Chapman BA, Cox CJ, Dalke A, Friedberg I, Hamelryck T, Kauff F, Wilczynski B, and de Hoon MJL (2009). Biopython: Freely available python tools for computational molecular biology and bioinformatics. *Bioinformatics*, **25**(11), pp. 1422–1423. doi:10.1093/bioinformatics/btp163.

120. Rodriguez JV, Kaandorp JA, Dobrzynski M, and Blom JG (2006). Spatial stochastic modelling of the phosphoenolpyruvate-dependent phosphotransferase (PTS) pathway in Escherichia coli. *Bioinformatics*, **22**(15), pp. 1895–1901. doi:10.1093/bioinformatics/btl271.

121. Lampoudi S, Gillespie DT, and Petzold LR (2009). The multinomial simulation algorithm for discrete stochastic simulation of reaction–diffusion systems. *J. Chem. Phys.*, **130**(9), p. 094104. doi:10.1063/1.3074302.

122. Berk V, Zhang W, Pai RD, and Cate JHD (2006). Structural basis for mRNA and tRNA positioning on the ribosome. *Proc. Natl. Acad. Sci. USA*, **103**(43), pp. 15830–15834. doi:10.1073/pnas.0607541103.

123. Best RB, Zhu X, Shim J, Lopes PEM, Mittal J, Feig M, and MacKerell AD (2012). Optimization of the additive CHARMM all-atom protein force field targeting improved sampling of the backbone ϕ, ψ and side-chain χ_1 and χ_2 dihedral angles. *J. Chem. Theor. Comp.*, **8**(9), pp. 3257–3273. doi:10.1021/ct300400x.

124. Denning EJ, Priyakumar UD, Nilsson L, and Mackerell AD (2011). Impact of 2′-hydroxyl sampling on the conformational properties of RNA: Update of the CHARMM all-atom additive force field for RNA. *J. Comp. Chem.*, **32**(9), pp. 1929–1943. doi:10.1002/jcc.21777.

125. Phillips JC, Braun R, Wang W, Gumbart J, Tajkhorshid E, Villa E, Chipot C, Skeel RD, Kale L, and Schulten K (2005). Scalable molecular dynamics with NAMD. *J. Comp. Chem.*, **26**(16), pp. 1781–1802. doi:10.1002/jcc.20289.

126. Recht MI and Williamson JR (2001). Central domain assembly: Thermodynamics and kinetics of S6 and S18 binding to an S15–RNA complex. *J. Mol. Biol.*, **313**(1), pp. 35–48. doi:10.1006/jmbi.2001.5018.

127. Zimmermann RA, Muto A, Fellner P, Ehresmann C, and Branlant C (1972). Location of ribosomal protein binding sites on 16S ribosomal RNA. *Proc. Natl. Acad. Sci. USA*, **69**(5), pp. 1282–1286. doi:10.1073/pnas.69.5.1282.

128. de Narvaez CC and Schaup HW (1979). In vivo transcriptionally coupled assembly of Escherichia coli ribosomal subunits. *J. Mol. Biol.*, **134**(1), pp. 1–22. doi:10.1016/0022-2836(79)90411-x.

129. Powers T, Daubresse G, and Noller HF (1993). Dynamics of in vitro assembly of 16S rRNA into 30S ribosomal subunits. *J. Mol. Biol.*, **232**(2), pp. 362–374. doi:10.1006/jmbi.1993.1396.

130. Chen K, Eargle J, Lai J, Kim H, Abeysirigunawardena S, Mayerle M, Woodson S, Ha T, and Luthey-Schulten Z (2012). Assembly of the five-way junction in the ribosomal small subunit using hybrid MD-gō simulations. *J. Phys. Chem. B*, **116**(23), pp. 6819–6831. doi:10.1021/jp212614b.

131. Lai J, Chen K, and Luthey-Schulten Z (2013). Structural intermediates and folding events in the early assembly of the ribosomal small subunit. *J. Phys. Chem. B*, **117**(42), pp. 13335–13345. doi:10.1021/jp404106r.

132. Gutell RR (2002). Comparative RNA web site and project. `http://www.rna.ccbb.utexas.edu/`.

133. Goss DJ, Parkhurst LJ, and Wahba AJ (1980). Kinetics of ribosome dissociation and subunit association. the role of initiation factor IF3 as an effector. *J. Biol. Chem.*, **255**(1), pp. 225–229. doi:10.1016/s0006-3495(80)84957-5.

134. Studer SM and Joseph S (2006). Unfolding of mRNA secondary structure by the bacterial translation initiation complex. *Mol. Cell*, **22**(1), pp. 105–115. doi:10.1016/j.molcel.2006.02.014.

135. Milon P, Konevega AL, Peske F, Fabbretti A, Gualerzi CO, and Rodnina MV, Transient kinetics, fluorescence, and FRET in studies of initiation of translation in bacteria. In *Methods in Enzymology*, pp. 1–30 (Elsevier BV, 2007). doi:10.1016/s0076-6879(07)30001-3.

136. Kalwarczyk T, Tabaka M, and Holyst R (2012). Biologistics–diffusion coefficients for complete proteome of Escherichia coli. *Bioinformatics*, **28**(22), pp. 2971–2978. doi:10.1093/bioinformatics/bts537.

137. Golding I and Cox EC (2004). RNA dynamics in live Escherichia coli cells. *Proc. Natl. Acad. Sci. USA*, **101**(31), pp. 11310–11315. doi:10.1073/pnas.0404443101.

138. Mandiyan V, Tumminia SJ, Wall JS, Hainfeld JF, and Boublik M (1991). Assembly of the Escherichia coli 30S ribosomal subunit reveals protein-dependent folding of the 16S rRNA domains. *Proc. Natl. Acad. Sci. USA*, **88**(18), pp. 8174–8178.

139. Wiggins PA, Cheveralls KC, Martin JS, Lintner R, and Kondev J (2010). Strong intranucleoid interactions organize the Escherichia coli chromosome into a nucleoid filament. *Proc. Natl. Acad. Sci. USA*, **107**(11), pp. 4991–4995. doi:10.1073/pnas.0912062107.

140. Selinger DW (2003). Global RNA half-life analysis in Escherichia coli reveals positional patterns of transcript degradation. *Genome Res.*, **13**(2), pp. 216–223. doi:10.1101/gr.912603.

141. Northrup SH and Erickson HP (1992). Kinetics of protein-protein association explained by Brownian dynamics computer simulation. *Proc. Natl. Acad. Sci. USA*, **89**(8), pp. 3338–3342. doi:10.1073/pnas.89.8.3338.

142. Lindahl L (1975). Intermediates and time kinetics of the in vivo assembly of Escherichia coli ribosomes. *J. Mol. Biol.*, pp. 15–37. doi:10.1016/0022-2836(75)90089-3.

143. Shajani Z, Sykes MT, and Williamson JR (2011). Assembly of bacterial ribosomes. *Ann. Rev. Biochem.*, **80**, pp. 501–526. doi:10.1146/annurev-biochem-062608-160432.

144. Taghbalout A and Rothfield L (2007). RNaseE and the other constituents of the RNA degradosome are components of the bacterial cytoskeleton. *Proc. Natl. Acad. Sci. USA*, **104**(5), pp. 1667–1672. doi:10.1073/pnas.0610491104.

145. Khemici V, Poljak L, Luisi BF, and Carpousis AJ (2008). The RNase E of Escherichia coli is a membrane-binding protein. *Mol. Microbiol.* doi:10.1111/j.1365-2958.2008.06454.x.

146. Bunner AE, Nord S, Wikström PM, and Williamson JR (2010). The effect of ribosome assembly cofactors on in vitro 30S subunit reconstitution. *J. Mol. Biol.*, **398**(1), pp. 1–7. doi:10.1016/j.jmb.2010.02.036.

147. Connolly K, Rife JP, and Culver G (2008). Mechanistic insight into the ribosome biogenesis functions of the ancient protein KsgA. *Mol. Microbiol.*, **70**(5), pp. 1062–1075. doi:10.1111/j.1365-2958.2008.06485.x.

148. Nevo-Dinur K, Nussbaum-Shochat A, Ben-Yehuda S, and Amster-Choder O (2011). Translation-independent localization of mRNA in E. coli. *Science*, **331**(6020), pp. 1081–1084. doi:10.1126/science.1195691.

149. Llopis PM, Jackson AF, Sliusarenko O, Surovtsev I, Heinritz J, Emonet T, and Jacobs-Wagner C (2010). Spatial organization of the flow of genetic information in bacteria. *Nature*, **466**(7302), pp. 77–81. doi:10.1038/nature09152.

150. Mondal J, Bratton BP, Li Y, Yethiraj A, and Weisshaar JC (2011). Entropy-based mechanism of ribosome–nucleoid segregation in E. coli cells. *Biophys. J.*, **100**(11), pp. 2605–2613. doi:10.1016/j.bpj.2011.04.030.

151. Forchhammer J and Lindahl L (1971). Growth rate of polypeptide chains as a function of the cell growth rate in a mutant of Escherichia coli 15. *J. Mol. Biol.*, **55**(3), pp. 563–568. doi:10.1016/0022-2836(71)90337-8.

152. Scott M, Gunderson CW, Mateescu EM, Zhang Z, and Hwa T (2010). Interdependence of cell growth and gene expression: Origins and consequences. *Science*, **330**(6007), pp. 1099–1102. doi:10.1126/science.1192588.

153. Scott M and Hwa T (2011). Bacterial growth laws and their applications. *Curr. Op. Cell Biol.*, **22**(4), pp. 559–565. doi:10.1016/j.copbio.2011.04.014.

154. Lerman JA, Hyduke DR, Latif H, Portnoy VA, Lewis NE, Orth JD, Schrimpe-Rutledge AC, Smith RD, Adkins JN, Zengler K, and Palsson BO (2012). In silico method for modelling metabolism and gene product expression at genome scale. *Nat. Commun.*, **3**, p. 929. doi:10.1038/ncomms1928.

155. Liu JK, O'Brien EJ, Lerman JA, Zengler K, Palsson BO, and Feist AM (2014). Reconstruction and modeling protein translocation and compartmentalization in Escherichia coli at the genome-scale. *BMC Syst. Biol.*, **8**(1), p. 110. doi:10.1186/s12918-014-0110-6.

156. Earnest TM, Cole JA, Peterson JR, Hallock MJ, Kuhlman TE, and Luthey-Schulten Z (2016). Ribosome biogenesis in replicating cells: integration of experiment and theory. *Biopolymers*, **105**(10), pp. 735–751. doi:10.1002/bip.22892.

157. Valgepea K, Adamberg K, Seiman A, and Vilu R (2013). Escherichia coli achieves faster growth by increasing catalytic and translation rates of proteins. *Mol. Biosys.*, **9**(9), p. 2344. doi:10.1039/c3mb70119k.

158. Woese CR (1987). Bacterial evolution. *Microbiol. Rev.*, **51**(2), pp. 221–271. doi:10.1016/0022-2836(69)90095-3.

159. Woese CR, Kandler O, and Wheelis ML (1990). Towards a natural system of organisms: proposal for the domains archaea, bacteria, and eucarya. *Proc. Natl. Acad. Sci. USA*, **87**(12), pp. 4576–4579. doi:10.1073/pnas.87.12.4576.

160. Chen K, Eargle J, Sarkar K, Gruebele M, and Luthey-Schulten Z (2010). Functional role of ribosomal signatures. *Biophys. J.*, **99**(12), pp. 3930–3940. doi:10.1016/j.bpj.2010.09.062.

161. Hosokawa K, Fujimura RK, and Nomura M (1966). Reconstitution of functionally active ribosomes from inactive subparticles and proteins. *Proc. Natl. Acad. Sci. USA*, **55**(1), pp. 198–204. doi:10.1073/pnas.55.1.198.

162. Chai Q, Singh B, Peisker K, Metzendorf N, Ge X, Dasgupta S, and Sanyal S (2014). Organization of ribosomes and nucleoids in Escherichia coli cells during growth and in quiescence. *J. Biol. Chem.*, **289**(16), pp. 11342–11352. doi:10.1074/jbc.m114.557348.

163. Swain PS, Elowitz MB, and Siggia ED (2002). Intrinsic and extrinsic contributions to stochasticity in gene expression. *Proc. Natl. Acad. Sci. USA*, **99**(20), pp. 12795–12800. doi:10.1073/pnas.162041399.

164. Jones DL, Brewster RC, and Phillips R (2014). Promoter architecture dictates cell-to-cell variability in gene expression. *Science*, **346**(6216), pp. 1533–1536. doi:10.1126/science.1255301.

165. Taheri-Araghi S, Bradde S, Sauls JT, Hill NS, Levin PA, Paulsson J, Vergassola M, and Jun S (2015). Cell-size control and homeostasis in bacteria. *Curr. Biol.*, **25**(3), pp. 385–391. doi:10.1016/j.cub.2014.12.009.

166. Amir A (2014). Cell size regulation in bacteria. *Phys. Rev. Lett.*, **112**(20), p. 208102. doi:10.1103/physrevlett.112.208102.

167. Adiciptaningrum A, Osella M, Moolman MC, Lagomarsino MC, and Tans SJ (2015). Stochasticity and homeostasis in the E. coli replication and division cycle. *Sci. Rep.*, **5**, p. 18261. doi:10.1038/srep18261.

168. Wang P, Robert L, Pelletier J, Dang WL, Taddei F, Wright A, and Jun S (2010). Robust growth of Escherichia coli. *Curr. Biol.*, **20**(12), pp. 1099–1103. doi:10.1016/j.cub.2010.04.045.

169. Mir M, Wang Z, Shen Z, Bednarz M, Bashir R, Golding I, Prasanth SG, and Popescu G (2011). Optical measurement of cycle-dependent cell growth. *Proc. Natl. Acad. Sci. USA*, **108**(32), pp. 13124–13129. doi:10.1073/pnas.1100506108.

170. Cooper S (2006). Distinguishing between linear and exponential cell growth during the division cycle: Single-cell studies, cell-culture studies, and the object of cell-cycle research. *Theor. Biol. Med. Model*, **3**(1), pp. 1–15. doi:10.1186/1742-4682-3-10.

171. Godin M, Delgado FF, Son S, Grover WH, Bryan AK, Tzur A, Jorgensen P, Payer K, Grossman AD, Kirschner MW, and Manalis SR (2010). Using buoyant mass to measure the growth of single cells. *Nat. Methods*, **7**(5), pp. 387–390. doi:10.1038/nmeth.1452.

172. Campos M, Surovtsev IV, Kato S, Paintdakhi A, Beltran B, Ebmeier SE, and Jacobs-Wagner C (2014). A constant size extension drives bacterial cell size homeostasis. *Cell*, **159**(6), pp. 1433–1446. doi:10.1016/j.cell.2014.11.022.

173. Koch AL (1966). Distribution of cell size in growing cultures of bacteria and the applicability of the Collins-Richmond principle. *J. Gen. Microbiol.*, **45**(3), pp. 409–417. doi:10.1099/00221287-45-3-409.

174. Koch AL and Schaechter M (1962). A model for statistics of the cell division process. *J. Gen. Microbiol.*, **29**(3), pp. 435–454. doi:10.1099/00221287-29-3-435.

175. Storn R and Price K (1997). Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces. *J. Global Optim.*, **11**(4), pp. 341–359. doi:10.1023/A:1008202821328.

176. Skarstad K, Steen HB, and Boye E (1983). Cell cycle parameters of slowly growing Escherichia coli B/r studied by flow cytometry. *J. Bacteriol.*, **154**(2), pp. 656–662.

177. Michelsen O (2003). Precise determinations of C and D periods by flow cytometry in Escherichia coli K-12 and B/r. *Microbiol.*, **149**(4), pp. 1001–1010. doi:10.1099/mic.0.26058-0.

178. Powell EO (1956). Growth rate and generation time of bacteria, with special reference to continuous culture. *J. Gen. Microbiol.*, **15**(3), pp. 492–511. doi:10.1099/00221287-15-3-492.

179. Osella M, Nugent E, and Lagomarsino MC (2014). Concerted control of Escherichia coli cell division. *Proc. Natl. Acad. Sci. USA*, **111**(9), pp. 3431–3435. doi:10.1073/pnas.1313715111.

180. Soifer I, Robert L, and Amir A (2016). Single-cell analysis of growth in budding yeast and bacteria reveals a common size regulation strategy. *Curr. Biol.*, **26**(3), pp. 356–361. doi:10.1016/j.cub.2015.11.067.

181. Bates D and Kleckner N (2005). Chromosome and replisome dynamics in E. coli: Loss of sister cohesion triggers global chromosome movement and mediates chromosome segregation. *Cell*, **121**(6), pp. 899–911. doi:10.1016/j.cell.2005.04.013.

182. Bates D, Epstein J, Boye E, Fahrner K, Berg H, and Kleckner N (2005). The Escherichia coli baby cell column: a novel cell synchronization method provides new insight into the bacterial cell cycle. *Mol. Microbiol.*, **57**(2), pp. 380–391. doi:10.1111/j.1365-2958.2005.04693.x.

183. Richmond CS, Glasner JD, Mau R, Jin H, and Blattner FR (1999). Genome-wide expression profiling in Escherichia coli K-12. *Nucl. Acids Res.*, **27**(19), pp. 3821–3835. doi:10.3724/sp.j.1035.2010.00591.

184. Sundararaj S (2004). The CyberCell database (CCDB): a comprehensive, self-updating, relational database to coordinate and facilitate in silico modeling of Escherichia coli. *Nucl. Acids Res.*, **32**(90001), pp. 293D–295. doi:10.1093/nar/gkh108.

185. Li GW, Burkhardt D, Gross C, and Weissman JS (2014). Quantifying absolute protein synthesis rates reveals principles underlying allocation of cellular resources. *Cell*, **157**(3), pp. 624–635. doi:10.1016/j.cell.2014.02.033.

186. Kuhlman TE and Cox EC (2010). Site-specific chromosomal integration of large synthetic constructs. *Nucl. Acids Res.*, **38**(6), pp. e92–e92. doi:10.1093/nar/gkp1193.

187. Kuhlman TE and Cox EC (2012). Gene location and DNA density determine transcription factor distributions in Escherichia coli. *Mol. Sys. Biol.*, **8**. doi:10.1038/msb.2012.42.

188. Kuhlman TE and Cox EC (2013). DNA-binding-protein inhomogeneity in E. coli modeled as biphasic facilitated diffusion. *Phys. Rev. E*, **88**(2). doi:10.1103/physreve.88.022701.

189. Datsenko KA and Wanner BL (2000). One-step inactivation of chromosomal genes in Escherichia coli K-12 using PCR products. *Proc. Natl. Acad. Sci. USA*, **97**(12), pp. 6640–6645. doi:10.1073/pnas.120163297.

190. Kuhlman TE and Cox EC (2010). A place for everything. *Bioeng. Bugs*, **1**(4), pp. 298–301. doi:10.4161/bbug.1.4.12386.

191. Tas H, Nguyen CT, Patel R, Kim NH, and Kuhlman TE (2015). An integrated system for precise genome modification in Escherichia coli. *PLoS ONE*, **10**(9), p. e0136963. doi:10.1371/journal.pone.0136963.

192. Joshi MC, Bourniquel A, Fisher J, Ho BT, Magnan D, Kleckner N, and Bates D (2011). Escherichia coli sister chromosome separation includes an abrupt global transition with concomitant release of late-splitting intersister snaps. *Proc. Natl. Acad. Sci. USA*, **108**(7), pp. 2765–2770. doi:10.1073/pnas.1019593108.

193. Pérez F and Granger BE (2007). IPython: a system for interactive scientific computing. *Comput. Sci. Eng.*, **9**(3), pp. 21–29. doi:10.1109/MCSE.2007.53.

194. Jones E, Oliphant T, Peterson P, et al. (2001–). SciPy: Open source scientific tools for Python. `http://www.scipy.org/`. [Online; accessed 2016-03-18].

195. van der Walt S, Schönberger JL, Nunez-Iglesias J, Boulogne F, Warner JD, Yager N, Gouillart E, and Yu T (2014). scikit-image: image processing in python. *PeerJ*, **2**, p. e453. doi:10.7717/peerj.453.

196. Behnel S, Bradshaw R, Citro C, Dalcin L, Seljebotn DS, and Smith K (2011). Cython: The best of both worlds. *Comput. Sci. Eng.*, **13**(2), pp. 31–39. doi:10.1109/MCSE.2010.118.

197. Choudhary K, Oehler S, and Narang A (2014). Protein distributions from a stochastic model of the lac operon of E. coli with DNA looping: Analytical solution and comparison with experiments. *PLoS ONE*, **9**(7), p. e102580. doi:10.1371/journal.pone.0102580.

198. Portier C, Dondon L, and Grunberg-Manago M (1990). Translational autocontrol of the escherichia coli ribosomal protein s15. *J. Mol. Biol.*, **211**(2), pp. 407–414. doi:10.1016/0022-2836(90)90361-o.

199. Gralla JD (2004). Escherichia coli ribosomal RNA transcription: regulatory roles for ppGpp, NTPs, architectural proteins and a polymerase-binding protein. *Mol. Microbiol.*, **55**(4), pp. 973–977. doi:10.1111/j.1365-2958.2004.04455.x.

200. Bradley MD, Beach MB, de Koning APJ, Pratt TS, and Osuna R (2007). Effects of fis on Escherichia coli gene expression during different growth stages. *Microbiol.*, **153**(9), pp. 2922–2940. doi:10.1099/mic.0.2007/008565-0.

201. Taylor PR, Baker RE, Simpson MJ, and Yates CA (2016). Coupling volume-excluding compartment-based models of diffusion at different scales: Voronoi and pseudo-compartment approaches. *J. R. Soc. Interface*, **13**(120), p. 20160336. doi:10.1098/rsif.2016.0336.

202. Bruna M and Chapman SJ (2015). Diffusion in spatially varying porous media. *SIAM J. Appl. Math.*, **75**(4), pp. 1648–1674. doi:10.1137/141001834.

203. Bruna M and Chapman SJ (2012). Diffusion of multiple species with excluded-volume effects. *J. Chem. Phys.*, **137**(20), p. 204116. doi:10.1063/1.4767058.

204. Chirico G and Langowski J (1994). Kinetics of DNA supercoiling studied by Brownian dynamics simulation. *Biopolymers*, **34**(3), pp. 415–433. doi:10.1002/bip.360340313.

205. Klenin K, Merlitz H, and Langowski J (1998). A Brownian dynamics program for the simulation of linear and circular DNA and other wormlike chain polyelectrolytes. *Biophys. J.*, **74**(2), pp. 780–788. doi:10.1016/s0006-3495(98)74003-2.

206. Cagliero C, Grand RS, Jones MB, Jin DJ, and O'Sullivan JM (2013). Genome conformation capture reveals that the Escherichia coli chromosome is organized by replication and transcription. *Nucl. Acids Res.*, **41**(12), pp. 6058–6071. doi:10.1093/nar/gkt325.

207. Balaeff A, Eargle J, and Roberts E (2005). Ionize v 1.6. `http://www.scs.illinois.edu/schulten/software/mdtools/ionize/`.

208. Jorgensen WL, Chandrasekhar J, Madura JD, Impey RW, and Klein ML (1983). Comparison of simple potential functions for simulating liquid water. *J. Chem. Phys.*, **79**, pp. 926–935. doi:10.1063/1.445869.

209. Grubmueller H and Groll V (1996). Solvate v 1.0. `http://www.mpibpc.mpg.de/grubmueller/solvate`.

210. Humphrey W, Dalke A, and Schulten K (1996). VMD–Visual Molecular Dynamics. *J. Mol. Graphics*, **14**, pp. 33–38. doi:10.1016/0263-7855(96)00018-5.

211. Auffinger P and Westhof E (1997). RNA hydration: Three nanoseconds of multiple molecular dynamics simulations of the solvated tRNA(Asp) anticodon hairpin. *J. Mol. Biol.*, **269**(3), pp. 326–341. doi:10.1006/jmbi.1997.1022.

212. Caliskan G, Hyeon C, Perez-Salas U, Briber RM, Woodson SA, and Thirumalai D (2005). Persistence length changes dramatically as RNA folds. *Phys. Rev. Lett.*, **95**, p. 268303. doi:10.1103/PhysRevLett.95.268303.

213. Eargle J and Luthey-Schulten Z, Simulating dynamics in RNA and protein complexes. In N Leontis and E Westhof, eds., *RNA 3D Structure Analysis and Prediction*, vol. 27 of *Nucleic Acids and Molecular Biology*, pp. 213–238 (Springer Berlin Heidelberg, 2012). ISBN 9783-642-2573-9-1. doi:10.1007/978-3-642-25740-7_11.

214. Cannone J, Subramanian S, Schnare M, Collett J, D'Souza L, Du Y, Feng B, Lin N, Madabusi L, Muller K, Pande N, Shang Z, Yu N, and Gutell R (2002). The comparative RNA web (CRW) site: an online database of comparative sequence and structure information for ribosomal, intron, and other RNAs. *BMC Bioinformatics*, **3**(1), p. 15. doi:10.1186/1471-2105-3-2.

215. Skinner SO, Xu H, Nagarkar-Jaiswal S, Freire PR, Zwaka TP, and Golding I (2016). Single-cell analysis of transcription kinetics across the cell cycle. *eLife*, **5**. doi:10.7554/elife.12175.

