# Uncovering principles of the central dogma of biology: noise, growth robustness, and overabundance

Teresa Lo

A dissertation
submitted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy

University of Washington

2024

Reading Committee:
Paul A. Wiggins, Chair
Beth Traxler
Andrew Laszlo

Program Authorized to Offer Degree:
Physics

University of Washington

## Abstract

Uncovering principles of the central dogma of biology: noise, growth robustness, and overabundance

Teresa Lo

Chair of the Supervisory Committee:
Paul A. Wiggins
Department of Physics

The central dogma of biology describes the most essential processes for all living systems to function: the interactions between messenger RNA, DNA, and protein. Each process in the central dogma occurs on the scale of single molecules and is inherently stochastic, leading to noisy gene expression. Protein noise for eukaryotic and prokaryotic cells has been observed experimentally and modeled theoretically, but its significance has not been understood at the fundamental level. Furthermore, theories have not been able to explain the paradox of how gene expression should be optimal after millions of years of evolution, yet observed to be overabundant: produced in excess of what is required for function.

This dissertation covers modeling to explore protein noise, and discusses the cellular strategy of overabundance. In addition, I document my development of OmniSegger, a package improving the accuracy of analyses of cellular timelapses and enabling analysis of unusual cell morphologies, which was critical to quantifying overabundance experimentally.

# Contents

# List of Figures

# GLOSSARY

*A. baylyi*: *Acinetobacter baylyi.* A nonmotile, gram-negative coccobacillus that grows under aerobic conditions.

DNA: Deoxyribonucleic acid

*E. coli*: *Escherichia coli.* A gram-negative bacterium commonly found in the lower intestine of warm-blooded organisms.

GFP: Green fluorescent protein.

MATLAB: MATrix LABoratory. A programming language and software specializing in computations involving matrices.

MRNA: Messenger ribonucleic acid

RLTO: Robustness-Load Trade-Off Model

RNA: Ribonucleic acid

TFNSEQ: Transformation transposon insertion mutant sequencing/transformation Tn-seq

# ACKNOWLEDGMENTS

This thesis has been the culmination of six long years of work in the Wiggins Lab, and many years of study prior to graduate school.

My journey in physics began with Ms. Guthrie, my high school physics teacher. In undergrad, I shared many long nights and text messages discussing physics with Courtney and Will, and had wonderful mentorship from Deborah Fygenson. I would also like to thank the Inverted Pentagram for keeping my sleep-deprived sanity mostly intact. My decision to apply to the PhD program at UW was greatly inspired and aided by Cruz Ortiz and Yvette Martinez-Vu from the McNair Scholars Program. I would like to thank my first year cohort, specifically Arnab, Ryan, and Wan Jin, who have been able to relate to the very specific situations of going through our physics program. I am also very grateful to have met my friends Murali, Nikita, and Roland. In addition, my first year office-mate, Dan, and I shared many evenings in PAB discussing *extremely* important topics such as Panofsky and quantum scattering.

I was introduced on my first day in the Wiggins Lab to Sarah Mangiameli, who showed me many of the techniques for Expansion Microscopy. I met my lab mates Isaac Shelby and Dean Huang, who helped me from small tasks like autoclaving lab waste to large tasks like answering scientific questions. Thank you to Beth Traxler and Eli Gachelet for patiently teaching me how to be a microbiologist. I really appreciate Kevin Cutler and Dani Koch, who I am glad to have joined together with as people who enjoy computers, coding, cell segmentation, and cats. Han Kyou James Choi is the final victim in the Wiggins Lab (so far), and it has been nice to work together and keep my sanity in my later years. Of course,

I could not have done this work without my advisor, Paul Wiggins, who has been incredibly inspiring both scientifically and personally, with an unlimited amount of entertaining (and sometimes educational) anecdotes. Finally, I would like to thank the undergraduate students that I have had the opportunity to work with: Nandor Marosan, Sherry Yang, and Brighton Reed.

I cannot forget the sacrifice of the millions of cells that have passed away over the years of all my experiments. Thank you.

I would like to acknowledge my funding, without which would have made my degree impossible: TA funding from the Physics department and grant funding from the NIH and NSF. Thank you to my committee for their support and advice: Andrew Laszlo, Beth Traxler, Armita Nourmohammad, Jason Detwiler, and Mark Rudner.