216. Llopis PM, Sliusarenko O, Heinritz J, and Jacobs-Wagner C (2012). In vivo biochemistry in bacterial cells using FRAP: Insight into the translation cycle. *Biophys. J.*, **103**(9), pp. 1848–1859. doi:10.1016/j.bpj.2012.09.035.

Appendix A

Supporting information for Chapter 2 *

A.1 Obtaining the switching rate functions $k_{\mathrm{fn}}([\mathrm{I}])$ and $k_{\mathrm{nf}}([\mathrm{I}])$

In the present work we relied on a microscopic model previously developed to describe the *lac* system in the absence of DNA looping[33]. In that work the authors developed a detailed model of the microscopic interactions between inducer, repressor, and operator that could reproduce the intrinsic noise of the *lac* system. They also noted that their microscopic model was well-approximated by a two-state model with *Off*→*On* and *On*→*Off* rate functions that appeared Hill-like when plotted against the inducer concentration. To obtain expressions for $k_{\mathrm{fn}}([\mathrm{I}])$ and $k_{\mathrm{nf}}([\mathrm{I}])$ to use in our study we fit the data describing the *Off*→*On* and *On*→*Off* transitions from Roberts et al. [33] to the Hill-like functions:

$$k_{\mathrm{fn}}([\mathrm{I}]) = k^0_{\mathrm{fn}} + (k^1_{\mathrm{fn}} - k^0_{\mathrm{fn}})\frac{[\mathrm{I}]^{H_{\mathrm{fn}}}}{{I_{\mathrm{fn}}}^{H_{\mathrm{fn}}} + [\mathrm{I}]^{H_{\mathrm{fn}}}} \tag{A.1a}$$

$$k_{\mathrm{nf}}([\mathrm{I}]) = k^0_{\mathrm{nf}} + (k^1_{\mathrm{nf}} - k^0_{\mathrm{nf}})\frac{[\mathrm{I}]^{H_{\mathrm{nf}}}}{{I_{\mathrm{nf}}}^{H_{\mathrm{nf}}} + [\mathrm{I}]^{H_{\mathrm{nf}}}}. \tag{A.1b}$$

The k^0_x and k^1_x parameters in these equations describe the rates in the absence of inducer and at saturating inducer, respectively. These values can be obtained directly from the microscopic

[*] Appendix A contains material reproduced from Earnest TM, Roberts E, Assaf M, Dahmen K, and Luthey-Schulten Z (2013). DNA looping increases the range of bistability in a stochastic model of the lac genetic switch. *Phys. Biol.*, **10**(2), p. 026002. doi:10.1088/1478-3975/10/2/026002. © IOP Publishing. Reproduced with permission. All rights reserved.

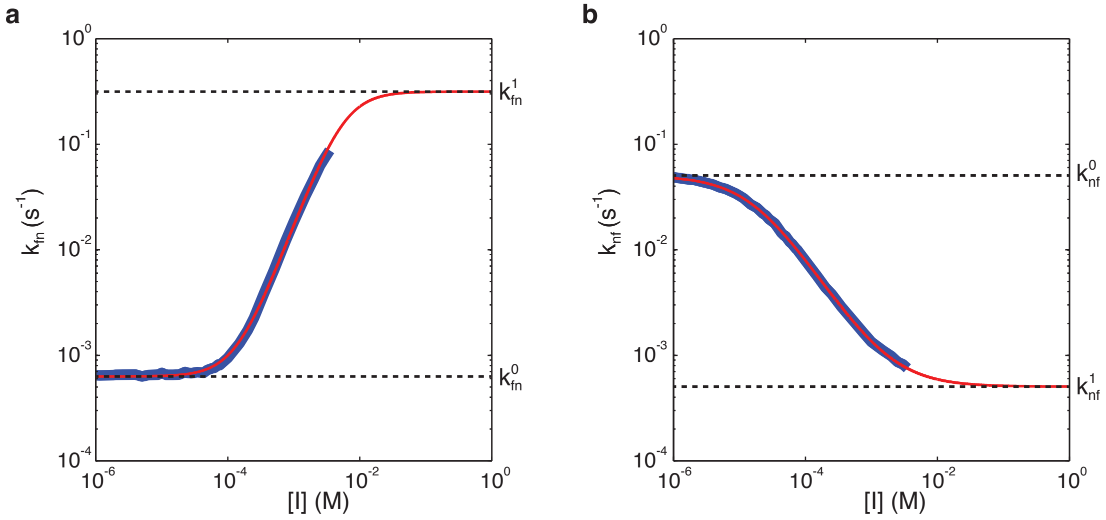


Figure A.1 (a) Best fit of Eq. A.1a (red) to the *Off*→*On* transition rates shown in Figure 15E of Roberts et al.[33] (blue). (b) The same as (a) for Eq. A.1b and the *On*→*Off* rates from Figure 15F.

model of Roberts et al.[33] as:

$$k^0_{\text{fn}} = k_{\text{roff}} = 6.30\times10^{-4}\,s^{-1}, \tag{A.2a}$$

$$k^1_{\text{fn}} = k_{\text{i2roff}} = 3.15\times10^{-1}\,s^{-1}, \tag{A.2b}$$

$$k^0_{\text{nf}} = \frac{N_{\text{LacI}}\cdot k_{\text{ron}}}{N_A\cdot V} = 5.04\times10^{-2}\,s^{-1}, \tag{A.2c}$$

$$k^1_{\text{nf}} = \frac{N_{\text{LacI}}\cdot k_{\text{i2ron}}}{N_A\cdot V} = 5.04\times10^{-4}\,s^{-1}, \tag{A.2d}$$

where N_A is the Avogadro constant, N_{LacI} is the number of *lac* repressor dimers ($N_{\text{LacI}} = 10$), and V is the volume of the cell (8×10^{-16} L). The rate constants here are the binding and unbinding rates of repressor to operator for repressor with no bound inducer (r) and with both inducer sites bound (i2). To obtain values for the remaining H and I parameters for each equation we fit the data plotted as the blue lines in Figure 15ef in Roberts et al.[33] to Eq. A.1a + Eq. A.1b, respectively, using a nonlinear least squares method and the Matlab curve fitting toolbox. The fits are shown in Figure A.1 and give the following values for the free parameters:

$$I_{\text{fn}} = 5.68\times10^{-3}\,M, \tag{A.2e}$$

$$H_{\text{fn}} = 1.67 \tag{A.2f}$$

$$I_{\text{nf}} = 1.74\times10^{-5}\,M, \tag{A.2g}$$

$$H_{\text{nf}} = 1.00 \tag{A.2h}$$

A.2 Comparison of approximation methods

To compare the major computational methods and approximations in this work, we have plotted the probability distributions obtained from stochastic simulation algorithm (SSA) simulations and finite state projection (FSP) solution ofEqs. 2.16a–2.16c. We compare to these two approximate methods, the geometric bursting approximation and the mean messenger RNA (mRNA) approximation. The results for the two-state model are presented in Figure A.2a. The SSA and FSP distributions, aside from sampling noise in the SSA probability distribution function (PDF), are indistinguishable. This suggests that the error from the use of the finite state projection is minimal, or at least dominated by other sources.

The geometric burst approximation and the mean mRNA approximation both computed using the FSP, are plotted in Figure A.2a in dash-dot green and dash-dot-dot magenta, respectively. To see why eliminating the mRNA dependence using its mean is a poor approximation, we have computed PDF for both the two- and three-state models using mean mRNA. This approximation is applied by simply replacing the β-galactoside permease (LacY) translation propensity $k_{\mathrm{tl}} m$ with

$$k_{\mathrm{tl}}\langle m\rangle = k_{\mathrm{degp}}\mathscr{N}\mathscr{F}(n), \tag{A.3}$$

and ignoring the mRNA terms. For the two-state model, the differences are subtle. The tails of the distribution are shorter in the mean mRNA calculation for both populations (see inset). However with the three-state model (Figure A.2b), the mean mRNA approximation fails completely. The noise in the LO and HI states are not well represented using this method. It is clear that simply eliminating the mRNA dependence with its average is unacceptably misrepresents the noise in the two induction states. This will lead to inaccurate calculations of switching times. The geometric burst approximation does not have this problem.

A.3 Parameter sensitivities

To see how the various parameters affect the bistability range and the characteristic lifetime $\tau_{50\%}$, we performed scans of the parameters over the likely range starting with the mean parameter set (Figure A.3). The jagged nature of the curves is due to the fact that as the parameters change, the location of n_0 changes. Since we measure the switching time through a sink placed at n_0, the discontinuous change of the location of the unstable fixed point causes a discontinuous change in the computed values. The parameters which appear to affect the bistability range and lifetime the most over the biologically reasonable region of parameter space, are k_{fl} (Figure A.3i) and I_{lf} (Figure A.3f) respectively. The parameter k_{lf}^0 (Figure A.3ab) seems to be acceptable over the broadest range compared to the other parameters. The Hill coefficient H_{lf} (Figure A.3gh) also appears to be insensitive over its range. This is interesting since k_{lf}^0 seems to depend on k_{fl} and I_{lf} on H_{lf}. The values of k_{l}^0 and I_{lf} for the most part do not matter so long as k_{fl} and I_{lf} are chosen to satisfy their dependence.

We tested the sensitivity of the model to changing the leakage factor ϵ (Figure A.4). Over the likely range of ϵ from 5.7×10^{-4} to 2.1×10^{-3}, the range of bistability only changed from 23.0 μM to 22.75 μM. The switching lifetime $\tau_{50\%}$ only changed from 54.7 to 53.8 cell cycles.

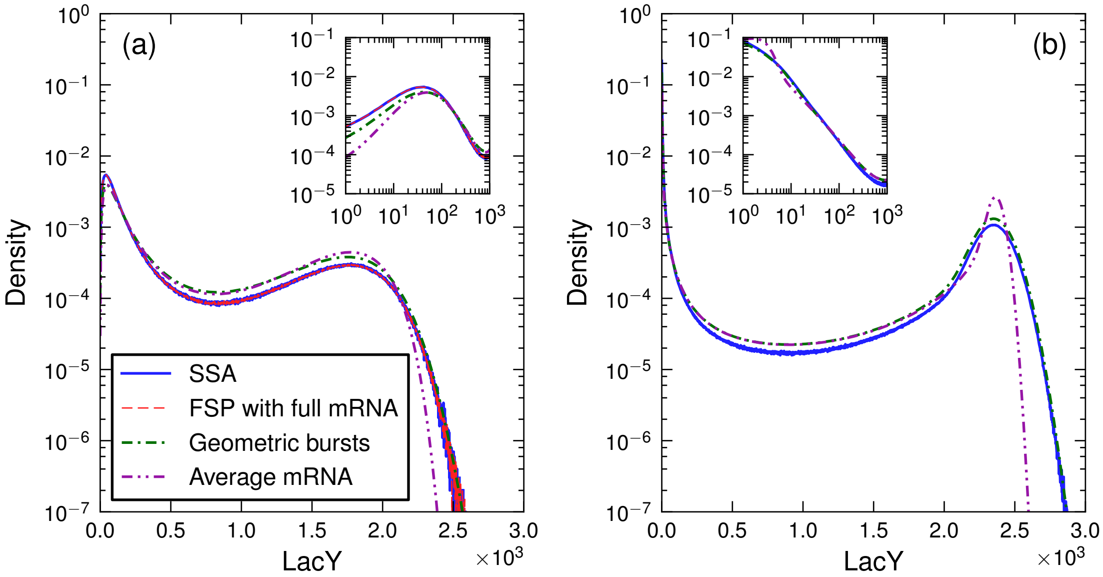


Figure A.2 Comparison of the different methods for computing the LacY abundance probability distributions. (a) Two-state model. The stochastic simulation of Eqs. 2.16a–2.16c and the full FSP calculation (explicit mRNA dependence) are shown in solid blue and dashed red, respectively. The agreement between the two curves is high enough that they completely overlap in the figure. The approximate solutions using the geometric burst approximation and mean mRNA are shown in dash-dot green and dash-dot-dot magenta respectively. (b) Three-state model. The mean mRNA approximation fails to estimate the width of the HI state.

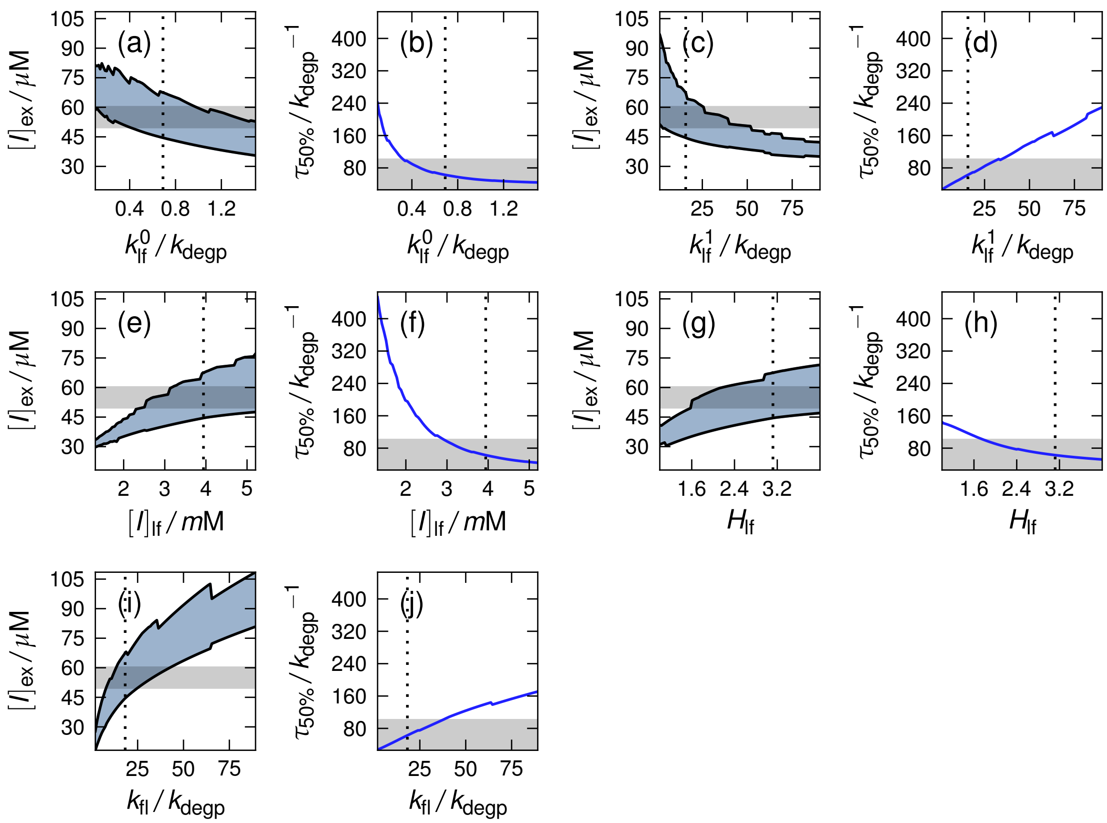


Figure A.3 Dependence of bistability range and $\tau_{50\%}$ on parameters. The dotted line indicates the values for the mean parameter set. The shaded gray region denotes the range of acceptance for the parameter searches.

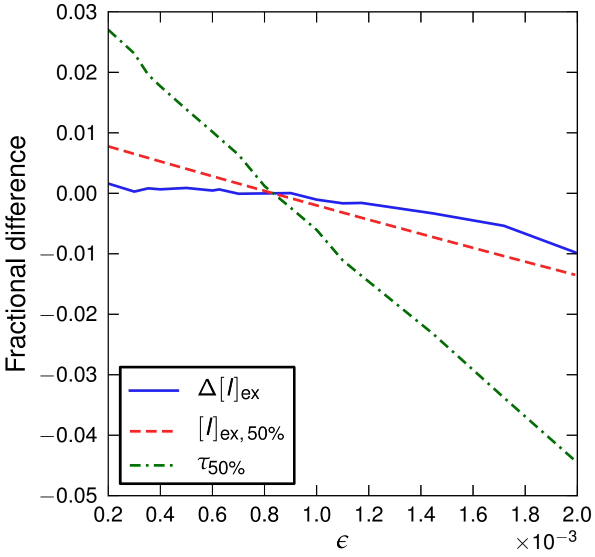


Figure A.4 Dependence on bistability range, equal induction probability concentration, and lifetime on leakage factor ϵ. Over the uncertainty range, the computed values vary no more than 4%

A.4 Calculating transcription state probabilities from repression values

We use the repression values, defined by the ratio of the maximum β-galactosidase (LacZ) activity to the repressed LacZ activity, to compute free energies for each possible repressor/operon state (Figure A.5). From Eq. 1 in Vilar and Leibler[91], the free energy of binding for an operator is

$$\Delta G_{\mathrm{bn},O} = -\ln \frac{R_O - 1}{N} \tag{A.4}$$

where R_O is the repression value for a mutant with only operator O, placed in the O_1 location in the operon. The free energy of looping is computed from Eq. 3 in Vilar and Leibler[91],

$$\Delta G_{\mathrm{lp},O_m O_{a,n}} = -\ln \frac{(R_{O_m O_{a,n}} - 1)(1 + N\mathrm{e}^{-\Delta G_{\mathrm{bn},Oa}}) - N\mathrm{e}^{-\Delta G_{\mathrm{bn},Om}} - N(N-1)\mathrm{e}^{-\Delta G_{\mathrm{bn},Om} - \Delta G_{\mathrm{bn},Oa}}}{N\mathrm{e}^{-\Delta G_{\mathrm{bn},Om} - \Delta G_{\mathrm{bn},Oa}}}, \tag{A.5}$$

Where $R_{O_m O_{a,n}}$ is the repression level for a loop with O_m in the O_1 position and O_a in the position of operator O_n. In both equations, N represents the total number of *lac* repressor (LacI) dimers present.

From Oehler et al.[46], these repression values are reported:

$$R_{O_1} = 200 \tag{A.6a}$$
$$R_{O_2} = 21 \tag{A.6b}$$
$$R_{O_3} = 1.3 \tag{A.6c}$$
$$R_{O_1 O_{2,2}} = 2300 \tag{A.6d}$$
$$R_{O_1 O_{3,3}} = 6100 \tag{A.6e}$$

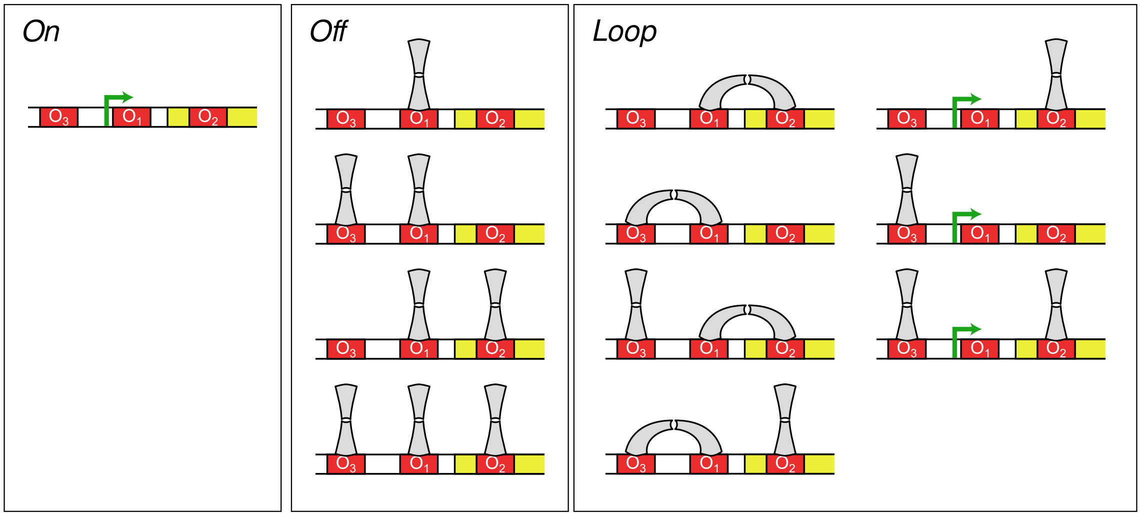


Figure A.5 Enumeration of the LacI/operator states included in each coarse-grained state.

for LacI count $N = 50$. The parameters used in our simulations are for $N = 10$. There appears to be a dependence of these free energies to the number of LacI, however the dependence seems small enough to neglect.

To compute probabilities, we use the free energies to calculate the Boltzmann factor for each state. We will use the notation $P_{\mathrm{s}n}$ to denote a single repressor bound to operator n, $P_{\mathrm{s}nm}$ for two repressors bound at n and m and P_{123} for three bound repressors. Looped states are represented by $P_{\mathrm{l}n}$ for the state with O_1-O_n looped, and $P_{\mathrm{sl}n}$ for the state with O_1-O_n looped and another repressor bound on the remaining operator. The probabilities are

$$P_{\mathrm{s}n} = Z^{-1} N \mathrm{e}^{-\Delta G_{\mathrm{bn},n}} \tag{A.7a}$$

$$P_{\mathrm{s}nm} = Z^{-1} N(N-1)\, \mathrm{e}^{-\Delta G_{\mathrm{bn},n} - \Delta G_{\mathrm{bn},m}} \tag{A.7b}$$

$$P_{\mathrm{s}123} = Z^{-1} N(N-1)(N-2)\, \mathrm{e}^{-\Delta G_{\mathrm{bn},1} - \Delta G_{\mathrm{bn},2} - \Delta G_{\mathrm{bn},3}} \tag{A.7c}$$

$$P_{\mathrm{l}n} = Z^{-1} N \mathrm{e}^{-\Delta G_{\mathrm{bn},n} - \Delta G_{\mathrm{bn},1} - \Delta G_{O_1 O_n,n}} \tag{A.7d}$$

$$P_{\mathrm{sl}n} = Z^{-1} N(N-1)\, \mathrm{e}^{-\Delta G_{\mathrm{bn},1} - \Delta G_{\mathrm{bn},2} - \Delta G_{\mathrm{bn},3} - \Delta G_{O_1 O_n,n}} \tag{A.7e}$$

where Z is the partition function. Finally we can write the probabilities for each of our states

$$P_{On} = Z^{-1} \tag{A.8a}$$

$$P_{Off} = P_{\mathrm{s}1} + P_{\mathrm{s}12} + P_{\mathrm{s}13} + P_{\mathrm{s}123} \tag{A.8b}$$

$$P_{Loop} = P_{\mathrm{l}2} + P_{\mathrm{l}3} + P_{\mathrm{sl}2} + P_{\mathrm{sl}3} + P_{\mathrm{s}2} + P_{\mathrm{s}3}, \tag{A.8c}$$

and

$$\epsilon = \frac{P_{\mathrm{s}2} + P_{\mathrm{s}3}}{P_{Loop}}. \tag{A.8d}$$

In Oehler et al.[46], the repression values were measured for 50 and 900 intracellular LacI. If we assume that the free energies depend on the repressor count linearly, we can extrapolate to the free energies at $N = 10$. This yields probabilities of

$$P_{On}^{\text{ex}} = 6.0 \times 10^{-5} \tag{A.9a}$$

$$P_{Off}^{\text{ex}} = 1.1 \times 10^{-2} \tag{A.9b}$$

$$\epsilon^{\text{ex}} = 2.6 \times 10^{-4} \tag{A.9c}$$

which compare better with our FSP results.

Appendix B

Supporting information for Chapter 3 [*]

B.1 Derivation of user-supplied functions for ODE solver

B.1.1 Definitions

We are solving the system

$$\dot{\boldsymbol{y}} = \boldsymbol{f}(\boldsymbol{y}, \{\mathscr{R}\}) \tag{B.1a}$$

where the concentrations are denoted by

$$\boldsymbol{y} = \begin{bmatrix} y_0 & y_1 & y_2 & \dots \end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^N \tag{B.1b}$$

and the rate constants as

$$\boldsymbol{k} = \begin{bmatrix} k_0 & k_1 & k_2 & \dots \end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^M \tag{B.1c}$$

The chemical reactions are from the set

$$(r_1, r_2, p, \kappa) \in \mathscr{R} \tag{B.2}$$

where r_1 is the binding protein index, r_2 is the intermediate index, p is the product index, and κ is the rate constant index.

To compute the derivative of functional of the solution to a system of ODEs, such as the root mean square (RMS) error objective function used to fit the small subunit (SSU) assembly model to experimental data, we use adjoint sensitivity analysis. This method requires the standard numerical solution to the ODE system, as well as the solution to an auxiliary system which is

[*] Appendix B contains material reproduced with the permission of Elsevier from Earnest TM, Lai J, Chen K, Hallock MJ, Williamson JR, and Luthey-Schulten Z (2015). Toward a whole-cell model of ribosome biogenesis: Kinetic modeling of SSU assembly. *Biophys. J.*, **109**(6), pp. 1117–1135. doi:10.1016/j.bpj.2015.07.030.

integrated backwards in time.

B.1.2 Derivation of functions describing forward problem

The right hand side function, $\boldsymbol{f}(\boldsymbol{y}, \{\mathscr{R}\})$ is defined from the chemical reactions and concentrations as

$$f_i = \sum_{(r_1,r_2,p,\kappa)\in\mathscr{R}} k_\kappa y_{r_0} y_{r_2} (\delta_{p\,i} - \delta_{r_1\,i} - \delta_{r_2\,i}). \tag{B.3}$$

Taking the derivative with respect to $\boldsymbol{y}$ we get the Jacobian

$$\partial_j f_i = \sum_{(r_1,r_2,p,\kappa)\in\mathscr{R}} k_\kappa (\delta_{r_1\,j} y_{r_2} + \delta_{r_2\,j} y_{r_1})(\delta_{p\,i} - \delta_{r_1\,i} - \delta_{r_2\,i}) \tag{B.4}$$

The Jacobian-vector product function is then simply

$$\partial_j f_i v_j = \sum_{(r_1,r_2,p,\kappa)\in\mathscr{R}} k_\kappa (v_{r_1} y_{r_2} + v_{r_2} y_{r_1})(\delta_{p\,i} - \delta_{r_1\,i} - \delta_{r_2\,i}). \tag{B.5}$$

The right hand side function Eq. B.3 and the Jacobian-vector product function Eq. B.5 are sufficient to define the forward system for the ODE solver.

B.1.3 Derivation of functions describing adjoint problem

To optimize the system to the experimental data, we construct the objective functional

$$\Phi[\boldsymbol{y}] = \int_{T_0}^{T_1} \mathrm{d}t\, \phi(\boldsymbol{y}) \tag{B.6a}$$

with

$$\phi(\boldsymbol{y}) = \frac{1}{\mathscr{N}_{\text{expt.}} \mathscr{N}_{\text{prot.}} (T_1 - T_0)\, t} \sum_{\substack{e\in\{\text{expts.}\} \\ s\in\{\text{r-prot.}\}}} \left[\chi(y_{e,s}(t)) - \chi_{e,s}^{\text{expt}}(t)\right]^2 \tag{B.6b}$$

where $y_{e,s}(t)$ is the concentration of protein s starting with an initial prebinding intermediate from experiment e, $\chi(y)$ is the experiment function derived in Eq. B.28, and $\chi_{e,s}^{\text{expt}}(t)$ is a single exponential fit to the pulse/chase quantitative mass spectrometry (P/C qMS) experiment. The t in the denominator weights the mean-squared error to treat each time decade equally.

In order to compute the sensitivities of $\Phi[\boldsymbol{y}]$ we will use adjoint sensitivity analysis, We are going to compute $\mathrm{d}_\mu \Phi$, where we are taking the convention that Greek indices are rate constant variables and roman indices are concentration variables. The idea is we compute the derivatives of $\Phi[\boldsymbol{y}]$ with a Lagrange multiplier, $\boldsymbol{\lambda}$, that forces $\boldsymbol{y}(t)$ to solve Eq. B.1a. The augmented functional is

$$\tilde{\Phi} = \Phi - \int_{T_0}^{T_1} \mathrm{d}t\, \boldsymbol{\lambda}^{\mathrm{T}} (\dot{\boldsymbol{y}} - \boldsymbol{f}) \tag{B.7}$$

and its sensitivities are

$$\mathrm{d}_\mu \Phi = \mathrm{d}_\mu \tilde{\Phi} = \int_{T_0}^{T_1} \mathrm{d}t \left[\mathrm{d}_\mu \phi - \lambda_i (\mathrm{d}_\mu \dot{y}_i - \mathrm{d}_\mu f_i) \right] \tag{B.8a}$$

$$= \int_{T_0}^{T_1} \mathrm{d}t \left[\partial_\mu \phi + \partial_i \phi \partial_\mu y_i - \lambda_i (\partial_\mu \dot{y}_i - \partial_\mu f_i - \partial_j f_i \partial_\mu y_j) \right]. \tag{B.8b}$$

Now we use integration by parts on the time derivative term,

$$\mathrm{d}_\mu \Phi = -\lambda_i \partial_\mu y_i \Big|_0^T + \int_{T_0}^{T_1} \mathrm{d}t \left[\partial_\mu \phi + \partial_i \phi \partial_\mu y_i + \dot{\lambda}_i \partial_\mu y_i + \lambda_i \partial_\mu f_i + \lambda_i \partial_j f_i \partial_\mu y_j \right], \tag{B.9}$$

and require that the Lagrange multiplier solve the final value problem

$$\begin{cases} \dot{\lambda}_i = -\partial_i f_j \lambda_j - \partial_i \phi \\ \lambda_i(T) = 0 \end{cases} \tag{B.10}$$

Substituting these definitions in, we have

$$\begin{aligned} \mathrm{d}_\mu \Phi =& \lambda_i(0) \partial_\mu y_i(0) + \int_{T_0}^{T_1} \mathrm{d}t \, [\partial_\mu \phi + \partial_i \phi \partial_\mu y_i - \partial_i f_j \partial_\mu y_i \lambda_j - \partial_i \phi \partial_\mu y_i \\ & + \lambda_i \partial_\mu f_i + \lambda_i \partial_j f_i \partial_\mu y_j] \end{aligned} \tag{B.11a}$$

$$= \lambda_i(0) \partial_\mu y_i(0) + \int_{T_0}^{T_1} \mathrm{d}t \left[\partial_\mu \phi + \lambda_i \partial_\mu f_i \right] \tag{B.11b}$$

So to compute the sensitivities of $\Phi[\boldsymbol{y}]$, we integrate Eq. B.10 backward in time using our previous solution to Eq. B.1a and substitute into the integral Eq. B.11b. To avoid saving intermediate λ_i values, we integrate Eq. B.11b backwards as well

$$\mathrm{d}_\mu \Phi = \lambda_i(0) \partial_\mu y_i(0) - \int_T^0 \mathrm{d}t \left[\partial_\mu \phi + \lambda_i \partial_\mu f_i \right] \tag{B.12}$$

For the chemical reaction system, in the r.h.s. of Eq. B.10 the first term is just the product $\mathbf{J}^{\mathrm{T}} \boldsymbol{\lambda}$

$$\partial_j f_i \lambda_i = \sum_{(r_1, r_2, p, \kappa) \in \mathscr{R}} k_\kappa (\delta_{r_1 j} y_{r_2} + \delta_{r_2 j} y_{r_1})(\lambda_p - \lambda_{r_1} - \lambda_{r_2}). \tag{B.13}$$

The second term is

$$\partial_i \phi = \sum_{\substack{e \in \{\text{expts.}\} \\ s \in \{\text{r-prot.}\}}} 2 \delta_{i\,s} \left(\chi_s(y_s) - \chi_s^{\text{expt}}(t) \right) \partial_s \chi_s(y_s), \tag{B.14}$$

where

$$\partial_s \chi_s(y_s) = -\frac{p_{0\,s}^*(p_{0\,s}^* + p_{0\,s} - r_0)}{r_0 \cdot (y_s + p_{0\,s}^*)^2} \tag{B.15}$$

The Jacobian with respect to the Lagrange variables is

$$\partial_{j'} \dot{\lambda}_i = -\sum_{(r_1, r_2, p, \kappa) \in \mathscr{R}} k_\kappa (\delta_{r_1 i} y_{r_2} + \delta_{r_2 i} y_{r_1})(\delta_{j'\,p} - \delta_{j'\,r_1} - \delta_{j'\,r_2}). \tag{B.16}$$

and the Jacobian vector product is

$$v_{j'}\partial_{j'}\dot{\lambda}_i = -\sum_{(r_1,r_2,p,\kappa)\in\mathscr{R}} k_\kappa(\delta_{r_1 i}y_{r_2} + \delta_{r_2 i}y_{r_1})(v_p - v_{r_1} - v_{r_2}). \tag{B.17}$$

There is no explicit parameter dependence in our objective function and the initial conditions do not depend on our parameters either, so we need only worry about the second integral term in Eq. B.11b:

$$\partial_\mu f_i = \sum_{(r_1,r_2,p,\kappa)\in\mathscr{R}} \delta_{\mu\kappa} y_{r_1} y_{r_2}(\delta_{p\,i} - \delta_{r_1\,i} - \delta_{r_2\,i}) \tag{B.18}$$

$$\lambda_i\partial_\mu f_i = \sum_{(r_1,r_2,p,\kappa)\in\mathscr{R}} \delta_{\mu\kappa} y_{r_1} y_{r_2}(\lambda_p - \lambda_{r_1} - \lambda_{r_2}). \tag{B.19}$$

B.2 Pulse/chase simulation

A simple function can be derived to convert protein concentration to an idealized pulse/chase fraction. Using the notation P for protein, R for any intermediate without P bound and B as the intermediate with P bound, the binding reaction can be presented as

$$\mathrm{P} + \mathrm{R} \longrightarrow \mathrm{B} \tag{B.20}$$

From simple stoichiometry, its clear that

$$b(t) = p_0 - y(t) \tag{B.21}$$

and

$$r(t) = r_0 - b(t) = r_0 - p_0 + y(t) \tag{B.22}$$

where $y(t)$ is the instantaneous protein concentration. We define the pulse/chase ratio to be

$$\chi(t_\mathrm{c}) = \frac{b(t_\mathrm{inc}, t_\mathrm{c})}{b^*(t_\mathrm{inc}, t_\mathrm{c}) + b(t_\mathrm{inc}, t_\mathrm{c})} \tag{B.23}$$

where the * indicates that the species is unlabeled. The time t_c is the time of the addition of chase ^{14}N protein. The time t_inc is the amount of time past the addition of chase protein that the reaction is allowed to progress before measuring the pulse/chase ratio. If we assume that t_inc goes to infinity, we can simplify $\chi(t_\mathrm{c})$ into a function of $y(t_\mathrm{c})$ alone, where the concentration of P is measured immediately before the introduction of the chase.

The pulse/chase ratio can be written in terms of the molar quantities of the intermediates as

$$\chi(t_c) = \frac{N_\mathrm{B}(t_c)}{N_{\mathrm{B}^*}(t_c) + N_\mathrm{B}(t_c)} \tag{B.24}$$

after taking the limit $t_\mathrm{inc} \to \infty$. For brevity we will write the chase addition time as t without the subscript. The molar amount N_B is the sum of the amount of labeled intermediate created before the addition of the chase and the amount created in the presence of the unlabeled protein after chase addition. The molar amount of B at the chase is simply $b(t) \cdot V_0$ using the volume of the system initially. The amount added after the chase, assuming all reactions go to completion, is going to be the amount of ribosomal RNA (rRNA) at the time of the chase converted to labeled intermediate. Assuming that labeled and unlabeled protein bind at the same rate, the amount of unlabeled and labeled intermediate created after completion is the amount of remaining rRNA times the fraction of unlabeled and labeled protein respectively. Thus

$$N_{\mathrm{B}}(t) = b(t)V_0 + r(t)V_0 \cdot \frac{y(t)V_0}{y(t)V_0 + N_{\mathrm{P}}(t)}. \tag{B.25}$$

The total amount of intermediate is

$$N_{\mathrm{B}*}(t) + N_{\mathrm{B}}(t) = r_0 V_0, \tag{B.26}$$

since all rRNA is converted to intermediate. Using Eq. B.24, Eq. B.25, and Eq. B.26 we have

$$\chi(t) = \frac{1}{r_0}\left(b(t) + r(t) \cdot \frac{y(t)}{y(t) + N_{\mathrm{P}}}\right). \tag{B.27}$$

Substituting in Eq. B.21 and Eq. B.22 we get

$$\chi(y) = \frac{1}{r_0}\left(p_0 + y + \frac{(r_0 - p_0 + y)\cdot y}{y + p_0^*}\right) \tag{B.28}$$

where

$$p_0^* = N_{\mathrm{P}^*}(t)/V_0. \tag{B.29}$$

We can then rearrange Eq. B.28 as

$$\chi(y) = \frac{p_0}{p_0^* + p_0} + \frac{p_0^*(p_0^* - r_0 + p_0)}{r_0(p_0^* + p_0)}\left(\frac{p_0 - y}{p_0^* + y}\right) \tag{B.30}$$

showing the contribution of the pulse/chase ratio at $t \to 0$. The error in a Taylor expansion Eq. B.28 around $y(t) = 0$ to first order can be shown to be

$$\epsilon(y(t)) = \frac{(p_0 + p_0^* - r_0)\cdot y(t)^2}{p_0^* \cdot r_0 \cdot (p_0^* + y(t))}. \tag{B.31}$$

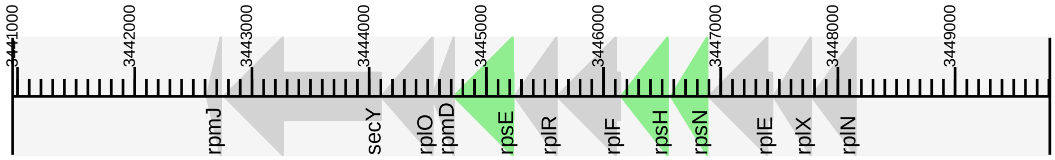
Using $r_0 = 0.305\,\mu\mathrm{M}$, $p_0^* = 2.29\,\mu\mathrm{M}$, and $p_0 = 0.381\,\mu\mathrm{M}$, the maximum error is 18.5%. This means that directly comparing intermediate concentrations to P/C qMS experiments after a reasonable translation and scaling will introduce up to 18.5% error in the comparison.

B.3 Molecular dynamics protocol

Systems were neutralized by placing sodium ions according to the local electrostatic potential of the RNA using Ionize[207]. The systems were carefully solvated in two phases with the TIP3P water model[208]: first, Solvate[209] was used to place the first solvation layers (8 Å), and, second, the Visual Molecular Dynamics solvate plugin[210] to complete the water box with a minimum of 20 Å buffer region on each side. The resulting systems had sizes similar to 1 100 000 atoms.

MD simulations were performed using the latest version of NAMD 2.10[125]. To guarantee correct local solvent density and ion solvation shell around the highly charged backbone and deep groove of the RNA molecules[211–213], all prepared systems were minimized and equilibrated in a step-wise fashion. Minimization was carried out using the conjugate gradient method in NAMD, first with positional constraints on all heavy-atoms for 2000 steps. Constraints were then released for the water molecules for 3000 steps. Protein and nucleic acid side-chains, as well as the ions were set free for the next 5000 steps. Finally all atoms were set free for the last 20 000 steps of minimization. Thermalization was conducted using a temperature jump protocol with step-wise positional restraints to allow waters and ions to diffuse slowly into and pack against the RNA structure. The initial temperature was set to 100K, and ions and heavy atoms in the RNA and protein were harmonically restrained for 25 ps. Then, the temperature was raised to 200K, and ions and the backbone atoms were harmonically restrained for 25 ps. In the next step, the backbone atoms were harmonically restrained at the temperature of 250K for another 50 ps. Force constants for all harmonic restraints were set to 1 kcal·mol^{-1}·Å^{-1}. Finally the temperature was raised up to 300K and all atoms were freed for further equilibration.

B.4 Example of translation rate derivation for *spc* operon



Translation step	Step length/aa	Product
init → *rpsN*	524	uS14
rpsN → *rpsH*	142	uS8
rpsH → *rpsE*	476	uS5
rpsE → term	703	

The translation of the messenger from the *spc* operon is broken into 5 reactions.

Initiation1 The SSU and messenger RNA (mRNA) associate to form pre-translation complex. The rate constant for this reaction is chosen independently of the identity of the messenger.

Initiation2 The large subunit (LSU) associates to the SSU/mRNA complex to form the translating ribosome. The same rate constant for this reaction is for all reactions of this type irrespective of the messenger.

Elongation1 The ribosome translates from the start of the messenger to the end of the first SSU r-protein on the transcript included in our model. The rate for this step is $k_{\mathrm{tl1}}/524$, where k_{tl1} is the nominal translation rate per amino acid (10 a.a./sec), and 524 is the number of intervening codons between the start codon, and the end of the first gene product considered, *rpsN*. This reaction changes the ribosome state placing it now at the beginning of *rpsH* and creates the product of *rpsN*, uS14.

Elongation2 The ribosome is now at the starting position of *rpsH*, the next reaction moves the ribosome to the end of the gene, with no intervening genes to consider. This reaction progresses at the rate $k_{\mathrm{tl1}}/142$, where 142 is the number of codons encoding *rpsH*. The gene product uS8 is produced, and the ribosome is left at the end of r psH.

Elongation3 Two intervening LSU proteins are skipped over and *rpsE* is translated to produce uS5. The distance between the end of *rpsH* and the end of *rpsE* is 476 codons, so the reaction rate is $k_{\mathrm{tl1}}/476$.

Elongation4 There are no remaining SSU proteins are coded for on this messenger, however the time to for the ribosome to process to the end of the mRNA must be accounted for. The reaction rate to the translation termination complex is $k_{\mathrm{tl1}}/703$, where 703 is the length of the final four genes: *rmpD*, *rplO*, *secY*, and *rpmJ*.

Termination Finally, the ribosome has reached the end of the transcript. The translating complex dissociates into 30S, 50S, and mRNA species in a single first-order reaction at a rate independent of the identity of the mRNA.

B.5 Supplementary figures

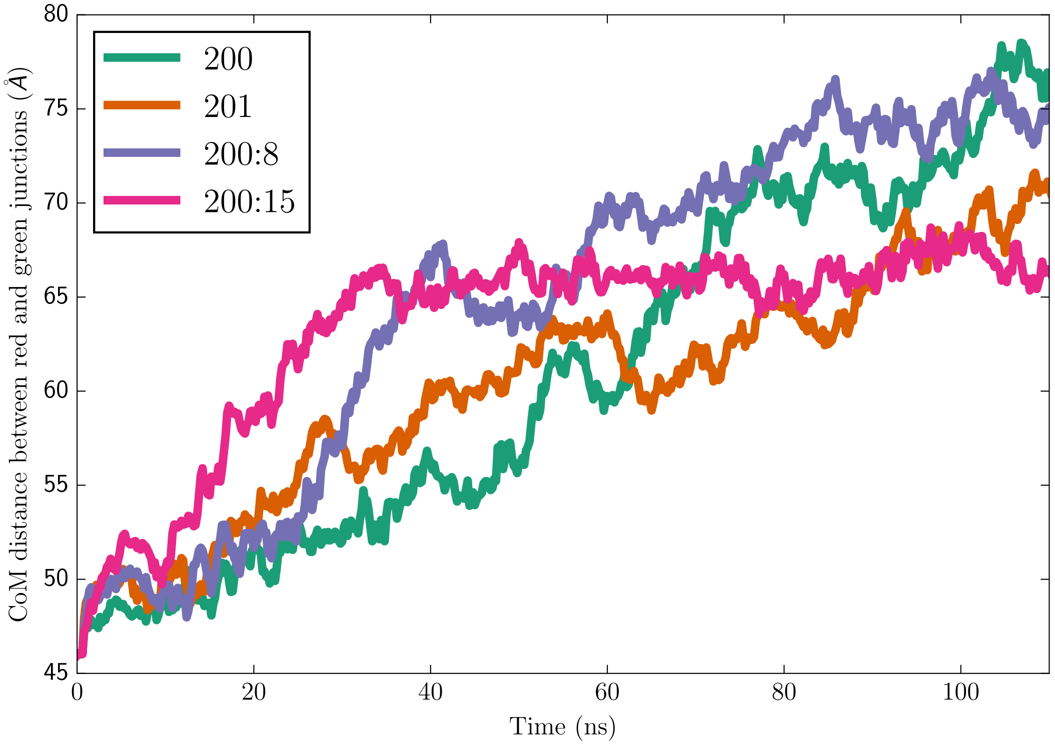


Figure B.1 Separation of junctions in the 3′ domain of *T. thermophilus*. The figure is analogous to Figure 3.5. Starting conformation taken from the PDB: 1HR0. The simulation protocol is identical to the one used for the *E. coli* simulation.

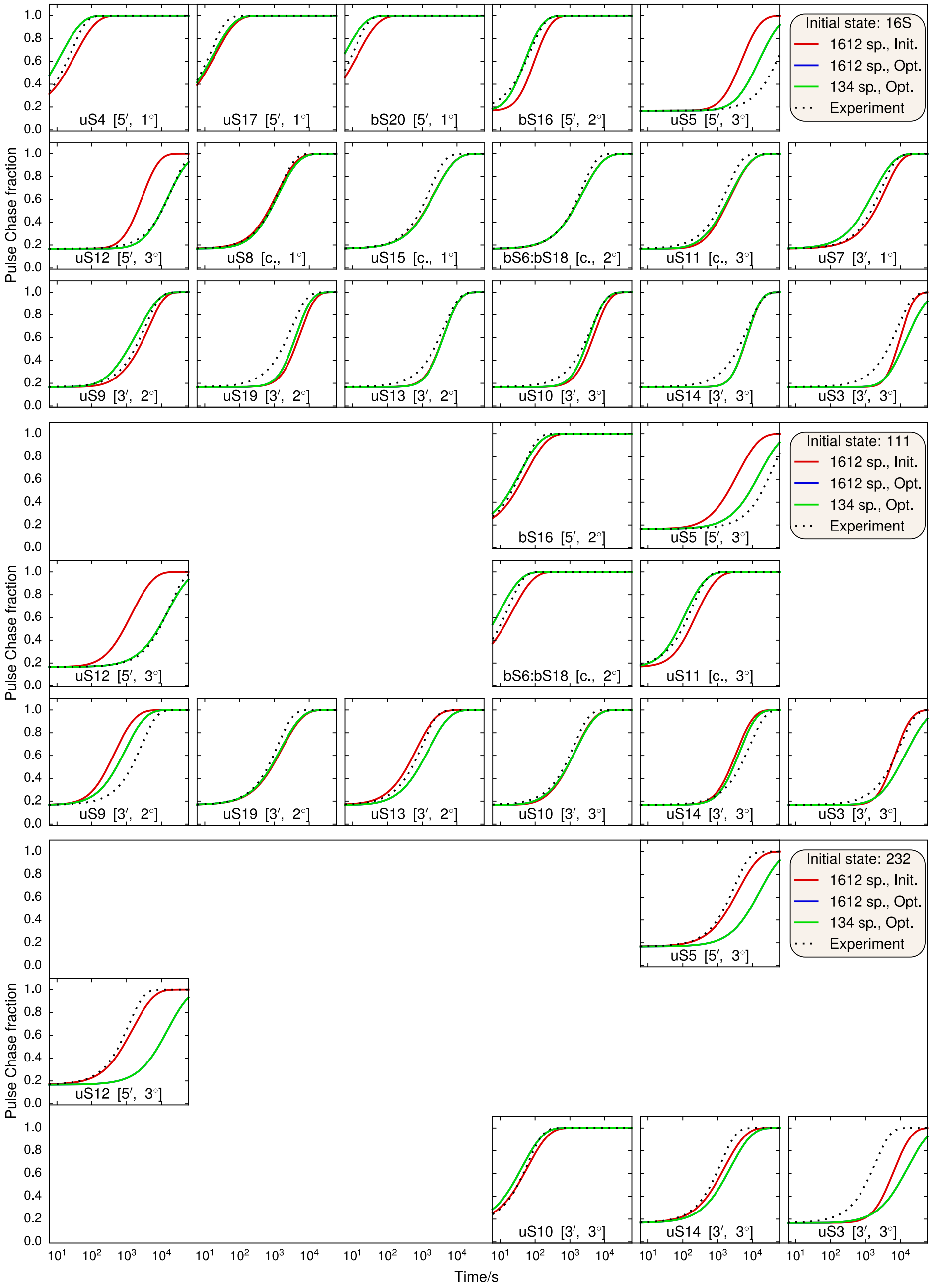


Figure B.2 Simulated P/C qMS fractions from model fit to single-exponential functions from the low temperature experiment. The black dotted curve is the experimental function the model was fit to, the red curves are the results from the model before optimization, the blue curves are the optimized result, and the green curves are the reduced model. The reduced model fits the control experiment data as well as the full model, however it cannot reproduce some of the prebinding experiments since either the prebinding intermediate no longer exists in the model or a linking intermediate was removed.