In my personal life, I would like to thank Lan-Anh, April, Jasmine, Meghann, Josette, Devin, Elise, Paco, and Syrian. I would also like to acknowledge my favorite musicians: MAMAMOO, the Cocteau Twins, Men I Trust, Hoody, jeebanoff, Kali Uchis, Summer Walker, Kehlani, and Jimmy Brown. I very greatly appreciate David, who has been the world's best source of inspiration, joy, and perseverance to me as a speaking coach, cooking consultant, personal trainer, comedian, philosopher, chocolatier, and more. Finally, thank you to my family and Sonni, who have supported me in my research and everything beyond. Everyone has their own path to follow. ▾

# Chapter 1
# INTRODUCTION

Entropy is present at the most fundamental level in living systems, resulting in stochastic variation across both microscopic and macroscopic scales. Proteins are essential to regulating biological systems in order to maximize their fitness, yet gene expression levels can vary significantly among individual cells. Experimental observations indicate that different genes exhibit varying degrees of this expression variability; however, efforts to model and predict these variations have, at best, yielded imprecise results. Furthermore, previous work has not been able to explain the influence of gene expression noise in optimizing cell fitness. There have been several hypotheses about fundamental strategies that cells use to manage noise in gene expression, including mechanisms such as bet-hedging or evolutionary regulation aimed at producing a precise amount of protein during each cell cycle. While these hypotheses addressed superficial cellular strategies, a comprehensive analysis of how cells maintain growth robustness in the face of protein expression noise at a mechanistic and fundamental level remained largely unaddressed.

A series of experiments in *Acinetobacter baylyi* (*A. baylyi*)[1] lead to surprising observations: when essential genes were knocked out, cells continued to proliferate. Motivated by these observations, our lab began to explore the hypothesis that cells produce and carry more protein than minimally necessary, which is referred to as "overabundance". In order to develop this hypothesis, we first developed a mathematical framework to describe and predict protein expression noise. The model predicts a fundamental strategy for transcription

[1] *A. baylyi* is an emerging model organism due to its extremely high natural competency/transformation efficiency: roughly 1/100 cells. In comparison, *E. coli* has a competency of about 1/1,000,000 cells.

conserved between eukaryotic and prokaryotic cells, which we termed the *one message rule* for essential genes. Furthermore, the mathematical framework also predicts protein noise as experimentally observed for yeast more accurately than canonical models.

Building upon our noise model framework and observations in *A. baylyi*, we developed a mathematical model to describe cellular growth and fitness in terms of cell cycle parameters: the Robustness-Load Trade-Off (RLTO) Model. In the model, there are two key factors which lead to loss in growth rate: the slowing of essential processes due to lack of essential protein, and the cost placed on metabolism from producing protein. The RLTO model infers predictions for translation efficiencies and proteomes for eukaryotic cells, and agrees with the implications of the *one message rule*.

The quantitative analysis of imaging experiments in *A. baylyi* demonstrating overabundance would not have been possible without a key tool I developed: OmniSegger. The deletion of essential genes leads to cells developing unusual morphologies, which were unable to be segmented and analyzed by analysis packages at the time of the experiments. OmniSegger combines the great improvements to cell segmentation from Kevin Cutler's Omnipose package, with the Wiggins Lab's MATLAB-based analysis suite, SuperSegger, which was developed to extract cell properties from image timelapses of growing microcolonies of rod-shaped *Escherichia coli* (*E. coli)*. In addition, my undergraduate mentee, Sherry Yang, developed a new cell tracking algorithm, Bactrack. OmniSegger is a powerful and versatile tool which enables image processing, visualization, and data analysis for timelapses of diverse cell morphologies.

In my thesis, I will discuss the noise framework, the RLTO model predicting protein overabundance, and finally the development of OmniSegger.

# Chapter 2

# GENE EXPRESSION IS NOISY!

This chapter is a modified reproduction of ref. [10].