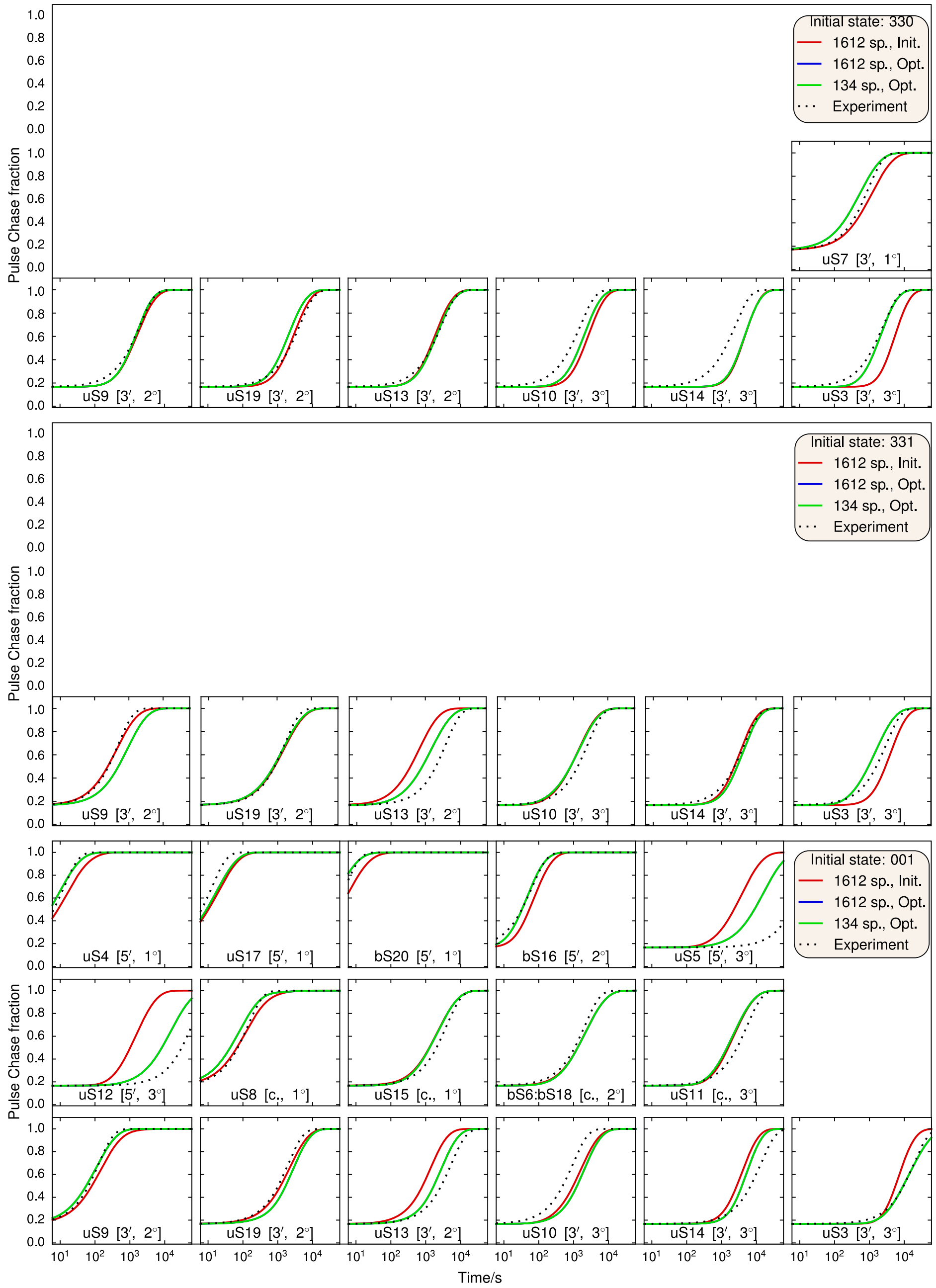


Figure B.2 (cont.) Fit of kinetic model to P/C qMS

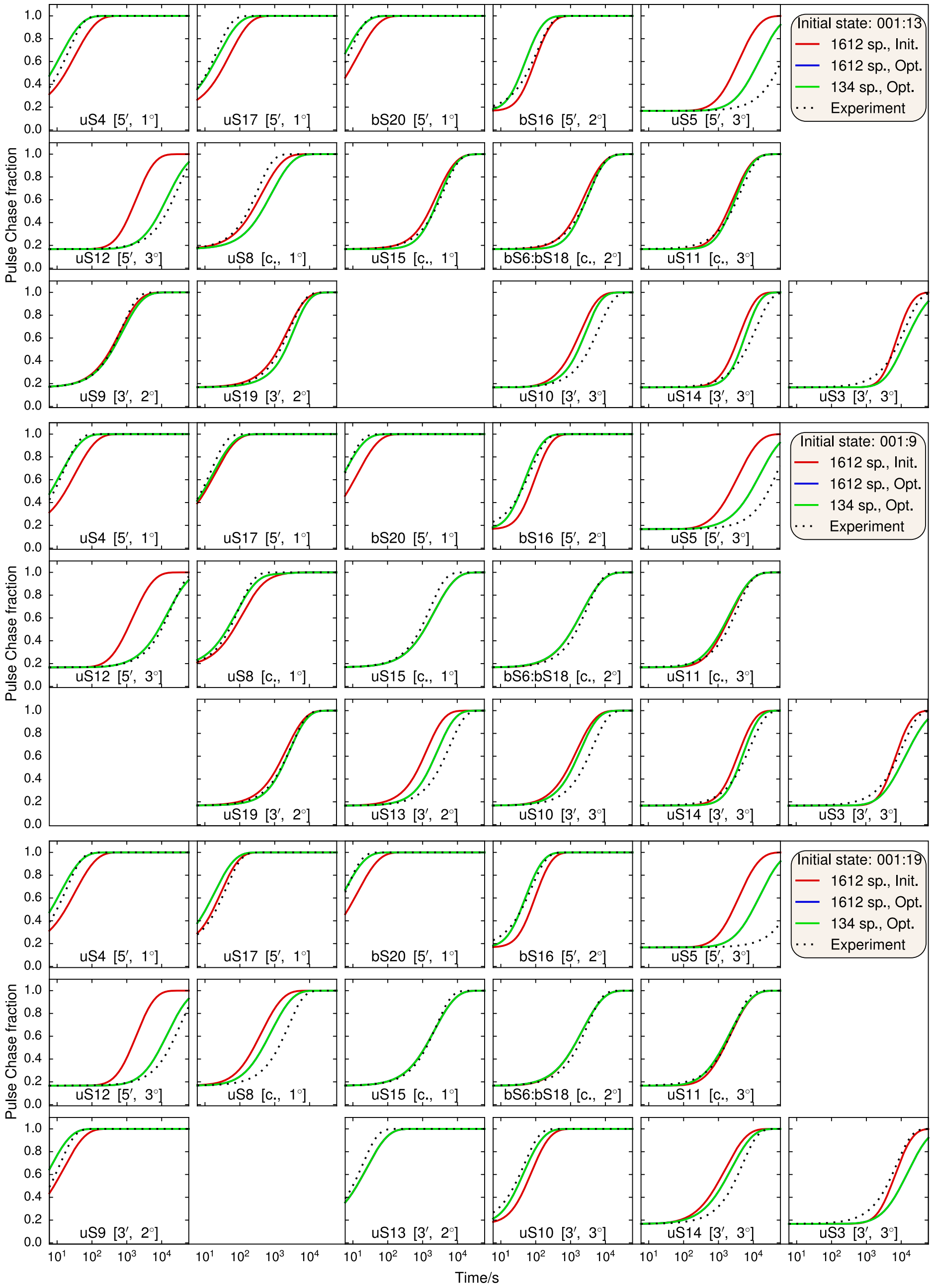


Figure B.2 (cont.) Fit of kinetic model to P/C qMS

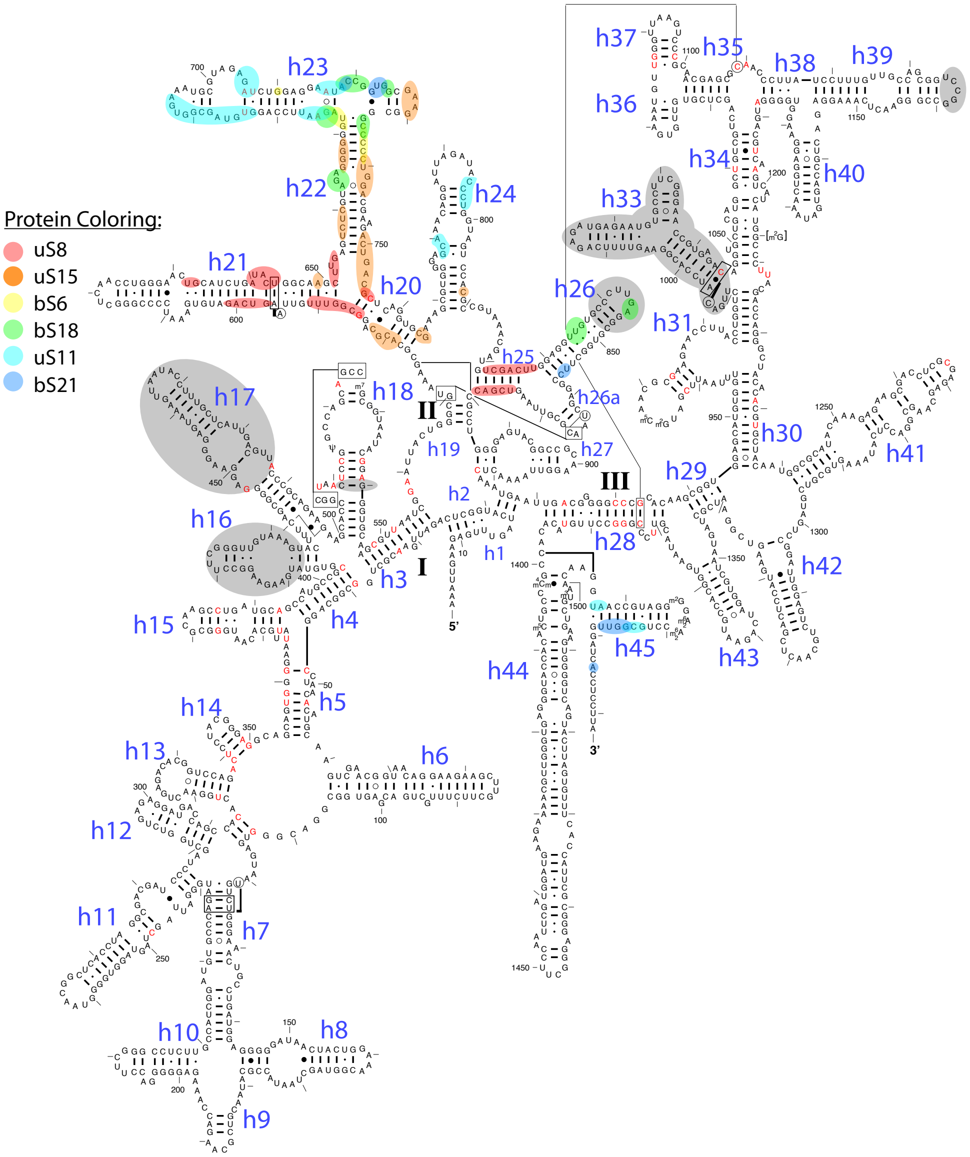


Figure B.3 Secondary structure diagram of *E. coli* with central domain r-protein binding sites (in the folded 30S subunit) labeled. R-protein binding sites determined using a 5Å from the crystal structure PDB: 2I2P[122]. Red letters and gray shapes denote sequence and structural rRNA signatures respectively[96]. Map is based on 16S rRNA map from Cannone et al.[214].

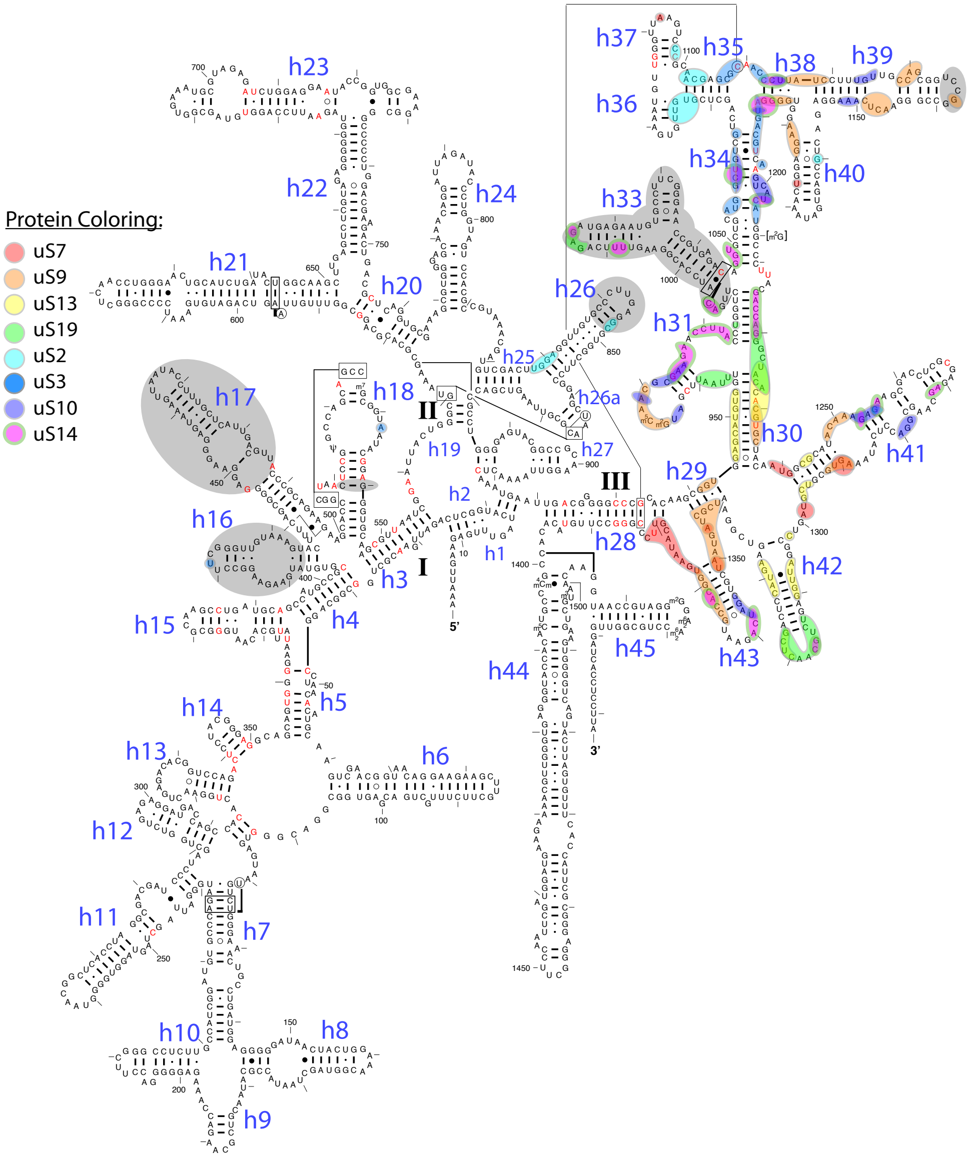


Figure B.3 (cont.) Secondary structure diagram of *E. coli* with 3′ domain r-protein binding sites (in the folded 30S subunit) labeled.

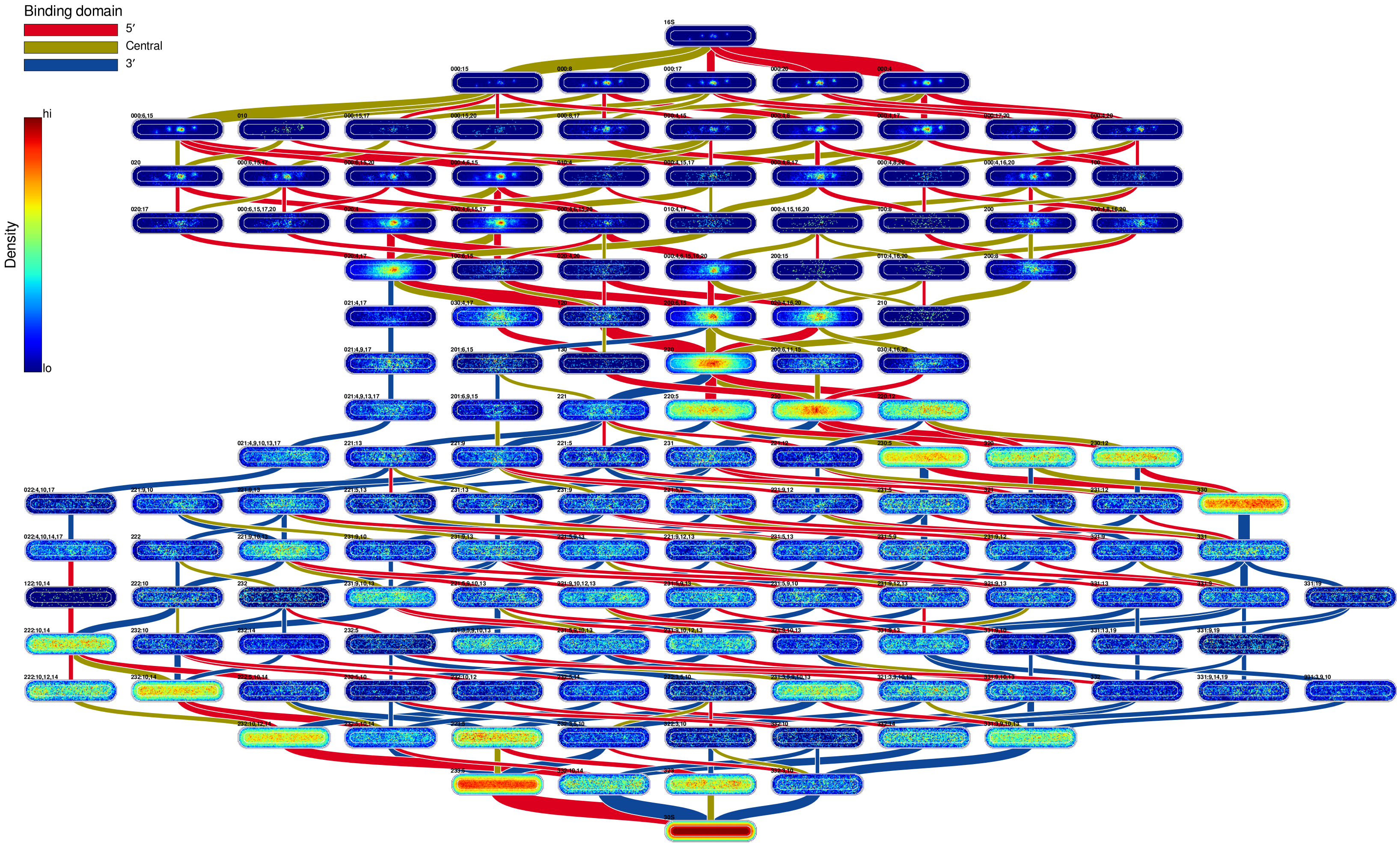


Figure B.4 Assembly map and intermediate distributions for 40 °C *in vivo* model.

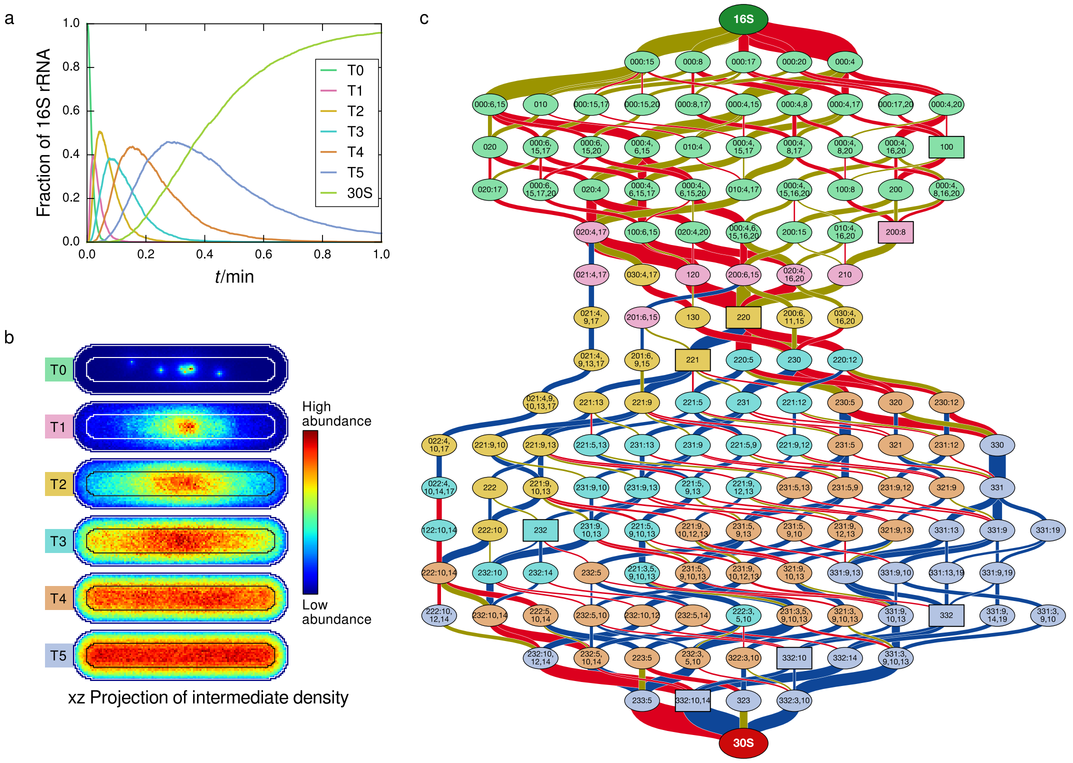


Figure B.5 Temporal clustering of 30S assembly intermediates. (a) Total concentration of each cluster as a function of time (reproduced from Figure 3.9a). (b) Spatial distribution of the temporal classes (reproduced from Figure 3.9b). (c) Assignment of intermediates to temporal classes.

B.6 Supplementary tables

Table B.1 Fold increase in binding rates due to kinetic cooperativity

			Cooperativity in prebinding experiments at 15°C								
	Protein	Min. rate (s^{-1})	Control	1°	1° & 2°	5′ & cent.	5′, cent., & uS7	uS7	uS7 & uS13	uS7 & uS9	uS7 & uS19
5′ domain	uS4	3.833×10^{-2}	~					2.0	~	~	~
	uS17	2.417×10^{-2}	2.7					3.2	~	2.6	~
	bS20	9.167×10^{-2}	~					2.2	~	~	~
	bS16	2.333×10^{-2}	‡	~				‡	‡	‡	‡
	uS5	3.333×10^{-4}	‡	‡	~			‡	‡	‡	‡
	uS12	8.667×10^{-4}	‡	‡	~			‡	‡	‡	‡
central	uS8	4.0×10^{-4}	~					17.9	7.4	25.4	~
	uS15	2.367×10^{-4}	2.5					~	~	2.5	~
	bS6:bS18	5.75×10^{-2}	‡	~				‡	‡	‡	‡
	uS11	5.333×10^{-3}	‡	‡				‡	‡	‡	‡
3′ domain	uS7	3.5×10^{-4}	~			3.0					
	uS9	3.667×10^{-4}	‡	~		‡	5.5	21.4	3.7		222.7
	uS13	1.617×10^{-4}	‡	6.2		2.2(uS7)	~	~		~	319.6
	uS19	3.267×10^{-4}	‡	2.4		‡	~	~	~	~	
	uS3	4.0×10^{-4}	‡	‡	‡	‡	‡	‡	‡	‡	‡
	uS10	2.167×10^{-4}	‡	3.5(uS9)	73.8	2.9(uS9)	‡	4.8(uS9)	‡	~	96.9(uS9)
	uS14	8.0×10^{-4}	‡	‡	~	‡	‡	‡	‡	‡	‡

Fold increase in binding rates due to kinetic cooperativity compared to the minimum observed binding rate for that protein[107]. A ~ or ‡ denote that the ratio of that rate to the minimum rate is less than the 2× threshold. In the case of ‡, this rate was measured for an intermediate not satisfying all thermodynamic dependencies. If a significantly accelerated rate is observed involving a protein without all thermodynamic dependencies satisfied in the initial intermediate configuration, the missing dependencies are given in parenthesis.

Table B.2 Summary of MD simulations performed.

		Bound r-proteins				
Index	States	Central domain	3′ domain domain	Number of atoms	Dimensions	Simulation time (ns)
1	{200}	-	-	1 046 000	182 × 202 × 290	140
2	{201}	-	uS7	1 041 000	181 × 202 × 289	140
3	{200 : 8}	uS8	-	1 027 000	179 × 201 × 289	140
4	{200 : 15}	uS15	-	1 031 000	179 × 201 × 290	140
5	{201 : 8, 19}	uS8	uS7,uS19	1 052 000	180 × 205 × 290	140
6	{201 : 8, 9, 19}	uS8	uS7,uS9,uS19	1 011 000	176 × 200 × 290	140

All systems have the following 5′ domain r-proteins prebound: uS4, uS17, bS20, and bS16.

Table B.3 Rate constants and diffusion parameters used in the *in vivo* model. Citations marked with a dagger indicate that the parameter was either an assumption, a fitting parameter, or a modified value from the literature.

Symbol	Category	Value	Units	Compartments	Citation
$k_{\mathtt{a\_s3\_def}}$	Assembly	0.875	$\mu M^{-1}s^{-1}$	cytoplasm, nucleoid	†
$k_{\mathtt{a\_s4\_def}}$	Assembly	1.693	$\mu M^{-1}s^{-1}$	cytoplasm, nucleoid	†
$k_{\mathtt{a\_s5\_def}}$	Assembly	0.054	$\mu M^{-1}s^{-1}$	cytoplasm, nucleoid	†
$k_{\mathtt{a\_s6\_def}}$	Assembly	31.436	$\mu M^{-1}s^{-1}$	cytoplasm, nucleoid	†
$k_{\mathtt{a\_s7\_def}}$	Assembly	0.041	$\mu M^{-1}s^{-1}$	cytoplasm, nucleoid	†
$k_{\mathtt{a\_s8\_def}}$	Assembly	0.418	$\mu M^{-1}s^{-1}$	cytoplasm, nucleoid	†
$k_{\mathtt{a\_s9\_def}}$	Assembly	0.802	$\mu M^{-1}s^{-1}$	cytoplasm, nucleoid	†
$k_{\mathtt{a\_s10\_def}}$	Assembly	0.474	$\mu M^{-1}s^{-1}$	cytoplasm, nucleoid	†
$k_{\mathtt{a\_s11\_def}}$	Assembly	0.060	$\mu M^{-1}s^{-1}$	cytoplasm, nucleoid	†
$k_{\mathtt{a\_s12\_def}}$	Assembly	0.025	$\mu M^{-1}s^{-1}$	cytoplasm, nucleoid	†
$k_{\mathtt{a\_s13\_def}}$	Assembly	0.531	$\mu M^{-1}s^{-1}$	cytoplasm, nucleoid	†
$k_{\mathtt{a\_s14\_def}}$	Assembly	1.749	$\mu M^{-1}s^{-1}$	cytoplasm, nucleoid	†
$k_{\mathtt{a\_s15\_def}}$	Assembly	0.992	$\mu M^{-1}s^{-1}$	cytoplasm, nucleoid	†
$k_{\mathtt{a\_s16\_def}}$	Assembly	14.290	$\mu M^{-1}s^{-1}$	cytoplasm, nucleoid	†
$k_{\mathtt{a\_s17\_def}}$	Assembly	0.484	$\mu M^{-1}s^{-1}$	cytoplasm, nucleoid	†
$k_{\mathtt{a\_s19\_def}}$	Assembly	0.240	$\mu M^{-1}s^{-1}$	cytoplasm, nucleoid	†
$k_{\mathtt{a\_s20\_def}}$	Assembly	0.301	$\mu M^{-1}s^{-1}$	cytoplasm, nucleoid	†
$k_{\mathtt{lsu\_birth}}$	Assembly	3.08×10^{-4}	$\mu M \cdot s^{-1}$	cytoplasm, nucleoid	†
$k_{\mathtt{dil}}$	Dilution	9.627×10^{-5}	s^{-1}	cytoplasm, nucleoid	†
$k_{\mathtt{deg\_alpha}}$	Degradation	8.363×10^{-4}	s^{-1}	cytoplasm, nucleoid	140
$k_{\mathtt{deg\_rplM}}$	Degradation	1.197×10^{-3}	s^{-1}	cytoplasm, nucleoid	140
$k_{\mathtt{deg\_rpsF}}$	Degradation	8.955×10^{-4}	s^{-1}	cytoplasm, nucleoid	140
$k_{\mathtt{deg\_rpsJ}}$	Degradation	1.029×10^{-3}	s^{-1}	cytoplasm, nucleoid	140
$k_{\mathtt{deg\_rpsO}}$	Degradation	1.238×10^{-3}	s^{-1}	cytoplasm, nucleoid	140
$k_{\mathtt{deg\_rpsP}}$	Degradation	9.785×10^{-4}	s^{-1}	cytoplasm, nucleoid	140
$k_{\mathtt{deg\_rpsT}}$	Degradation	1.144×10^{-3}	s^{-1}	cytoplasm, nucleoid	140
$k_{\mathtt{deg\_spc}}$	Degradation	9.206×10^{-4}	s^{-1}	cytoplasm, nucleoid	140
$k_{\mathtt{deg\_str}}$	Degradation	8.062×10^{-4}	s^{-1}	cytoplasm, nucleoid	140
$k_{\mathtt{s6s18\_assoc}}$	Dimerization	1.000	$\mu M^{-1}s^{-1}$	cytoplasm, nucleoid	141
$k_{\mathtt{s6s18\_dissoc}}$	Dimerization	8.7×10^{-3}	s^{-1}	cytoplasm, nucleoid	126
$k_{\mathtt{ts\_alpha}}$	Transcription	8.33×10^{-3}	s^{-1}	nucleoid	†
$k_{\mathtt{ts\_rplM}}$	Transcription	5.292×10^{-3}	s^{-1}	nucleoid	†
$k_{\mathtt{ts\_rpsF}}$	Transcription	6.468×10^{-3}	s^{-1}	nucleoid	†
$k_{\mathtt{ts\_rpsJ}}$	Transcription	0.011	s^{-1}	nucleoid	†
$k_{\mathtt{ts\_rpsO}}$	Transcription	8.036×10^{-3}	s^{-1}	nucleoid	†
$k_{\mathtt{ts\_rpsP}}$	Transcription	6.86×10^{-3}	s^{-1}	nucleoid	†
$k_{\mathtt{ts\_rpsT}}$	Transcription	4.9×10^{-3}	s^{-1}	nucleoid	†
$k_{\mathtt{ts\_spc}}$	Transcription	0.012	s^{-1}	nucleoid	†
$k_{\mathtt{ts\_str}}$	Transcription	0.010	s^{-1}	nucleoid	†
$k_{\mathtt{ts\_rrnA}}$	Transcription	0.062	s^{-1}	nucleoid	†
$k_{\mathtt{ts\_rrnB}}$	Transcription	0.062	s^{-1}	nucleoid	†
$k_{\mathtt{ts\_rrnC}}$	Transcription	0.062	s^{-1}	nucleoid	†
$k_{\mathtt{ts\_rrnD}}$	Transcription	0.062	s^{-1}	nucleoid	†
$k_{\mathtt{ts\_rrnE}}$	Transcription	0.062	s^{-1}	nucleoid	†
$k_{\mathtt{ts\_rrnG}}$	Transcription	0.062	s^{-1}	nucleoid	†

Table B.3 (continued)

Symbol	Category	Value	Units	Compartments	Citation
$k_{\mathtt{ts\_rrnH}}$	Transcription	0.062	s^{-1}	nucleoid	†
$k_{\mathtt{mrna\_assoc}}$	Translation	100.000	$\mu M^{-1} s^{-1}$	cytoplasm, nucleoid	134
$k_{\mathtt{su\_assoc}}$	Translation	3.000	$\mu M^{-1} s^{-1}$	cytoplasm, nucleoid	135
$k_{\mathtt{su\_dissoc}}$	Translation	0.015	s^{-1}	cytoplasm, nucleoid	133
$k_{\mathtt{tl\_alpha0}}$	Translation	0.134	s^{-1}	cytoplasm, nucleoid	99,118
$k_{\mathtt{tl\_alpha1}}$	Translation	0.118	s^{-1}	cytoplasm, nucleoid	99,118
$k_{\mathtt{tl\_alpha2}}$	Translation	0.073	s^{-1}	cytoplasm, nucleoid	99,118
$k_{\mathtt{tl\_alpha3}}$	Translation	0.033	s^{-1}	cytoplasm, nucleoid	99,118
$k_{\mathtt{tl\_rplM0}}$	Translation	0.057	s^{-1}	cytoplasm, nucleoid	99,118
$k_{\mathtt{tl\_rpsF0}}$	Translation	0.121	s^{-1}	cytoplasm, nucleoid	99,118
$k_{\mathtt{tl\_rpsF1}}$	Translation	0.086	s^{-1}	cytoplasm, nucleoid	99,118
$k_{\mathtt{tl\_rpsF2}}$	Translation	0.098	s^{-1}	cytoplasm, nucleoid	99,118
$k_{\mathtt{tl\_rpsJ0}}$	Translation	0.154	s^{-1}	cytoplasm, nucleoid	99,118
$k_{\mathtt{tl\_rpsJ1}}$	Translation	0.018	s^{-1}	cytoplasm, nucleoid	99,118
$k_{\mathtt{tl\_rpsJ2}}$	Translation	0.045	s^{-1}	cytoplasm, nucleoid	99,118
$k_{\mathtt{tl\_rpsJ3}}$	Translation	0.055	s^{-1}	cytoplasm, nucleoid	99,118
$k_{\mathtt{tl\_rpsO0}}$	Translation	0.178	s^{-1}	cytoplasm, nucleoid	99,118
$k_{\mathtt{tl\_rpsO1}}$	Translation	0.020	s^{-1}	cytoplasm, nucleoid	99,118
$k_{\mathtt{tl\_rpsP0}}$	Translation	0.193	s^{-1}	cytoplasm, nucleoid	99,118
$k_{\mathtt{tl\_rpsP1}}$	Translation	0.027	s^{-1}	cytoplasm, nucleoid	99,118
$k_{\mathtt{tl\_rpsT0}}$	Translation	0.182	s^{-1}	cytoplasm, nucleoid	99,118
$k_{\mathtt{tl\_spc0}}$	Translation	0.031	s^{-1}	cytoplasm, nucleoid	99,118
$k_{\mathtt{tl\_spc1}}$	Translation	0.113	s^{-1}	cytoplasm, nucleoid	99,118
$k_{\mathtt{tl\_spc2}}$	Translation	0.034	s^{-1}	cytoplasm, nucleoid	99,118
$k_{\mathtt{tl\_spc3}}$	Translation	0.023	s^{-1}	cytoplasm, nucleoid	99,118
$k_{\mathtt{tl\_str0}}$	Translation	0.128	s^{-1}	cytoplasm, nucleoid	99,118
$k_{\mathtt{tl\_str1}}$	Translation	0.075	s^{-1}	cytoplasm, nucleoid	99,118
$k_{\mathtt{tl\_str2}}$	Translation	0.014	s^{-1}	cytoplasm, nucleoid	99,118
$D_{\mathtt{30Scyt}}$	Diffusion	0.400	$\mu m^2 s^{-1}$	cytoplasm	113, †
$D_{\mathtt{30Snuc}}$	Diffusion	0.040	$\mu m^2 s^{-1}$	nucleoid	113, †
$D_{\mathtt{50Scyt}}$	Diffusion	0.400	$\mu m^2 s^{-1}$	cytoplasm	113, †
$D_{\mathtt{50Snuc}}$	Diffusion	0.040	$\mu m^2 s^{-1}$	nucleoid	113, †
$D_{\mathtt{uS3}}$	Diffusion	2.605	$\mu m^2 s^{-1}$	cytoplasm, nucleoid	136,†
$D_{\mathtt{uS4}}$	Diffusion	2.853	$\mu m^2 s^{-1}$	cytoplasm, nucleoid	136,†
$D_{\mathtt{uS5}}$	Diffusion	3.668	$\mu m^2 s^{-1}$	cytoplasm, nucleoid	136,†
$D_{\mathtt{bS6}}$	Diffusion	4.161	$\mu m^2 s^{-1}$	cytoplasm, nucleoid	136,†
$D_{\mathtt{uS7}}$	Diffusion	3.282	$\mu m^2 s^{-1}$	cytoplasm, nucleoid	136,†
$D_{\mathtt{uS8}}$	Diffusion	4.422	$\mu m^2 s^{-1}$	cytoplasm, nucleoid	136,†
$D_{\mathtt{uS9}}$	Diffusion	4.239	$\mu m^2 s^{-1}$	cytoplasm, nucleoid	136,†
$D_{\mathtt{uS10}}$	Diffusion	5.159	$\mu m^2 s^{-1}$	cytoplasm, nucleoid	136,†
$D_{\mathtt{uS11}}$	Diffusion	4.497	$\mu m^2 s^{-1}$	cytoplasm, nucleoid	136,†
$D_{\mathtt{uS12}}$	Diffusion	4.527	$\mu m^2 s^{-1}$	cytoplasm, nucleoid	136,†
$D_{\mathtt{uS13}}$	Diffusion	4.720	$\mu m^2 s^{-1}$	cytoplasm, nucleoid	136,†
$D_{\mathtt{uS14}}$	Diffusion	5.215	$\mu m^2 s^{-1}$	cytoplasm, nucleoid	136,†
$D_{\mathtt{uS15}}$	Diffusion	5.753	$\mu m^2 s^{-1}$	cytoplasm, nucleoid	136,†
$D_{\mathtt{bS16}}$	Diffusion	6.291	$\mu m^2 s^{-1}$	cytoplasm, nucleoid	136,†
$D_{\mathtt{uS17}}$	Diffusion	6.022	$\mu m^2 s^{-1}$	cytoplasm, nucleoid	136,†
$D_{\mathtt{bS18}}$	Diffusion	6.405	$\mu m^2 s^{-1}$	cytoplasm, nucleoid	136,†
$D_{\mathtt{uS19}}$	Diffusion	5.681	$\mu m^2 s^{-1}$	cytoplasm, nucleoid	136,†

Table B.3 (continued)

Symbol	Category	Value	Units	Compartments	Citation
$D_{\texttt{bS20}}$	Diffusion	6.041	µm^2 s^{-1}	cytoplasm, nucleoid	136,[†]
$D_{\texttt{bS6:bS18}}$	Diffusion	2.779	µm^2 s^{-1}	cytoplasm, nucleoid	136,[†]
$D_{\texttt{imt00cyt}}$	Diffusion	0.150	µm^2 s^{-1}	cytoplasm	113,138,[†]
$D_{\texttt{imt00nuc}}$	Diffusion	0.015	µm^2 s^{-1}	nucleoid	113,138,[†]
$D_{\texttt{imt01cyt}}$	Diffusion	0.165	µm^2 s^{-1}	cytoplasm	113,138,[†]
$D_{\texttt{imt01nuc}}$	Diffusion	0.016	µm^2 s^{-1}	nucleoid	113,138,[†]
$D_{\texttt{imt02cyt}}$	Diffusion	0.179	µm^2 s^{-1}	cytoplasm	113,138,[†]
$D_{\texttt{imt02nuc}}$	Diffusion	0.018	µm^2 s^{-1}	nucleoid	113,138,[†]
$D_{\texttt{imt03cyt}}$	Diffusion	0.194	µm^2 s^{-1}	cytoplasm	113,138,[†]
$D_{\texttt{imt03nuc}}$	Diffusion	0.019	µm^2 s^{-1}	nucleoid	113,138,[†]
$D_{\texttt{imt04cyt}}$	Diffusion	0.209	µm^2 s^{-1}	cytoplasm	113,138,[†]
$D_{\texttt{imt04nuc}}$	Diffusion	0.021	µm^2 s^{-1}	nucleoid	113,138,[†]
$D_{\texttt{imt05cyt}}$	Diffusion	0.223	µm^2 s^{-1}	cytoplasm	113,138,[†]
$D_{\texttt{imt05nuc}}$	Diffusion	0.022	µm^2 s^{-1}	nucleoid	113,138,[†]
$D_{\texttt{imt06cyt}}$	Diffusion	0.238	µm^2 s^{-1}	cytoplasm	113,138,[†]
$D_{\texttt{imt06nuc}}$	Diffusion	0.024	µm^2 s^{-1}	nucleoid	113,138,[†]
$D_{\texttt{imt07cyt}}$	Diffusion	0.253	µm^2 s^{-1}	cytoplasm	113,138,[†]
$D_{\texttt{imt07nuc}}$	Diffusion	0.025	µm^2 s^{-1}	nucleoid	113,138,[†]
$D_{\texttt{imt08cyt}}$	Diffusion	0.268	µm^2 s^{-1}	cytoplasm	113,138,[†]
$D_{\texttt{imt08nuc}}$	Diffusion	0.027	µm^2 s^{-1}	nucleoid	113,138,[†]
$D_{\texttt{imt09cyt}}$	Diffusion	0.282	µm^2 s^{-1}	cytoplasm	113,138,[†]
$D_{\texttt{imt09nuc}}$	Diffusion	0.028	µm^2 s^{-1}	nucleoid	113,138,[†]
$D_{\texttt{imt10cyt}}$	Diffusion	0.297	µm^2 s^{-1}	cytoplasm	113,138,[†]
$D_{\texttt{imt10nuc}}$	Diffusion	0.030	µm^2 s^{-1}	nucleoid	113,138,[†]
$D_{\texttt{imt11cyt}}$	Diffusion	0.312	µm^2 s^{-1}	cytoplasm	[†]
$D_{\texttt{imt11nuc}}$	Diffusion	0.031	µm^2 s^{-1}	nucleoid	113,138,[†]
$D_{\texttt{imt12cyt}}$	Diffusion	0.326	µm^2 s^{-1}	cytoplasm	113,138,[†]
$D_{\texttt{imt12nuc}}$	Diffusion	0.033	µm^2 s^{-1}	nucleoid	113,138,[†]
$D_{\texttt{imt13cyt}}$	Diffusion	0.341	µm^2 s^{-1}	cytoplasm	113,138,[†]
$D_{\texttt{imt13nuc}}$	Diffusion	0.034	µm^2 s^{-1}	nucleoid	113,138,[†]
$D_{\texttt{imt14cyt}}$	Diffusion	0.356	µm^2 s^{-1}	cytoplasm	113,138,[†]
$D_{\texttt{imt14nuc}}$	Diffusion	0.036	µm^2 s^{-1}	nucleoid	113,138,[†]
$D_{\texttt{imt15cyt}}$	Diffusion	0.371	µm^2 s^{-1}	cytoplasm	113,138,[†]
$D_{\texttt{imt15nuc}}$	Diffusion	0.037	µm^2 s^{-1}	nucleoid	113,138,[†]
$D_{\texttt{imt16cyt}}$	Diffusion	0.385	µm^2 s^{-1}	cytoplasm	113,138,[†]
$D_{\texttt{imt16nuc}}$	Diffusion	0.039	µm^2 s^{-1}	nucleoid	113,138,[†]
$D_{\texttt{mRNA}}$	Diffusion	0.300	µm^2 s^{-1}	cytoplasm, nucleoid	137
$D_{\texttt{operon}}$	Diffusion	0.0	µm^2 s^{-1}	nucleoid	[†]
$D_{\texttt{ribosomeCyt}}$	Diffusion	0.055	µm^2 s^{-1}	cytoplasm	113
$D_{\texttt{ribosomeNuc}}$	Diffusion	5.5×10^{-3}	µm^2 s^{-1}	nucleoid	113,[†]
$\Gamma_{\texttt{uS3}}$	Compartment Transition	2.605	µm^2 s^{-1}	cytoplasm, nucleoid	136,[†]
$\Gamma_{\texttt{uS4}}$	Compartment Transition	2.853	µm^2 s^{-1}	cytoplasm, nucleoid	136,[†]
$\Gamma_{\texttt{uS5}}$	Compartment Transition	3.668	µm^2 s^{-1}	cytoplasm, nucleoid	136,[†]
$\Gamma_{\texttt{bS6}}$	Compartment Transition	4.161	µm^2 s^{-1}	cytoplasm, nucleoid	136,[†]
$\Gamma_{\texttt{bS6:bS18}}$	Compartment Transition	2.779	µm^2 s^{-1}	cytoplasm, nucleoid	136,[†]
$\Gamma_{\texttt{uS7}}$	Compartment Transition	3.282	µm^2 s^{-1}	cytoplasm, nucleoid	136,[†]
$\Gamma_{\texttt{uS8}}$	Compartment Transition	4.422	µm^2 s^{-1}	cytoplasm, nucleoid	136,[†]
$\Gamma_{\texttt{uS9}}$	Compartment Transition	4.239	µm^2 s^{-1}	cytoplasm, nucleoid	136,[†]
$\Gamma_{\texttt{uS10}}$	Compartment Transition	5.159	µm^2 s^{-1}	cytoplasm, nucleoid	136,[†]

Table B.3 (continued)

Symbol	Category	Value	Units	Compartments	Citation
$\Gamma_{\texttt{uS11}}$	Compartment Transition	4.497	$\mu\mathrm{m}^2\,\mathrm{s}^{-1}$	`cytoplasm, nucleoid`	136,†
$\Gamma_{\texttt{uS12}}$	Compartment Transition	4.527	$\mu\mathrm{m}^2\,\mathrm{s}^{-1}$	`cytoplasm, nucleoid`	136,†
$\Gamma_{\texttt{uS13}}$	Compartment Transition	4.720	$\mu\mathrm{m}^2\,\mathrm{s}^{-1}$	`cytoplasm, nucleoid`	136,†
$\Gamma_{\texttt{uS14}}$	Compartment Transition	5.215	$\mu\mathrm{m}^2\,\mathrm{s}^{-1}$	`cytoplasm, nucleoid`	136,†
$\Gamma_{\texttt{uS15}}$	Compartment Transition	5.753	$\mu\mathrm{m}^2\,\mathrm{s}^{-1}$	`cytoplasm, nucleoid`	136,†
$\Gamma_{\texttt{bS16}}$	Compartment Transition	6.291	$\mu\mathrm{m}^2\,\mathrm{s}^{-1}$	`cytoplasm, nucleoid`	136,†
$\Gamma_{\texttt{uS17}}$	Compartment Transition	6.022	$\mu\mathrm{m}^2\,\mathrm{s}^{-1}$	`cytoplasm, nucleoid`	136,†
$\Gamma_{\texttt{bS18}}$	Compartment Transition	6.405	$\mu\mathrm{m}^2\,\mathrm{s}^{-1}$	`cytoplasm, nucleoid`	136,†
$\Gamma_{\texttt{uS19}}$	Compartment Transition	5.681	$\mu\mathrm{m}^2\,\mathrm{s}^{-1}$	`cytoplasm, nucleoid`	136,†
$\Gamma_{\texttt{bS20}}$	Compartment Transition	6.041	$\mu\mathrm{m}^2\,\mathrm{s}^{-1}$	`cytoplasm, nucleoid`	136,†
$\Gamma_{\texttt{imt00}}$	Compartment Transition	0.047	$\mu\mathrm{m}^2\,\mathrm{s}^{-1}$	`cytoplasm, nucleoid`	113,138,†
$\Gamma_{\texttt{imt01}}$	Compartment Transition	0.052	$\mu\mathrm{m}^2\,\mathrm{s}^{-1}$	`cytoplasm, nucleoid`	113,138,†
$\Gamma_{\texttt{imt02}}$	Compartment Transition	0.057	$\mu\mathrm{m}^2\,\mathrm{s}^{-1}$	`cytoplasm, nucleoid`	113,138,†
$\Gamma_{\texttt{imt03}}$	Compartment Transition	0.061	$\mu\mathrm{m}^2\,\mathrm{s}^{-1}$	`cytoplasm, nucleoid`	113,138,†
$\Gamma_{\texttt{imt04}}$	Compartment Transition	0.066	$\mu\mathrm{m}^2\,\mathrm{s}^{-1}$	`cytoplasm, nucleoid`	113,138,†
$\Gamma_{\texttt{imt05}}$	Compartment Transition	0.071	$\mu\mathrm{m}^2\,\mathrm{s}^{-1}$	`cytoplasm, nucleoid`	113,138,†
$\Gamma_{\texttt{imt06}}$	Compartment Transition	0.075	$\mu\mathrm{m}^2\,\mathrm{s}^{-1}$	`cytoplasm, nucleoid`	113,138,†
$\Gamma_{\texttt{imt07}}$	Compartment Transition	0.080	$\mu\mathrm{m}^2\,\mathrm{s}^{-1}$	`cytoplasm, nucleoid`	113,138,†
$\Gamma_{\texttt{imt08}}$	Compartment Transition	0.085	$\mu\mathrm{m}^2\,\mathrm{s}^{-1}$	`cytoplasm, nucleoid`	113,138,†
$\Gamma_{\texttt{imt09}}$	Compartment Transition	0.089	$\mu\mathrm{m}^2\,\mathrm{s}^{-1}$	`cytoplasm, nucleoid`	113,138,†
$\Gamma_{\texttt{imt10}}$	Compartment Transition	0.099	$\mu\mathrm{m}^2\,\mathrm{s}^{-1}$	`cytoplasm, nucleoid`	113,138,†
$\Gamma_{\texttt{imt11}}$	Compartment Transition	0.103	$\mu\mathrm{m}^2\,\mathrm{s}^{-1}$	`cytoplasm, nucleoid`	113,138,†
$\Gamma_{\texttt{imt12}}$	Compartment Transition	0.108	$\mu\mathrm{m}^2\,\mathrm{s}^{-1}$	`cytoplasm, nucleoid`	113,138,†
$\Gamma_{\texttt{imt13}}$	Compartment Transition	0.113	$\mu\mathrm{m}^2\,\mathrm{s}^{-1}$	`cytoplasm, nucleoid`	113,138,†
$\Gamma_{\texttt{imt14}}$	Compartment Transition	0.117	$\mu\mathrm{m}^2\,\mathrm{s}^{-1}$	`cytoplasm, nucleoid`	113,138,†
$\Gamma_{\texttt{imt15}}$	Compartment Transition	0.122	$\mu\mathrm{m}^2\,\mathrm{s}^{-1}$	`cytoplasm, nucleoid`	113,138,†
$\Gamma_{\texttt{subunit}}$	Compartment Transition	0.126	$\mu\mathrm{m}^2\,\mathrm{s}^{-1}$	`cytoplasm, nucleoid`	113,†
$\Gamma_{\texttt{rib}\rightarrow\texttt{nuc}}$	Compartment Transition	4.348×10^{-3}	$\mu\mathrm{m}^2\,\mathrm{s}^{-1}$	`cytoplasm`	113,†
$\Gamma_{\texttt{rib}\rightarrow\texttt{cyt}}$	Compartment Transition	0.017×10^{-3}	$\mu\mathrm{m}^2\,\mathrm{s}^{-1}$	`nucleoid`	113,†
$\Gamma_{\texttt{mRNA}}$	Compartment Transition	0.300	$\mu\mathrm{m}^2\,\mathrm{s}^{-1}$	`cytoplasm, nucleoid`	137

Table B.4 Initial and final counts for all species in *in vivo* model.