## *2.1 Author summary*

In order to understand and mathematically study protein overabundance, we first developed a framework to describe gene expression noise. Protein abundance is gamma-distributed and described by two key parameters: the message number, which is the number of messenger RNAs produced per cell cycle, and the translation efficiency, which is the number of proteins translated per messenger RNA. We had two important insights about these parameters: 1) the number of messages reported from experiment need to be rescaled by the doubling time and message lifetime of the organism, and 2) the translation efficiency is not necessarily constant. Once these parameters were properly defined, we confirmed the model via Gillespie simulation. The noise for the gamma-distributed protein abundance is solely determined by message number. We tested our model against data from literature, and found that the noise in yeast is better described by our model than by the canonical noise model. Our model also predicts the *one message rule*: a minimum of one message is transcribed per cell cycle for the evolutionarily divergent organisms human, yeast, and *E. coli*, suggesting a transcriptional program to ensure cellular robustness to noise.

## *2.2 Introduction*

All molecular processes are inherently stochastic on a cellular scale, including the processes of the central dogma, responsible for gene expression [11, 12]. As a result, the expression of every protein is subject to cell-to-cell variation in abundance [11]. Many interesting proposals

have been made to describe the potential biological significance of this noise, including bet-hedging strategies, the necessity of feedback in gene regulatory networks, *etc* [11, 13, 14]. However, it is less clear to what extent noise plays a central role in determining the function of the gene expression process more generally. For instance, Hausser *et al.* have described how the tradeoff between economy (*e.g.* minimizing the number of transcripts) and precision (minimizing the noise) explains why genes with high transcription rates and low translation rates are not observed [15]. Although these results suggest that noise may provide some coarse limits on the function of gene expression, this previous work does not directly address a central challenge posed by noise: How does the cell ensure that the lowest expression essential genes, which are subject to the greatest noise, have sufficient abundance in all cells for robust growth?

To investigate this question, we first focus on noise in *Saccharomyces cerevisiae* (yeast), and find that the noise scaling with protein abundance is not canonical. We re-analyze the canonical stochastic-kinetic model for gene expression [16, 17, 18], to understand the relationship between the underlying kinetic parameters and the distribution of protein abundance in the cell. As previously reported, we find that the protein abundance for a gene is described by a gamma distribution with two parameters: the *message number*, defined as the total gene message number transcribed per cell cycle, and the translation efficiency, which is the mean protein number translated per message. Protein expression noise is completely determined by the message number [13, 19]. Although these results have been previously reported, the distinction between message number *per cell* versus *per cell cycle* and even between *mean protein number* and *mean message number* is often neglected (*e.g.* [20]).

To explore the distinction between these parameters and provide clear evidence of the importance of the message number, we return to the analysis of noise in yeast. In yeast, the translation efficiency increases with message number [21]. By fitting an empirical model for the translation efficiency, we demonstrate that the noise should scale with a half-power of protein abundance. We demonstrate that this non-canonical scaling is observed and that our translation model makes a parameter-free prediction for the noise. The prediction is in close

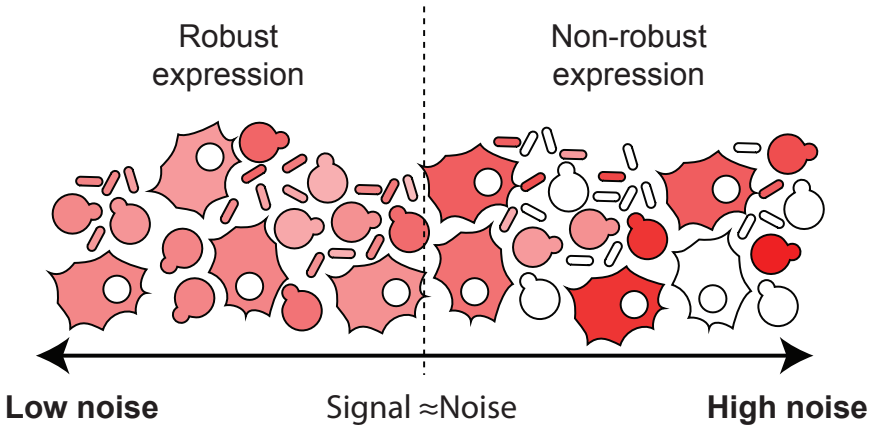



Figure 2.1: **Robustness hypothesis for essential genes:** The stochasticity in gene expression is represented by the red shading. We hypothesize that robust growth requires sufficiently low noise levels for cellular function; this critical noise level should be below the level where the signal (mean) is comparable to noise (standard deviation).

quantitative agreement with observation [1], confirming that the message number is the key determinant of noise strength.