		Initial count		Final count	
Species	Type	`cytoplasm`	`nucleoid`	`cytoplasm`	`nucleoid`
uS3	Protein	1454	439	2606	828
uS4	Protein	1460	442	3185	949
uS5	Protein	1405	426	2927	895
bS6	Protein	105	32	151	40
uS7	Protein	1400	424	2408	743
uS8	Protein	1431	432	2942	924
uS9	Protein	1196	363	2694	868
uS10	Protein	1497	454	2708	810
uS11	Protein	1428	433	3110	990
uS12	Protein	1402	426	2427	738
uS13	Protein	1446	438	3184	941
uS14	Protein	1417	429	2987	869

Table B.4 (continued)

		Initial count		Final count	
Species	Type	`cytoplasm`	`nucleoid`	`cytoplasm`	`nucleoid`
uS15	Protein	1296	393	2694	803
bS16	Protein	1394	421	2530	765
uS17	Protein	1445	438	2599	808
bS18	Protein	103	31	136	50
uS19	Protein	1477	448	2656	817
bS20	Protein	1437	434	3371	961
bS6:bS18	Protein Dimer	1694	514	3739	1058
d_{alpha}	Operon	0	1	0	1
d_{rplM}	Operon	0	1	0	1
d_{rpsF}	Operon	0	1	0	1
d_{rpsJ}	Operon	0	1	0	1
d_{rpsO}	Operon	0	1	0	1
d_{rpsP}	Operon	0	1	0	1
d_{rpsT}	Operon	0	1	0	1
d_{rrnA}	Operon	0	1	0	1
d_{rrnB}	Operon	0	1	0	1
d_{rrnC}	Operon	0	1	0	1
d_{rrnD}	Operon	0	1	0	1
d_{rrnE}	Operon	0	1	0	1
d_{rrnG}	Operon	0	1	0	1
d_{rrnH}	Operon	0	1	0	1
d_{spc}	Operon	0	1	0	1
d_{str}	Operon	0	1	0	1
m_{alpha}	mRNA	0	0	0	0
m_{rplM}	mRNA	0	0	0	0
m_{rpsF}	mRNA	0	0	0	0
m_{rpsJ}	mRNA	0	0	0	0
m_{rpsO}	mRNA	0	0	0	0
m_{rpsP}	mRNA	0	0	0	0
m_{rpsT}	mRNA	0	0	0	0
m_{spc}	mRNA	0	0	0	0
m_{str}	mRNA	0	0	0	0
16S	rRNA	0	0	0	0
$\{000\colon 15, 20\}$	30S Assembly Intermediate	0	0	0	0
$\{000\colon 15\}$	30S Assembly Intermediate	0	0	0	0
$\{000\colon 15, 17\}$	30S Assembly Intermediate	0	0	0	0
$\{000\colon 17, 20\}$	30S Assembly Intermediate	0	0	0	0
$\{000\colon 17\}$	30S Assembly Intermediate	0	0	0	0
$\{000\colon 20\}$	30S Assembly Intermediate	0	0	0	0
$\{000\colon 4\}$	30S Assembly Intermediate	0	0	0	0
$\{000\colon 4, 17\}$	30S Assembly Intermediate	0	0	0	0
$\{000\colon 4, 8\}$	30S Assembly Intermediate	0	0	0	0
$\{000\colon 4, 8, 20\}$	30S Assembly Intermediate	0	0	0	0
$\{000\colon 4, 8, 17\}$	30S Assembly Intermediate	0	0	0	0
$\{000\colon 4, 20\}$	30S Assembly Intermediate	0	0	0	0
$\{000\colon 4, 16, 20\}$	30S Assembly Intermediate	0	0	0	0
$\{000\colon 4, 8, 16, 20\}$	30S Assembly Intermediate	0	0	0	0
$\{000\colon 4, 15\}$	30S Assembly Intermediate	0	0	0	0
$\{000\colon 4, 15, 17\}$	30S Assembly Intermediate	0	0	0	0

Table B.4 (continued)

Species	Type	Initial count		Final count	
		cytoplasm	nucleoid	cytoplasm	nucleoid
{000: 4, 6, 15}	30S Assembly Intermediate	0	0	0	0
{000: 4, 6, 15, 20}	30S Assembly Intermediate	0	0	0	0
{000: 4, 6, 15, 17}	30S Assembly Intermediate	0	0	0	0
{000: 4, 6, 15, 16, 20}	30S Assembly Intermediate	0	0	0	0
{000: 4, 15, 16, 20}	30S Assembly Intermediate	0	0	0	0
{000: 6, 15}	30S Assembly Intermediate	0	0	0	0
{000: 6, 15, 20}	30S Assembly Intermediate	0	0	0	0
{000: 6, 15, 17}	30S Assembly Intermediate	0	0	0	0
{000: 6, 15, 17, 20}	30S Assembly Intermediate	0	0	0	0
{000: 8}	30S Assembly Intermediate	0	0	0	0
{000: 8, 17}	30S Assembly Intermediate	0	0	0	0
{010}	30S Assembly Intermediate	0	0	0	0
{010: 4}	30S Assembly Intermediate	0	0	0	0
{010: 4, 17}	30S Assembly Intermediate	0	0	0	0
{010: 4, 16, 20}	30S Assembly Intermediate	0	0	0	0
{020}	30S Assembly Intermediate	0	0	0	0
{020: 17}	30S Assembly Intermediate	0	0	0	0
{020: 4}	30S Assembly Intermediate	0	0	0	0
{020: 4, 20}	30S Assembly Intermediate	0	0	0	0
{020: 4, 17}	30S Assembly Intermediate	0	0	0	0
{020: 4, 16, 20}	30S Assembly Intermediate	0	0	0	0
{021: 4, 17}	30S Assembly Intermediate	0	0	0	0
{021: 4, 9, 17}	30S Assembly Intermediate	0	0	0	0
{021: 4, 9, 13, 17}	30S Assembly Intermediate	0	0	0	0
{021: 4, 9, 10, 13, 17}	30S Assembly Intermediate	0	0	0	0
{022: 4, 10, 14, 17}	30S Assembly Intermediate	0	0	0	1
{022: 4, 10, 17}	30S Assembly Intermediate	0	0	0	0
{030: 4, 17}	30S Assembly Intermediate	0	0	0	0
{030: 4, 16, 20}	30S Assembly Intermediate	0	0	0	0
{100}	30S Assembly Intermediate	0	0	0	0
{100: 6, 15}	30S Assembly Intermediate	0	0	0	0
{100: 8}	30S Assembly Intermediate	0	0	0	0
{120}	30S Assembly Intermediate	0	0	0	0
{122: 10, 14}	30S Assembly Intermediate	0	0	0	0
{130}	30S Assembly Intermediate	0	0	0	0
{200}	30S Assembly Intermediate	0	0	0	0
{200: 15}	30S Assembly Intermediate	0	0	0	0
{200: 6, 15}	30S Assembly Intermediate	0	0	0	0
{200: 6, 11, 15}	30S Assembly Intermediate	0	0	0	0
{200: 8}	30S Assembly Intermediate	0	0	0	0
{201: 6, 15}	30S Assembly Intermediate	0	0	0	0
{201: 6, 9, 15}	30S Assembly Intermediate	0	0	0	0
{210}	30S Assembly Intermediate	0	0	0	0
{220}	30S Assembly Intermediate	0	0	0	0
{220: 12}	30S Assembly Intermediate	0	0	0	0
{220: 5}	30S Assembly Intermediate	0	0	0	0
{221}	30S Assembly Intermediate	0	0	0	0
{221: 12}	30S Assembly Intermediate	0	0	0	0
{221: 13}	30S Assembly Intermediate	0	0	0	0
{221: 3, 5, 9, 10, 13}	30S Assembly Intermediate	0	0	0	0
{221: 5}	30S Assembly Intermediate	0	0	0	0

Table B.4 (continued)

		Initial count		Final count	
Species	Type	`cytoplasm`	`nucleoid`	`cytoplasm`	`nucleoid`
{221: 5, 9}	30S Assembly Intermediate	0	0	0	0
{221: 5, 13}	30S Assembly Intermediate	0	0	0	0
{221: 5, 9, 13}	30S Assembly Intermediate	0	0	0	0
{221: 5, 9, 10, 13}	30S Assembly Intermediate	0	0	0	0
{221: 9}	30S Assembly Intermediate	0	0	0	0
{221: 9, 10}	30S Assembly Intermediate	0	0	0	0
{221: 9, 12}	30S Assembly Intermediate	0	0	0	0
{221: 9, 13}	30S Assembly Intermediate	0	0	0	0
{221: 9, 10, 13}	30S Assembly Intermediate	0	0	0	0
{221: 9, 12, 13}	30S Assembly Intermediate	0	0	0	0
{221: 9, 10, 12, 13}	30S Assembly Intermediate	0	0	0	0
{222}	30S Assembly Intermediate	0	0	0	0
{222: 10, 14}	30S Assembly Intermediate	0	0	0	1
{222: 10}	30S Assembly Intermediate	0	0	0	0
{222: 10, 12, 14}	30S Assembly Intermediate	0	0	0	0
{222: 3, 5, 10}	30S Assembly Intermediate	0	0	0	0
{222: 5, 10, 14}	30S Assembly Intermediate	0	0	0	0
{223: 5}	30S Assembly Intermediate	0	0	0	0
{230}	30S Assembly Intermediate	0	0	0	0
{230: 12}	30S Assembly Intermediate	0	0	0	0
{230: 5}	30S Assembly Intermediate	0	0	0	0
{231}	30S Assembly Intermediate	0	0	0	0
{231: 12}	30S Assembly Intermediate	0	0	0	0
{231: 13}	30S Assembly Intermediate	0	0	0	0
{231: 3, 5, 9, 10, 13}	30S Assembly Intermediate	0	0	0	0
{231: 5}	30S Assembly Intermediate	0	0	0	0
{231: 5, 9}	30S Assembly Intermediate	0	0	0	0
{231: 5, 13}	30S Assembly Intermediate	0	0	0	0
{231: 5, 9, 10}	30S Assembly Intermediate	0	0	0	0
{231: 5, 9, 13}	30S Assembly Intermediate	0	0	0	0
{231: 5, 9, 10, 13}	30S Assembly Intermediate	0	0	0	0
{231: 9}	30S Assembly Intermediate	0	0	0	0
{231: 9, 10}	30S Assembly Intermediate	0	0	0	0
{231: 9, 12}	30S Assembly Intermediate	0	0	0	0
{231: 9, 13}	30S Assembly Intermediate	0	0	0	0
{231: 9, 10, 13}	30S Assembly Intermediate	0	0	0	0
{231: 9, 12, 13}	30S Assembly Intermediate	0	0	0	0
{231: 9, 10, 12, 13}	30S Assembly Intermediate	0	0	0	0
{232}	30S Assembly Intermediate	0	0	0	0
{232: 10}	30S Assembly Intermediate	0	0	0	0
{232: 10, 12}	30S Assembly Intermediate	0	0	0	0
{232: 10, 14}	30S Assembly Intermediate	0	0	0	0
{232: 10, 12, 14}	30S Assembly Intermediate	0	0	0	0
{232: 14}	30S Assembly Intermediate	0	0	0	0
{232: 3, 5, 10}	30S Assembly Intermediate	0	0	0	0
{232: 5}	30S Assembly Intermediate	0	0	0	0
{232: 5, 10}	30S Assembly Intermediate	0	0	0	0
{232: 5, 14}	30S Assembly Intermediate	0	0	0	0
{232: 5, 10, 14}	30S Assembly Intermediate	0	0	0	0
{233: 5}	30S Assembly Intermediate	0	0	0	0
{320}	30S Assembly Intermediate	0	0	0	0

Table B.4 (continued)

		Initial count		Final count	
Species	Type	`cytoplasm`	`nucleoid`	`cytoplasm`	`nucleoid`
{321}	30S Assembly Intermediate	0	0	0	0
{321: 3,9,10,13}	30S Assembly Intermediate	0	0	0	0
{321: 9}	30S Assembly Intermediate	0	0	0	0
{321: 9,13}	30S Assembly Intermediate	0	0	0	0
{321: 9,10,13}	30S Assembly Intermediate	0	0	0	0
{322: 3,10}	30S Assembly Intermediate	0	0	0	0
{323}	30S Assembly Intermediate	0	0	0	0
{330}	30S Assembly Intermediate	0	0	0	0
{331}	30S Assembly Intermediate	0	0	0	0
{331: 13,19}	30S Assembly Intermediate	0	0	0	0
{331: 13}	30S Assembly Intermediate	0	0	0	0
{331: 19}	30S Assembly Intermediate	0	0	0	0
{331: 3,9,10}	30S Assembly Intermediate	0	0	0	0
{331: 3,9,10,13}	30S Assembly Intermediate	0	0	0	0
{331: 9,19}	30S Assembly Intermediate	0	0	0	0
{331: 9,14,19}	30S Assembly Intermediate	0	0	0	0
{331: 9}	30S Assembly Intermediate	0	0	0	0
{331: 9,10}	30S Assembly Intermediate	0	0	0	0
{331: 9,13}	30S Assembly Intermediate	0	0	1	0
{331: 9,10,13}	30S Assembly Intermediate	0	0	0	0
{332}	30S Assembly Intermediate	0	0	0	0
{332: 10}	30S Assembly Intermediate	0	0	0	0
{332: 10,14}	30S Assembly Intermediate	0	0	0	0
{332: 14}	30S Assembly Intermediate	0	0	0	0
{332: 3,10}	30S Assembly Intermediate	0	0	0	0
30S	Small Subunit	1904	577	3952	1198
50S	Large Subunit	1840	557	4964	1515
30S:m$_{alpha}$	Initiation Complex	0	0	0	0
Rib_0^{alpha}	Ribosome	3	0	8	0
Rib_1^{alpha}	Ribosome	4	0	6	2
Rib_2^{alpha}	Ribosome	6	0	10	1
Rib_3^{alpha}	Ribosome	14	0	35	1
$\mathrm{Rib}_{\mathrm{term}}^{alpha}$	Ribosome	31	0	57	3
30S:m$_{rplM}$	Initiation Complex	0	0	0	0
Rib_0^{rplM}	Ribosome	8	0	18	1
Rib_1^{rplM}	Ribosome	32	0	56	6
30S:m$_{rpsF}$	Initiation Complex	0	0	0	0
Rib_0^{rpsF}	Ribosome	4	0	13	0
Rib_1^{rpsF}	Ribosome	6	0	7	1
Rib_2^{rpsF}	Ribosome	5	0	10	0
$\mathrm{Rib}_{\mathrm{term}}^{rpsF}$	Ribosome	35	0	69	8
30S:m$_{rpsJ}$	Initiation Complex	0	0	0	0
Rib_0^{rpsJ}	Ribosome	3	0	3	1
Rib_1^{rpsJ}	Ribosome	27	0	41	5
Rib_2^{rpsJ}	Ribosome	10	0	19	0
Rib_3^{rpsJ}	Ribosome	8	0	21	1
Rib_4^{rpsJ}	Ribosome	31	0	57	3

Table B.4 (continued)

Species	Type	Initial count		Final count	
		`cytoplasm`	`nucleoid`	`cytoplasm`	`nucleoid`
30S:m$_{rpsO}$	Initiation Complex	0	0	0	0
Rib_0^{rpsO}	Ribosome	3	0	5	0
Rib_1^{rpsO}	Ribosome	23	0	40	2
$\text{Rib}_{\text{term}}^{rpsO}$	Ribosome	31	0	60	3
30S:m$_{rpsP}$	Initiation Complex	0	0	0	0
Rib_0^{rpsP}	Ribosome	2	0	2	0
Rib_1^{rpsP}	Ribosome	17	0	30	2
$\text{Rib}_{\text{term}}^{rpsP}$	Ribosome	31	0	51	4
30S:m$_{rpsT}$	Initiation Complex	0	0	0	0
Rib_0^{rpsT}	Ribosome	2	0	5	0
Rib_1^{rpsT}	Ribosome	31	0	60	5
30S:m$_{spc}$	Initiation Complex	0	0	0	0
Rib_0^{spc}	Ribosome	15	0	38	2
Rib_1^{spc}	Ribosome	4	0	8	2
Rib_2^{spc}	Ribosome	14	0	25	3
Rib_3^{spc}	Ribosome	20	0	37	2
$\text{Rib}_{\text{term}}^{spc}$	Ribosome	31	0	60	7
30S:m$_{str}$	Initiation Complex	0	0	0	0
Rib_0^{str}	Ribosome	3	0	8	0
Rib_1^{str}	Ribosome	6	0	14	0
Rib_2^{str}	Ribosome	31	0	49	3
$\text{Rib}_{\text{term}}^{str}$	Ribosome	29	0	59	4

Appendix C

Supporting information for Chapter 4 [*]

C.1 Semi-analytical Modeling

C.1.1 Ribosomal protein operon messenger RNA (mRNA) statistics

We consider the mRNA statistics for the r-protein operons. From Eq. 4.27a, we can write out the chemical master equation (CME) for our system as:

$$
\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}t}P(m,n|t) = k_t(t)\Big[P(m-1,n|t) - P(m,n|t)\Big] \\
+ k_b\Big[(m+1)P(m+1,n-1|t) - mP(m,n|t)\Big] \\
+ k_u\Big[(n+1)P(m-1,n+1|t) - nP(m,n|t)\Big] \\
+ k_d\Big[(m+1)P(m+1,n|t) - mP(m,n|t)\Big],
\end{aligned}
\tag{C.1}
$$

where $k_t(t)$ represents the effective transcription rate as a function of time, with $k_t(t < t_r) = k_t$ and $k_t(t > t_r) = 2k_t$ (where t_r is the gene replication time, itself a function of the timing of chromosome replication and the gene's position on the chromosome). We note that we have implicitly assumed that transcription from both gene copies after duplication is independent and occurs at the same rate, which may not in general be true[215], but simplifies the model considerably. From this we can derive the system of ODEs to describe the time evolution of the mean counts of m and n,

[*] Appendix C contains material reproduced with permission from Earnest TM, Cole JA, Peterson JR, Hallock MJ, Kuhlman TE, and Luthey-Schulten Z (2016). Ribosome biogenesis in replicating cells: integration of experiment and theory. *Biopolymers*, **105**(10), pp. 735–751. doi:10.1002/bip.22892.

their mean squared counts, and the mean product of m and n:

$$\begin{aligned}
\frac{d}{dt}\langle m\rangle(t) &= k_t(t) - k_d\langle m\rangle(t) - k_b\langle m\rangle(t) + k_u\langle n\rangle(t) \\
\frac{d}{dt}\langle n\rangle(t) &= k_b\langle m\rangle(t) - k_u\langle n\rangle(t) \\
\frac{d}{dt}\langle m^2\rangle(t) &= 2k_t(t)\langle m\rangle(t) + k_t(t) - 2k_b\langle m^2\rangle(t) + k_b\langle m\rangle(t) \\
&\quad + 2k_u\langle mn\rangle(t) + k_u\langle n\rangle(t) - 2k_d\langle m^2\rangle(t) + k_d\langle m\rangle(t) \\
\frac{d}{dt}\langle n^2\rangle(t) &= 2k_b\langle mn\rangle(t) + k_b\langle m\rangle(t) - 2k_u\langle n^2\rangle(t) + k_u\langle n\rangle(t) \\
\frac{d}{dt}\langle mn\rangle(t) &= k_t(t)\langle n\rangle(t) - k_b\langle mn\rangle(t) + k_b\langle m^2\rangle(t) - k_b\langle m\rangle(t) \\
&\quad + k_u\langle n^2\rangle(t) - k_u\langle mn\rangle(t) - k_u\langle n\rangle(t) - k_d\langle mn\rangle(t).
\end{aligned} \tag{C.2}$$

We expect that at cell division all components are distributed to the daughter cells according to an unbiased binomial distribution. This can be used to derive constraints for our system of ODEs, namely:

$$\begin{aligned}
\langle m\rangle(0) &= \frac{1}{2}\langle m\rangle(t_D) \\
\langle n\rangle(0) &= \frac{1}{2}\langle n\rangle(t_D) \\
\langle m^2\rangle(0) &= \frac{1}{4}\Big[\langle m\rangle(t_D) + \langle m^2\rangle(t_D)\Big] \\
\langle n^2\rangle(0) &= \frac{1}{4}\Big[\langle n\rangle(t_D) + \langle n^2\rangle(t_D)\Big] \\
\langle mn\rangle(0) &= \frac{1}{4}\langle mn\rangle(t_D).
\end{aligned} \tag{C.3}$$

This system can be solved numerically, but parameters must be chosen carefully. Specifically we are concerned with the ribosome binding and unbinding rates. The binding rate, k_b, is clearly a function of the concentration of free ribosomes as well as other mRNA in the cell; as a first approximation, we might expect:

$$k_b \approx k_{b,0}[r_{\text{free}}] = k_{b,0}[r - C_r - n], \tag{C.4}$$

where $k_{b,0}$ represents the binding rate of a single messenger to a single ribosome, r represents the ribosome copy number in the cell, C_r represents the number of competing mRNA that are bound to ribosomes, n is the number of ribosome-bound versions of the messenger we are interested in, and square brackets e.g. $[x] \approx x\cdot 2^{-t/t_D}/2\ln(2)$ denotes a per-cell concentration. We might assume that the competing mRNA are in equilibrium with respect the ribosomes, meaning:

$$k_u[C_r] = k_{b,0}[C - C_r][r - C_r - n] \tag{C.5}$$

where C represents the total number of competing mRNA. Solving this for C_r and inserting the result into Eq. C.4 then yields:

$$k_b \approx k_{b,0}\Big([r]-[n]-\frac{1}{2}\Big(-\sqrt{(\frac{k_u}{k_{b,0}}-[n]+[r]+[C])^2+4([n][C]-[r][C])}+\frac{k_u}{k_{b,0}}-[n]+[r]+[C]\Big)\Big). \tag{C.6}$$

Inserting Eq. C.6, a value for k_u chosen such that the ribosome-bound messengers have an appropriate mean lifetime, the mean value of $[r] = 3000$[33], and $C = c(m+n)$ (where c denotes the number of competing genes, assuming that the competing mRNA production roughly keeps pace with that of the messenger we are interested in) into Eq. C.2 and solving the system numerically (using the NDSolve function in Mathematica) yields traces for the mRNA statistics over the cell cycle. We can then perform the appropriate time-averaging over the cell cycle (see Peterson et al. [1] for details) in order to calculate the mean and variance of our mRNA:

$$\begin{aligned}
\mathrm{E}[m] &= \int_0^{t_D} \frac{2\ln(2)}{t_D} 2^{-t/t_D} \langle m\rangle(t)\,\mathrm{d}t \\
\mathrm{E}[n] &= \int_0^{t_D} \frac{2\ln(2)}{t_D} 2^{-t/t_D} \langle n\rangle(t)\,\mathrm{d}t \\
\mathrm{Var}[m] &= \int_0^{t_D} \frac{2\ln(2)}{t_D} 2^{-t/t_D} \langle m^2\rangle(t)\,\mathrm{d}t - \mathrm{E}[m]^2 \\
\mathrm{Var}[n] &= \int_0^{t_D} \frac{2\ln(2)}{t_D} 2^{-t/t_D} \langle n^2\rangle(t)\,\mathrm{d}t - \mathrm{E}[n]^2 \\
\mathrm{Cov}[m,n] &= \int_0^{t_D} \frac{2\ln(2)}{t_D} 2^{-t/t_D} \langle mn\rangle(t)\,\mathrm{d}t - \mathrm{E}[m]\,\mathrm{E}[n].
\end{aligned} \tag{C.7}$$

From these we can compute the statistics of our total mRNA count:

$$\begin{aligned}
\mathrm{E}[m+n] &= \mathrm{E}[m] + \mathrm{E}[n] \\
\mathrm{Var}[m+n] &= \mathrm{Var}[m] + \mathrm{Var}[n] + 2\,\mathrm{Cov}[m,n].
\end{aligned} \tag{C.8}$$

C.1.2 Estimating Rate Parameters for an “Average mRNA”

Because the nine r-protein operons have varying rates of production, translation, and degradation, we attempted, for the sake of simplicity, to investigate the behavior of an “average mRNA”. We first computed the harmonic mean of the operons’ transcription and degradation rates (yielding 0.0042 s^{-1} and $9.84 \times 10^{-4}\ \mathrm{s}^{-1}$, respectively). Then, in order to estimate the ribosome unbinding rate for each operon we computed the mean lifetime of each mRNA-ribosome complex. Each operon has a different number of genes to be translated, each of which in turn has a different translation rate, meaning that each operon will be bound to a ribosome for a different amount of time. We can compute the mean mRNA-ribosome complex lifetime for each operon, and from that determine each operon’s effective unbinding rate:

$$k_{u,i} = \Big(\frac{1}{k^{-1}_{\text{su, dissoc}} + \sum_j k^{-1}_{t1,i,j}}\Big)^{-1} \tag{C.9}$$

where $k_{u,i}$ represents the unbinding rate for the i^{th} operon's messengers, $k_{\text{su, dissoc}}$ represents the rate at which a translated ribosome dissociates from the messenger, and $k_{tl,i,j}$ represents the translation rate of the j^{th} gene in operon i. The results of these computations are summarized in Table 4.5.

C.1.3 mRNA statistics in the Limit when $k_d \to 0$

Peterson et al.[1] derived expressions for mRNA statistics that accounted for gene duplication due to chromosome replication; specifically, Equation S36 is given as:

$$\begin{aligned}
\mathrm{E}[r] &= \frac{k_t}{k_d} 2^f \left[1 + \beta \frac{\mathrm{e}^{-k_d t_D (1-f)} - 2^{1-f}}{1 + \frac{k_d t_D}{\ln(2)}} + \gamma \frac{2^{-f} \mathrm{e}^{-k_d t_D f} - 1}{1 + \frac{k_d t_D}{\ln(2)}} \right] \\
\mathrm{Var}[r] &= \mathrm{E}[r] - \mathrm{E}[r]^2 \\
&\quad + \ln(2) \left(\frac{k_t}{k_d} \right)^2 \left[2\beta^2 \frac{1 - 2^{f-1} \mathrm{e}^{-2k_d t_D (1-f)}}{\ln(2) + 2k_d t_D} - 4\beta \frac{1 - 2^{f-1} \mathrm{e}^{-k_d t_D (1-f)}}{\ln(2) + k_d t_D} + \frac{2}{\ln(2)} (1 - 2^{f-1}) \right. \\
&\quad \left. + \gamma^2 \frac{2^f - \mathrm{e}^{-2k_d t_D f}}{\ln(2) + 2k_d t_D} - 4\gamma \frac{2^f - \mathrm{e}^{-k_d t_D f}}{\ln(2) + k_d t_D} - \frac{4}{\ln(2)} (1 - 2^f) \right],
\end{aligned} \tag{C.10}$$

where

$$\begin{aligned}
\beta &= \frac{\mathrm{e}^{-k_d t_D f}}{2 - \mathrm{e}^{-k_d t_D}} \\
\gamma &= \left(1 + \frac{\mathrm{e}^{-k_d t_D}}{2 - \mathrm{e}^{-k_d t_D}} \right).
\end{aligned} \tag{C.11}$$

In these equations, k_t and k_d are the RNA transcription and degradation rates, respectively, t_D is the cell doubling time, and f represents the fraction of the cell cycle after gene replication ($f = 1 - t_r / t_D$). For the purposes of the present work, we note that the nucleation and assembly of the ribosome occurs significantly faster than measured rates of mRNA degradation; as a result, we expect little ribosomal RNA (rRNA) to be lost, and essentially all of it to be found in the form of ribosomes in the cell. Similarly, because the mRNA-ribosome dissociation constant is small ($\sim 10^{-10}$ M), when the pool of ribosomes is large compared to the pool of available messengers, essentially all mRNA will remain bound to ribosomes and few will be degraded. We therefore consider the limit of the expressions in Eq. C.10 as k_d approaches zero:

$$\begin{aligned}
\lim_{k_d \to 0} \mathrm{E}[r] &= \frac{k_t t_D 2^f}{\ln(2)} \\
\lim_{k_d \to 0} \mathrm{Var}[r] &= \frac{k_t t_D}{\ln^2(2)} \Big[2^f \Big(\ln(2) + 2k_t t_D (3 + \ln(4)) \Big) - 4^f k_t t_D \\
&\quad - k_t t_D \Big(4 + 2(1+f)^2 \ln^2(2) + (1+f) \ln(16) \Big) \Big].
\end{aligned} \tag{C.12}$$

C.2 Estimating Cell Cycle Parameters from Copy Number Distributions

We consider the well known age distribution of exponentially growing cells[178]:

$$\phi(a) = 2\nu_m e^{-\nu_m a} \int_a^\infty f(\tau)\, \mathrm{d}\tau, \tag{C.13}$$

where $\phi(a)$ is the probability that a cell is of age a, ν_m is the growth rate of the population, and $f(\tau)$ is the probability of a cell dividing at age τ. As per Powell[178], ν_m can be determined from the constraint

$$2\int_0^\infty e^{-\nu_m \tau} f(\tau)\, \mathrm{d}\tau = 1. \tag{C.14}$$

Taylor expanding the LHS of Eq. C.14 about the mean division time:

$$\begin{aligned} 1 &= 2\int_0^\infty e^{-\nu_m \tau} f(\tau)\, \mathrm{d}\tau \\ &= 2\langle e^{-\nu_m \tau}\rangle \\ &\approx 2\Big[e^{-\nu_m \langle\tau\rangle} + \frac{1}{2}\frac{\mathrm{d}^2}{\mathrm{d}\tau^2} e^{-\nu_m \tau}|_{\langle\tau\rangle}\sigma_\tau^2\Big] \\ &= 2\Big[e^{-\nu_m t_D} + \frac{1}{2}\nu_m^2 e^{-\nu_m t_D}\sigma_\tau^2\Big] \\ &= (2+\nu_m^2\sigma_\tau^2)e^{-\nu_m t_D} \end{aligned} \tag{C.15}$$

where we assume that the mean age at division is t_D, and the division ages have some variance σ_τ^2. This can then be easily solved numerically for ν_m.

We now consider the probability that a cell has a single copy of a given gene. If t_r is the age at which the gene is replicated, we can write:

$$P_{\text{single copy}} = \int_0^{t_r} \phi(a)\, \mathrm{d}a. \tag{C.16}$$

For simplicity, we can assume the division times are normally distributed,

$$f(\tau) = N(\tau; t_D, \sigma_\tau). \tag{C.17}$$

and so,

$$\begin{aligned} P_{\text{single copy}}(t_r) &= \int_0^{t_r} 2\nu_m e^{-\nu_m a} \int_a^\infty f(\tau)\, \mathrm{d}\tau\, \mathrm{d}a \\ &= \int_0^{t_r} 2\nu_m e^{-\nu_m a} \frac{1}{2}\operatorname{erfc}\Big(\frac{a - t_D}{\sqrt{2}\sigma_\tau}\Big)\, \mathrm{d}a. \end{aligned} \tag{C.18}$$

Promoting t_r to a random variable distributed according to some probability function $P(t_r; \langle t_r\rangle, \sigma_{t_r})$ where $\langle t_r\rangle$ and σ_{t_r} are the mean and standard deviation of the replication time, respectively, we can write

$$\langle P_{\text{single copy}}\rangle = \int_0^\infty \mathrm{d}t_r P_{\text{single copy}}(t_r) P(t_r; \langle t_r\rangle, \sigma_{t_r}). \tag{C.19}$$

Now simply Taylor expanding about $\langle t_r \rangle$ yields

$$
\begin{aligned}
\langle P_{\text{single copy}}\rangle(\langle t_r\rangle, \sigma_{t_r}) \approx \int_0^{\langle t_r\rangle} & 2\nu_m e^{-\nu_m a}\frac{1}{2}\operatorname{erfc}\Big(\frac{a-t_D}{\sqrt{2}\sigma_\tau}\Big)\mathrm{d}a \\
&+ \frac{\sigma_{t_r}^2}{2}\Big[-2\nu_m^2 e^{-\nu_m\langle t_r\rangle}\frac{1}{2}\operatorname{erfc}\Big(\frac{\langle t_r\rangle - t_D}{\sqrt{2}\sigma_\tau}\Big) - 2\nu_m e^{-\nu_m\langle t_r\rangle} f(\langle t_r\rangle)\Big],
\end{aligned}
\tag{C.20}
$$

subject to Eq. C.14. Now, for each gene locus, we can estimate the fraction of cells we expect to see with a single gene copy. Inserting forms for the mean replication time, $\langle t_{r,i}\rangle = \mu_{\text{trep}} + \chi_i T_{\text{rep}}$, for a gene given its location, χ_i on the chromosome, as well as the standard deviation in the replication timing, $\sigma_{t_r} = \sigma_{\text{trep}}$ into Eq. C.20, we can construct a measure for for goodness of fit:

$$
\Theta = \sum_{i \in \text{genes}} \left(\frac{\langle P_{\text{single copy}}\rangle(\langle t_{r,i}\rangle, \sigma_{t_r}) - \frac{n_i}{m_i}}{\sqrt{\langle P_{\text{single copy}}\rangle(\langle t_{r,i}\rangle, \sigma_{t_r})\big(1 - \langle P_{\text{single copy}}\rangle(\langle t_{r,i}\rangle, \sigma_{t_r})\big)/m_i}} \right)^2 m
\tag{C.21}
$$

and vary μ_{trep}, T_{rep}, and σ_{trep} in order to minimize it. Importantly, here n_i denotes the number of cells (with gene i labeled) observed with a single copy, m_i denotes the total number of cells (with gene i labeled) observed, and the term $\sqrt{\langle P_{\text{single copy}}\rangle(\langle t_{r,i}\rangle, \sigma_{t_r})\big(1 - \langle P_{\text{single copy}}\rangle(\langle t_{r,i}\rangle, \sigma_{t_r})\big)/m_i}$ denotes an estimate for the error in the experimentally observed fraction of cells with one gene copy. This estimate is based in the assumption that a cell has probability $\langle P_{\text{single copy}}\rangle(\langle t_{r,i}\rangle, \sigma_{t_r})$ of being in a one-copy state, and that each measured cell represents an independent Bernoulli trial. This error estimate was introduced in order to give greater weight during fitting to the genes for which we have greater numbers of experimental images.

We assumed a value for σ_τ of 12 minutes, and, because Θ was found to change little with variations in σ_{t_r}, we initially required $20.2 < \sigma_{t_r} < 24.0$ (such that its value stays within the error bounds found by the more complete fitting method presented in the main manuscript, see Table 4.1). Θ was then minimized using the Minimize routine in Mathematica. This resulted in estimates for μ_{trep} and T_{rep} of 34.4 and 45.9 minutes, respectively. We note this value for T_{rep} is well within the standard error reported in Table 4.1 but the value for μ_{trep} differs from the results of the main text by approximately 2.6 standard deviations. Comparison of the fit and experimental single-gene fractions shows similar qualitative agreement as was obtained using the method presented in the main text (see Figure C.6). For the sake of comparison, releasing the bounds on σ_{t_r} had only a minor effect on Θ, μ_{trep}, and T_{rep} (their values changed by about 0.05%, 3%, and 0.2%, respectively), but the fit value of σ_{t_r} fell to an unreasonably low value of approximately 0.25 seconds.

C.3 Fitting mRNA Distributions

The exact analytical theory set out in Peterson et al. [1] describes the noise of idealized constitutively expressed genes which undergo duplication during the cell cycle[1]. As no experimental data is available to compare to the distributions of messengers computed in the ribosome biogenesis model (RBM) at the time of writing, the results were compared to this theory. Due to the fact that the ribosomal protein operon messengers are the only transcripts competing for the ribosomes, they are bound up and effectively prevented from degradation. In the future, all cellular transcripts will be considered, thus the RBM will need to be reparameterized. By fitting the theory of Peterson et al.[1] to the RBM simulations, we can estimate what the new parameters would need to be to give the same results as the RBM.

Distributions computed using Eq. S22 of Peterson et al.[1] were computed with varying k_t and k_d and compared to the RBM simulated distributions. The mean squared deviation was computed between these simulations. The k_t and k_d associated with the distribution that has the minimum deviation from the simulated distribution represent the "effective" transcription and degradation rates that will be applicable in future simulations that include realistic counts of competing mRNAs. The fitted rates and fits can be seen in Table 4.5 and Figure 4.6. Fitted k_t and k_d are essentially scaled versions of the rates used in the RBM (as demonstrated in Figure C.9). Fit k_d are about four times smaller than experimental values, while k_t are about four times as large.

C.4 Varying Numbers of Non-ribosomal Genes in the SAM

We used the SAM to study the simultaneous effect of varying gene loci (which effects the timing of gene replication) and c on the mRNA copy number statistics for an "average gene" (see Section C.1.2). Interestingly, we found that with increasing numbers of actively expressed genes (while holding k_t and k_d constant), gene expression became significantly less noisy (see Figure C.10b). This was found to be due largely to the fact that the messenger-ribosome dissociation constant was very small ($k_u/k_{\mathrm{mrna\_assoc}} \approx 10^{-10}$ M). By comparison, the concentration of a single mRNA in a bacterial cell of volume ~1 fl is approximately 1.7×10^{-9} M. This means that every messenger produced should have a high probability of being bound to a ribosome, provided a ribosome is available for binding. When small numbers of non-ribosomal genes are expressed (the small-c regime), the total number of messengers does not exceed the total number of ribosomes, and so every messenger is likely to be quickly bound and thereafter protected from degradation. The statistics of a given gene's mRNA in this regime are then essentially the same as those of a model in which *already bound* messengers are produced (at some transcription rate k_t) and only lost through cell division (with roughly half going to each daughter). The model of Peterson et al.[1] in the limit where $k_d \to 0$ (see Figure C.10b, left-most dots, and see Eq. C.12) gives exactly these statistics. At values of c between 30 and 50, the total mRNA content of the cell approaches and then surpasses the total number of ribosomes. From there on, with increasing values of c, the probability of a given messenger binding a ribosome becomes increasingly small. For any specific gene of interest, this results in an increase in the fraction of unbound mRNAs (relative to ribosome-bound mRNAs), and in turn an increase in the messenger's effective degradation rate, a decrease in its mean copy number, and a decrease in its Fano factor. In the limit where $c \to \infty$, the probability that any specific messenger is ever bound by a ribosome approaches zero; the total messenger count is then dominated by the unbound messengers and the statistics converge again to those of a model in which ribosome interactions are entirely neglected, e.g. that of Peterson et al.[1] (see Figure C.10b right-most dots). These findings are in good agreement with additional explicit stochastic simulations (see Figure C.10b, diamonds, and see Section 4.4.3). We note that the SAM described here remains somewhat incomplete. It neglects, for example, the potential for transient non-specific mRNA-ribosome interactions[216] which have been shown to significantly impact ribosome diffusivity. Such interactions have been estimated to last on the order of a few seconds and may play a role in ribosomal LSU–SSU subunit search and association[216]. Assuming that the transiently bound mRNA are, like the specifically bound mRNA, protected from degradation, then this type of interactions should have the net effect of lowering the free messenger counts, and in turn, lowering the effective degradation rate. When c in the SAM is small (e.g. $c = 8$), the effective degradation rate is already approximately zero, and so non-specific ribosome binding can not significantly affect the mRNA statistics. When c is in the biologically realistic regime ($c \approx 1000$), an upper-bound for the possible changes in the mRNA statistics can be roughly estimated by considering the effect of increasing the ribosome concentration in the SAM. For example, if we assume that only one messenger can transiently bind a ribosome at a given time, and that the transient mRNA-ribosome association rate is fast (e.g. occurring at a diffusion-limited rate on the order of 10^9 M^{-1} s^{-1}), then we can expect that the available non-specific binding sites should essentially always be occupied, and the mRNA statistics can be approximated simply by doubling the ribosome concentration in the SAM. For a gene situated on the chromosome halfway between the origin and terminus, this has the effect of increasing mean messenger count from 8.6 to 11.3 per cell, and only modestly changes the Fano factor from 1.6 to 1.7.

C.5 Algorithms used in the RBM

Algorithm C.1 details the process of pruning the intermediate species graph. This method is used by Earnest et al.[94] to reduce the approximately 1600 potential intermediates down to a number of species that can be simulated with reaction–diffusion master equation (RDME).

The RDME trajectory generation process is detailed in Algorithm C.2. The cell growth process is implemented using pyLM's "hybrid solver" interface[38] which allows for user-defined processes to occur during regular intervals when simulation data is saved. This is when cell geometry is updated to account for growth, and to add new DNA operons to the simulation to reflect the replication process.

The process for building the diving cell geometry is shown in Algorithm C.3. Algorithm C.4 is the process of discerning which discrete lattice sites fall inside a spherocylinder described via two points at the cylinder ends and a radius. Algorithm C.5 detects boundaries between different site types and changes those sites to a third type. It is used to construct the membrane between the cytoplasm and extracellular space.

```
Data: graph $G = (V, E)$ with intermediate species as vertices and directed edges indicating
      reactions, maximum species in final graph $N_{\max}$
Result: $G' = (V', E')$ representing pruned network
$V' \longleftarrow V$ ;
$E' \longleftarrow E$ ;
repeat
    Compute fluxes through $G'$ ;
    $D \longleftarrow$ vertex in $G'$ with lowest flux ;
    while $|D| > 0$ do
        for $v \in D$ do
            for $e \in E'$ do // Remove edges to and from v
                if $v \in e$ then
                    $E' \longleftarrow E' - e$ ;
            $V' \longleftarrow V' - v$ // Remove v;
        $D \longleftarrow \emptyset$;
        // Locate dead-end vertices
        for $v \in V'$ do
            if $\deg^{+}(v) = 0$ or $\deg^{-}(v) = 0$ then
                $D \longleftarrow D + v$;
until $\left|V'\right| < N_{\max}$;
```

Algorithm C.1 SSU assembly network pruning

```
Data: Initial particle lattice – $\boldsymbol{x}$, stoichiometric matrix – $\mathbf{S}$, propensity functions – $a_r(\boldsymbol{n})$,
      time step – $\tau$, cell doubling time – $t_{\text{div}}$, lattice write interval – $t_{\text{w}}$, cell geometry – $\mathscr{G}$,
      and maximum evaluation time – $t_{\text{f}}$.
Result: Final particle lattice – $\boldsymbol{x}$, final site lattice – $\boldsymbol{s}$

$t \longleftarrow 0$;
$t_{\text{last}} \longleftarrow 0$;
while $t < t_{\text{end}}$ do
    for $\xi \in \{x, y, z\}$ do
        parfor $v \in V$ do                                    // Execute on GPU
            diffusionKernel($\xi, v$);
    parfor $v \in V$ do                                        // Execute on GPU
        reactionKernel($v$);
    $t \longleftarrow t + \tau$;
    if $t - t_{\text{last}} > t_{\text{w}}$ then
        computeGeometry($t / t_{\text{div}}, \mathscr{G}$);
        $t_{\text{last}} \longleftarrow t$;
```

Algorithm C.2 RDME simulation with replication and cell growth

```
// Write the site type lattice for a growing cell of geometry $\mathscr{G}$, at a
   fraction $\chi$ through the growth cycle, into the lattice $\boldsymbol{s}$.
$(\ell, w, \phi_{\text{nuc.w.}}, \phi_{\text{nuc.l.}}, \boldsymbol{r}_0) \longleftarrow \mathscr{G}$;
$r \longleftarrow w/2$                    // Sphere/cylinder radius;
$h \longleftarrow (\ell - w)/2$                  // $\frac{1}{2}$ middle cylinder height;
$\chi' \longleftarrow \max\{1, \chi\}$;
$\Delta\ell \longleftarrow \ell \cdot (2^{\chi'} - 1) + \chi'$        // $+\chi'$ to ensure 1 site separation between cells
// Calculate the capsule foci for the mother and daughter cells
$\boldsymbol{u}_{\text{m}} \longleftarrow \boldsymbol{r}_0 + (-h - \Delta\ell)\,\hat{\boldsymbol{z}}$;
$\boldsymbol{v}_{\text{m}} \longleftarrow \boldsymbol{r}_0 + (+h - \Delta\ell)\,\hat{\boldsymbol{z}}$;
$\boldsymbol{u}_{\text{d}} \longleftarrow \boldsymbol{r}_0 + (-h + \Delta\ell)\,\hat{\boldsymbol{z}}$;
$\boldsymbol{v}_{\text{d}} \longleftarrow \boldsymbol{r}_0 + (+h + \Delta\ell)\,\hat{\boldsymbol{z}}$;
for $s \in \boldsymbol{s}$ do                  // Mark every site as extracellular initially
    $s \longleftarrow$ EXTRACELLULAR;
buildCapsule($\boldsymbol{u}_{\text{m}}, \boldsymbol{v}_{\text{m}}, \boldsymbol{s}, r,$ CYTOPLASM);
buildCapsule($\boldsymbol{u}_{\text{d}}, \boldsymbol{v}_{\text{d}}, \boldsymbol{s}, r,$ CYTOPLASM);
buildMembrane($\boldsymbol{s},$ MEMBRANE, CYTOPLASM, EXTRACELLULAR);
// Nucleoid width and length are scaled by $\phi_{\text{nuc.w.}}$ and $\phi_{\text{nuc.l.}}$,
   respectively.
$r_{\text{n}} \longleftarrow r\phi_{\text{nuc.w.}}$                 // Nucleoid sphere/cylinder radius;
$r_{\text{h}} \longleftarrow h - (1 - \phi_{\text{nuc.l.}}) \cdot \ell/2$         // Nucleoid middle cylinder $\frac{1}{2}$ height;
// Calculate the capsule foci for the mother and daughter nucleoid
   regions
$\boldsymbol{u}_{\text{mn}} \longleftarrow \boldsymbol{r}_0 + (-h_{\text{n}} - \Delta\ell)\,\hat{\boldsymbol{z}}$;
$\boldsymbol{v}_{\text{mn}} \longleftarrow \boldsymbol{r}_0 + (+h_{\text{n}} - \Delta\ell)\,\hat{\boldsymbol{z}}$;
$\boldsymbol{u}_{\text{dn}} \longleftarrow \boldsymbol{r}_0 + (-h_{\text{n}} + \Delta\ell)\,\hat{\boldsymbol{z}}$;
$\boldsymbol{v}_{\text{dn}} \longleftarrow \boldsymbol{r}_0 + (+h_{\text{n}} + \Delta\ell)\,\hat{\boldsymbol{z}}$;
buildCapsule($\boldsymbol{u}_{\text{mn}}, \boldsymbol{v}_{\text{mn}}, \boldsymbol{s}, r_{\text{n}},$ NUCLEOID);
buildCapsule($\boldsymbol{u}_{\text{dn}}, \boldsymbol{v}_{\text{dn}}, \boldsymbol{s}, r_{\text{n}},$ NUCLEOID);
```

Function C.3 computeGeometry($\chi, \mathscr{G}, \boldsymbol{s}$)

```
// Construct capsule by building two spheres of radius $r$, centered at
   points $\boldsymbol{u}$ and $\boldsymbol{v}$, and a cylinder of radius $r$ from $\boldsymbol{u}$ to $\boldsymbol{v}$
for $s \in \boldsymbol{s}$ do
    $\boldsymbol{w} \longleftarrow$ getSiteCoord($s$);
    $\boldsymbol{d}_{\mathrm{wu}} \longleftarrow \boldsymbol{w} - \boldsymbol{u}$;
    $\boldsymbol{d}_{\mathrm{wv}} \longleftarrow \boldsymbol{w} - \boldsymbol{v}$;
    if $\|\boldsymbol{d}_{\mathrm{wu}}\| < r$ or $\|\boldsymbol{d}_{\mathrm{wv}}\| < r$ or $(\|\boldsymbol{d}_{\mathrm{wu}} - (\boldsymbol{d}_{\mathrm{wu}} \cdot \hat{\boldsymbol{z}})\hat{\boldsymbol{z}}\| < r$ and $0 < \boldsymbol{d}_{\mathrm{wu}} \cdot \hat{\boldsymbol{z}} < \|\boldsymbol{v} - \boldsymbol{u}\|)$ then
        setSiteType($s$,$T$);
```

Function C.4 buildCapsule($\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{s}, r, T$)

```
// Locate boundary sites of an inside type that are touching a site
   marked as an outside type.  Convert these boundary sites to inside
   types.
for $s \in \boldsymbol{s}$ do
    if getSiteType($s$) = $T_{\mathrm{in}}$ then
        if $\exists s' \in$ nearestNeighborSites($s$) : getSiteType($s'$) = $T_{\mathrm{out}}$ then
            setSiteType($s$,$T_{\mathrm{mem}}$)
```

Function C.5 buildMembrane($\boldsymbol{s}, T_{\mathrm{mem}}, T_{\mathrm{in}}, T_{\mathrm{out}}$)

C.6 Supplementary figures

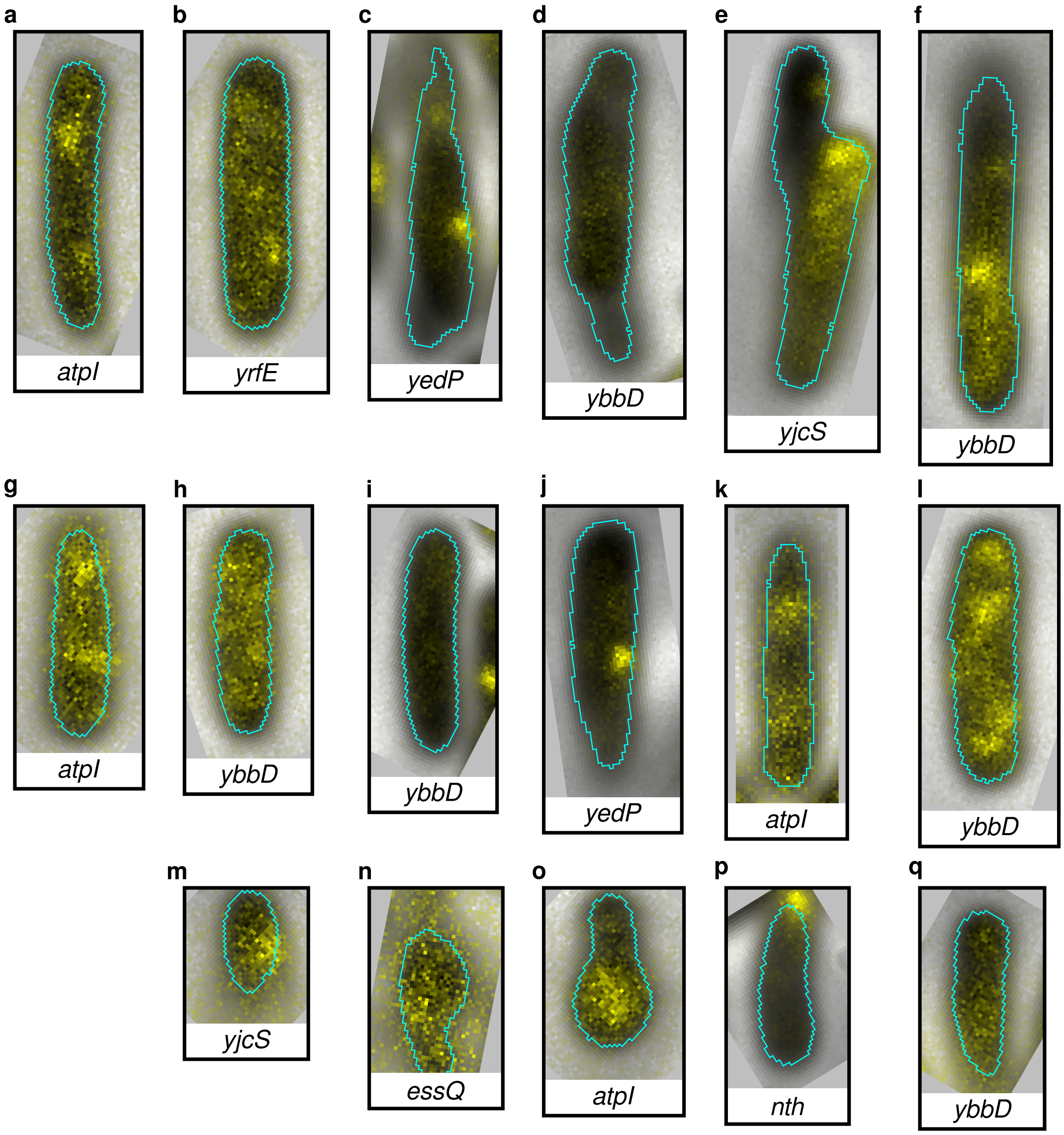


Figure C.1 Examples of rejected regions from fluorescence microscopy. (**f**, **j**, **q**) Cell partially out of focal plane. (**d**, **i**, **p**) No fluorescence. (**a**, **b**, **e**, **f**, **g**, **h**, **k**, **l**, **m**) Poor dynamic range in fluorescence channel. (**c**, **d**, **e**, **m**, **n**, **o**) Bad morphology of thresholded brightfield image. (**n**) Debris on surface.

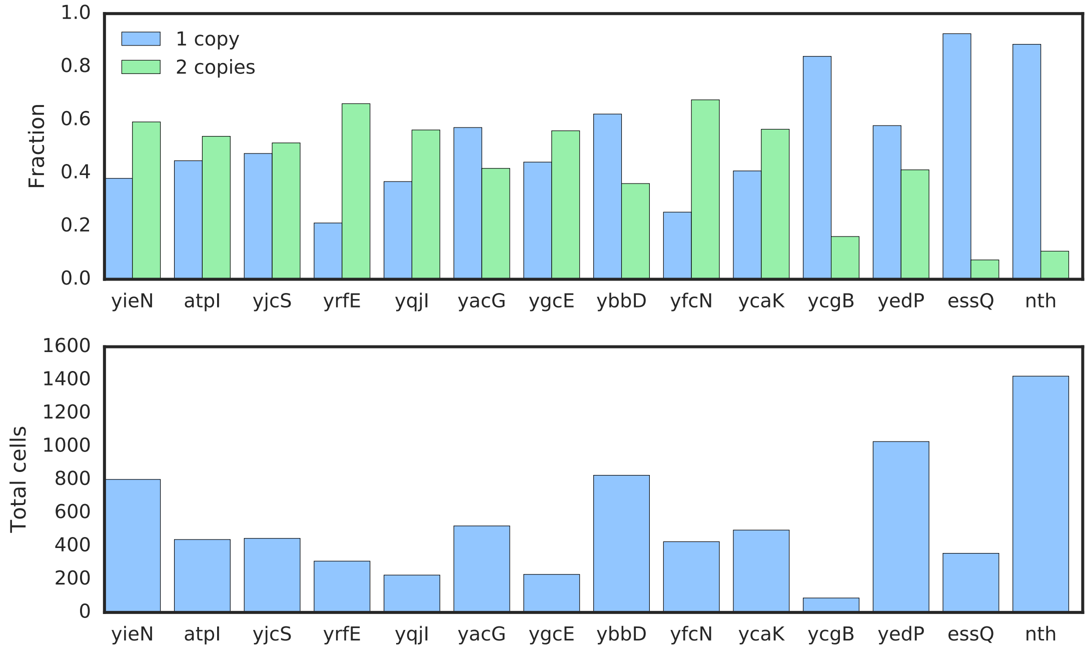


Figure C.2 Fraction of cells observed with one or two operon copies and total count of cells with one or two copies.

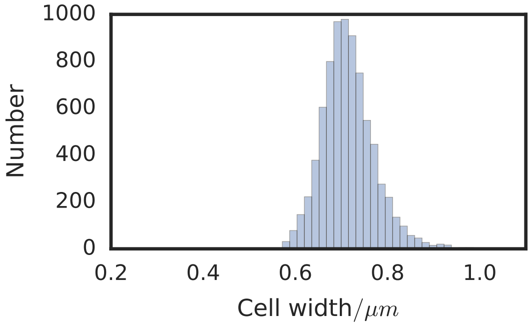


Figure C.3 Distribution of cell widths. The cells were measured to have a mean of 0.715 µm and standard deviation of 0.059 µm.

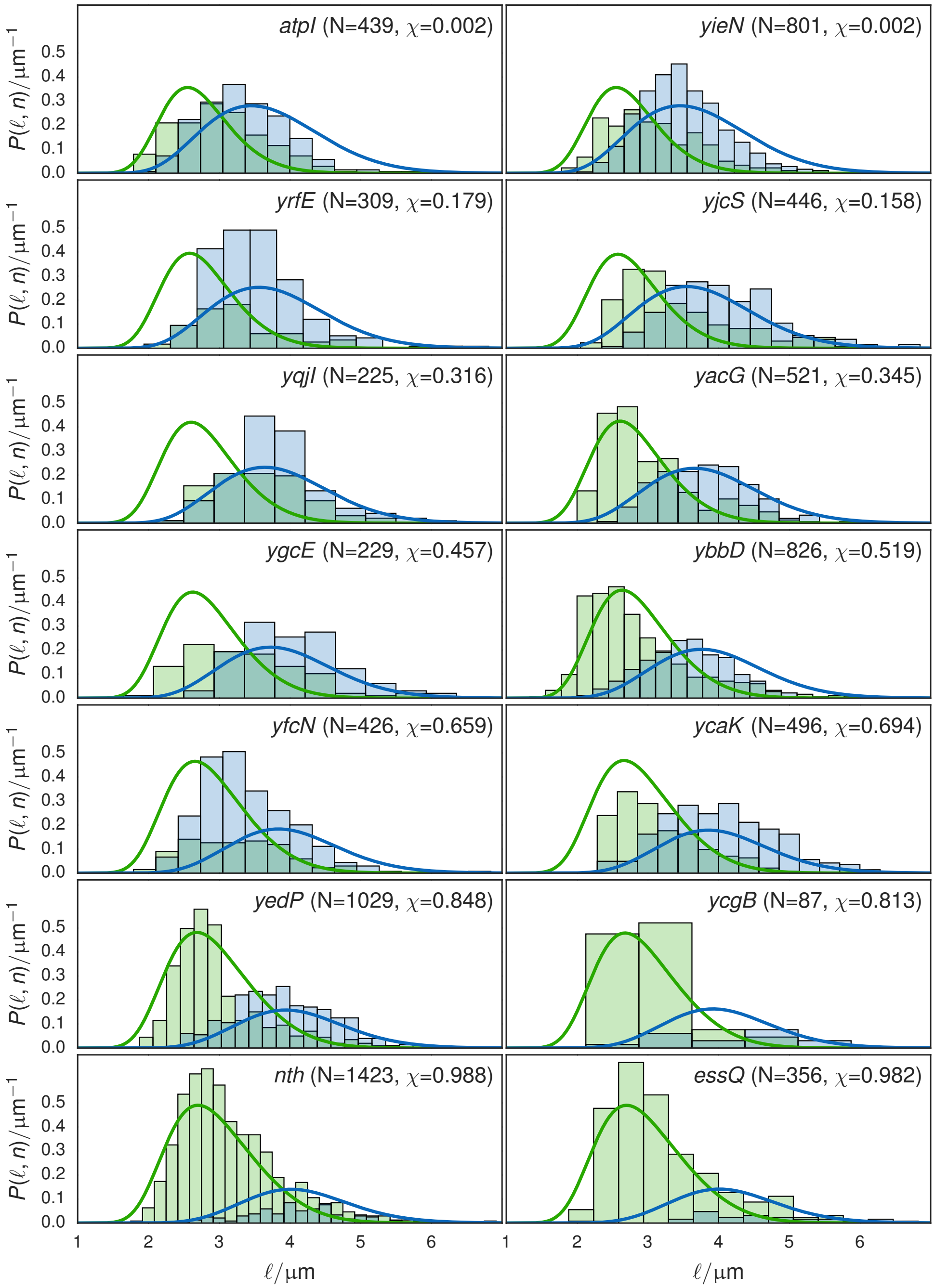


Figure C.4 Comparison of experimental length distributions to fitting of growth/replication model. The number of cells binned and fractional position of the gene along its replichore are specified in each subplot as N and χ respectively. Blue denotes cells with two copies, green with one copy.

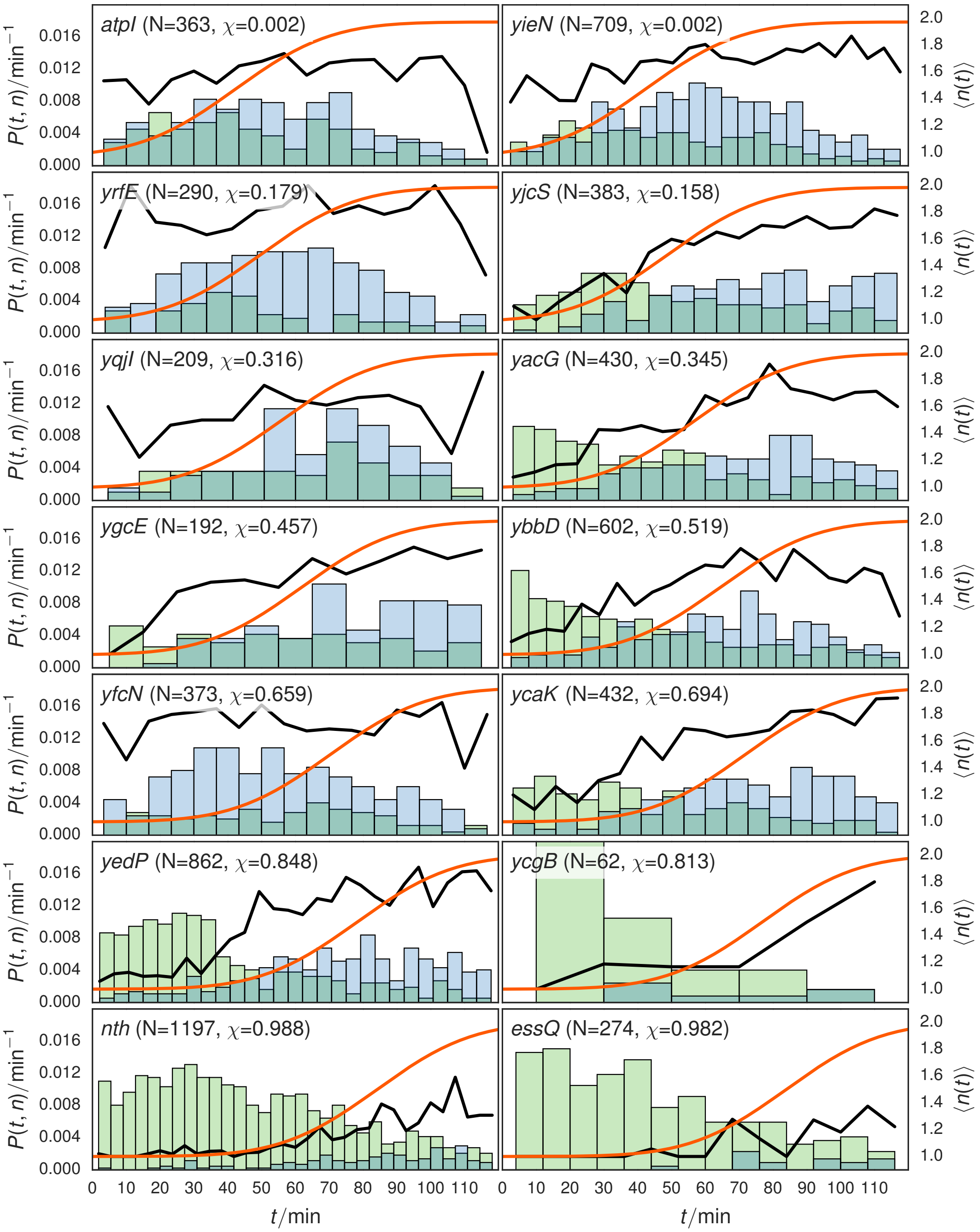


Figure C.5 Comparison of inferred cell age distributions to fitting of growth/replication model. The number of cells binned and fractional position of the gene along its replichore are specified in each subplot as N and χ respectively. Blue denotes cells with two copies, green with one copy, the orange and black curves are the predicted and observed average copy numbers respectively.

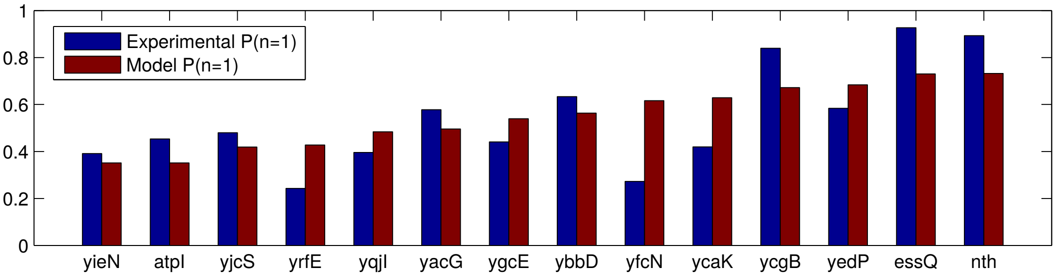


Figure C.6 Fractions of cells with a single gene copy. Blue bars indicate the fraction of experimental cells observed with a single copy, while red bars indicate the fraction predicted by minimizing Θ in Eq. C.21 (while requiring $20.2 < \sigma_{t_r} < 24.0$).

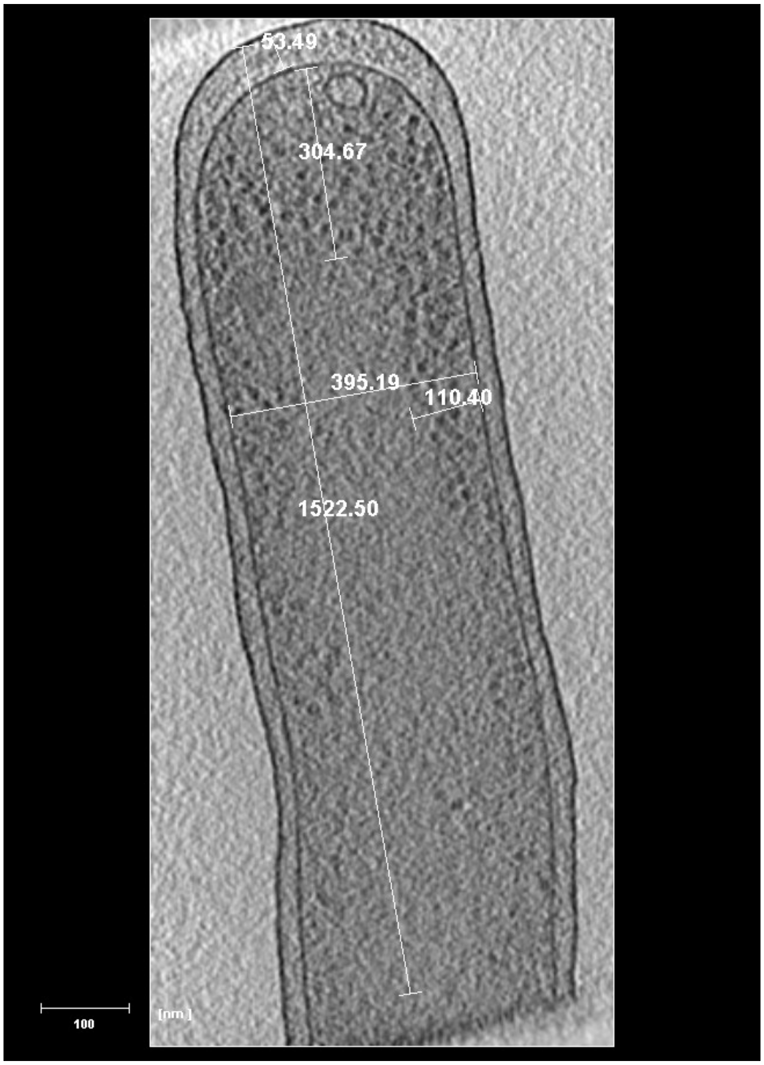


Figure C.7 Cryo-electron tomogram of slow-growing *E. coli*[33] used to measure the nucleoid geometry for the whole-cell simulations. Units are in nanometers.

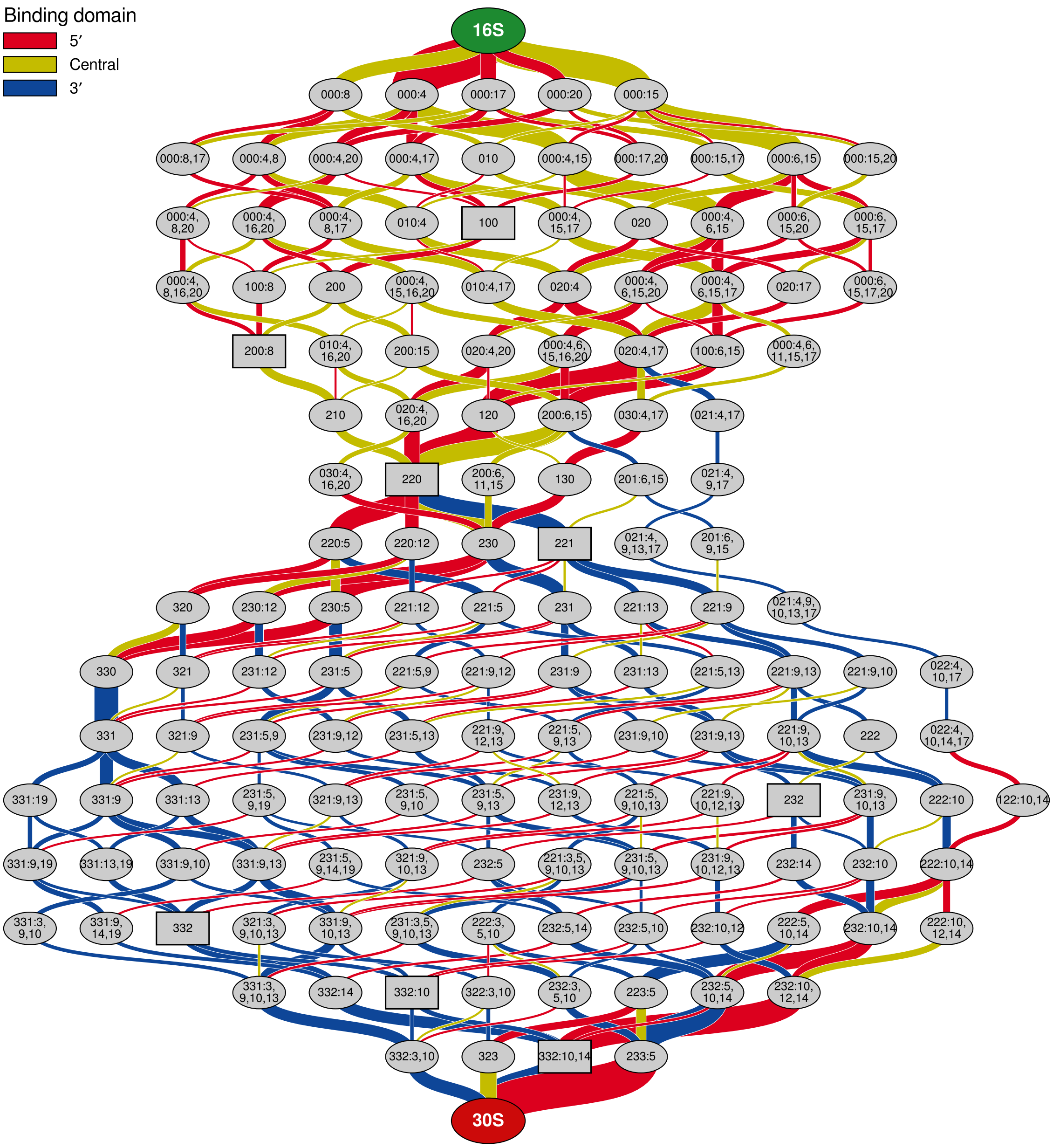


Figure C.8 Reduced assembly network for SSU assembly at 40 °C. There are 147 SSU intermediates used in the biogenesis model, excluding the completed SSU and bare 16S. Each node is an assembly intermediate, labeled according to which proteins are bound. A three digit number describes the set of r-proteins bound to each domain (5′-, central-, and 3′- respectively), All remaining r-proteins are listed after the three digit number. The edges connecting the intermediates represent the r-protein binding reactions. The width represents the total amount of intermediate converted by that reaction, and the color indicates the binding domain of that protein (5′-red, central-yellow, and 5′-blue.) Predicted assembly intermediates from P/C qMS and cryo-EM[108] are represented using rectangles.

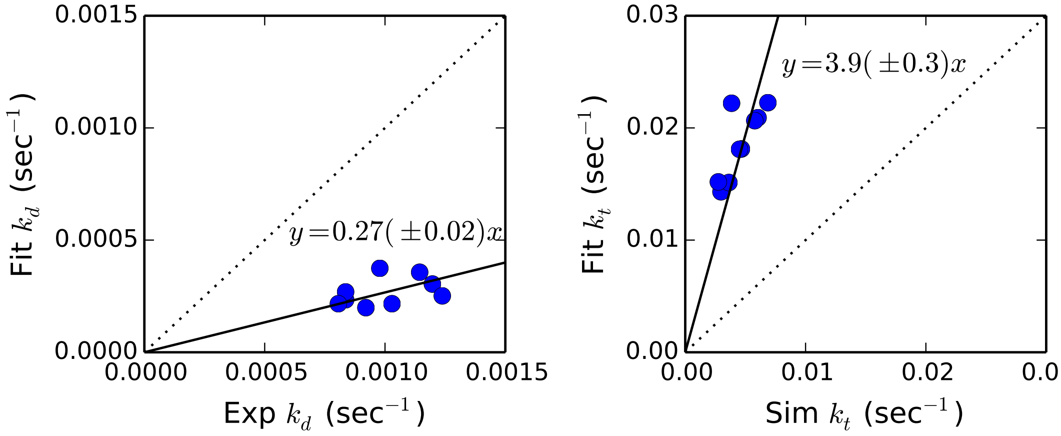


Figure C.9 Comparison of fitted k_d and k_t estimated by minimizing the mean square deviation between the distributions calculated in the RBM and those predicted by Peterson et al.[1]. The dotted lines indicate $y = x$. The solid line indicates the linear trend between the fitted k_t and the value used in the RBM.

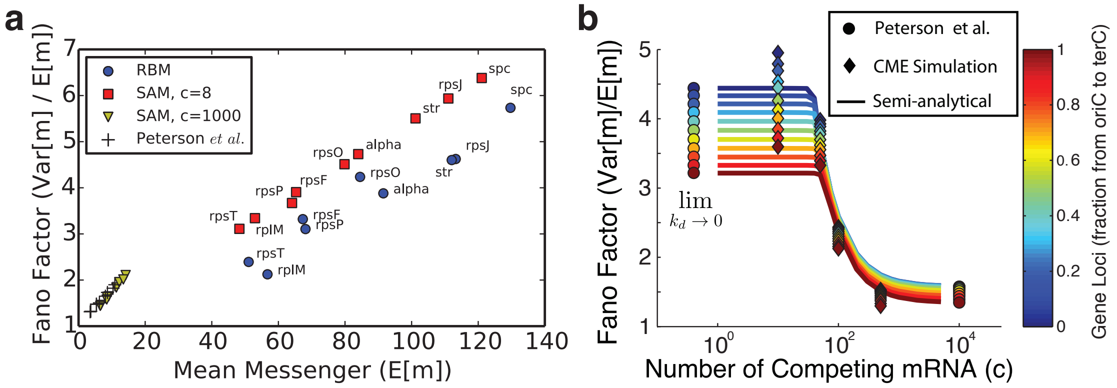


Figure C.10 (a) Fano factor plotted versus mean messenger for r-protein operon messengers from the biogenesis simulations. Statistics from ribosome biogenesis simulations (blue dots) are poorly represented by the analytical theory of Peterson et al.[1] (plus symbols). This prompts a new theory (SAM) that includes the effect of mRNA sequestration by unbound ribosomes (red squares) which better captures the simulated data. Additionally, it is shown that in the limit where the number of competing mRNAs is large, the SAM theory converges nearly to the theory of Peterson et al.[1] (yellow triangles). (b) Fano factors for an average r-protein mRNA as a function of gene loci and number of competing genes. Color indicates fraction along genome from *oriC* to *terC*. Circles indicate predictions from theory[1] either without modification (right-most) or in the limit where $k_d \to 0$ (left-most). Diamonds indicate the results of CME simulations of a models that explicitly account for gene duplication, cell division, interactions with ribosomes, and varying numbers of genes being expressed. Lines indicate the semi-analytical model developed herein.