Finally, we use this result to explore the hypothesis that there is a minimum expression level for essential genes, dictated by noise. The same mean expression level can be achieved by a wide range of different translation and transcription rates with different noise levels. We hypothesize that growth robustness requires that essential genes (but not non-essential genes) are subject to a floor expression level, below which there is too much cell-to-cell variation to ensure growth. To test this prediction, we analyze transcription in three model organisms, *Escherichia coli*, yeast, and *Homo sapiens* (human), with respect to three related gene characteristics: transcription rate, cellular message number, and message number per cell cycle. As predicted by the noise-based mechanism, we observe an organism-independent floor for the number of messages transcribed per cell cycle for essential genes, but not non-essential genes. We conclude that virtually all essential genes are transcribed at a rate of at least once per cell cycle. This analysis strongly supports the hypothesis that the same

biological optimization imperatives, which determine the transcription rates of many low-expression genes, are conserved from *E. coli* to human.

## 2.3 Results: Details of the noise model

### *2.3.1 Implications of noise on growth robustness*

With the realization of the stochasticity of central dogma processes, a key question is how cells can grow robustly in spite of cell-to-cell variations in protein expression. The noise in protein abundance is defined as the coefficient of variation squared [22, 23, 1]:

$$\mathrm{CV}_p^2 \equiv \frac{\sigma_p^2}{\mu_p^2}, \tag{2.1}$$

where $\sigma_p^2$ is the variance of protein number and $\mu_p \equiv \overline{N}_p$ its mean. It is important to emphasize that protein abundance must double between birth and cell division in symmetrically dividing cells during steady state growth. The protein abundance should therefore be interpreted either as expression per unit volume [2] or the abundance associated with cells of a defined volume [1].

The coefficient of variation is inversely related to protein abundance and therefore low-copy proteins have the highest noise [22, 23, 13, 19, 2, 1]. The challenge faced by the cell is that many essential proteins, strictly required for cell growth, are relatively low abundance. How does the cell ensure sufficient protein abundance in spite of cell-to-cell variation in protein number? It would seem that growth robustness demands that, for essential proteins, the mean should be greater than the standard deviation:

$$\mathrm{CV}_p^2 < 1, \tag{2.2}$$

in order to ensure that protein abundance is sufficiently high enough to avoid growth arrest (see Fig. 2.1). To what extent do essential proteins obey this noise threshold?

### 2.3.2 *Canonical models fail to characterize noise*

Noise has been argued to be proportional to inverse protein abundance (*e.g.* [13, 14, 20]):

$$\mathrm{CV}_p^2 \propto \mu_p^{-1}, \tag{2.3}$$

for low abundance proteins, motivated both by theoretical and experimental results [20, 2] and in some cases obeying a low-translation efficiency limit [2]:

$$\mathrm{CV}_p^2 \approx \mu_p^{-1}. \tag{2.4}$$

Can this model be used to make quantitative predictions of the noise? *E.g.*, is the scaling of Eq. 2.3 correct? Can the coefficient of proportionality be predicted? Although Eq. 2.3 appears to describe *E. coli* quite well [2], the situation in yeast is more complicated [1]. To analyze the statistical significance of the deviation from the canonical noise model in yeast, we can fit an empirical model to the noise [22, 23]:

$$\mathrm{CV}_p^2 = \frac{b}{\mu_p^a} + c. \tag{2.5}$$

In the null hypothesis, $a = 1$ (canonical scaling), while $b$ and $c$ are unknown parameters. $c$ corresponds to the noise floor.

In the alternative hypothesis, the maximum likelihood estimate (MLE) of the empirical noise model (Eq. 2.5) parameters are (Fig. 2.2):

$$a = 0.57 \pm 0.02, \tag{2.6}$$

$$b = 3.0 \pm 0.5, \tag{2.7}$$

$$c = 0.013 \pm 0.001, \tag{2.8}$$

[1] Although there have been claims that Eq. 2.3 is consistent with the data [20], these authors did not fit competing models, nor did they perform a proteome-wide analysis of protein abundance and noise.

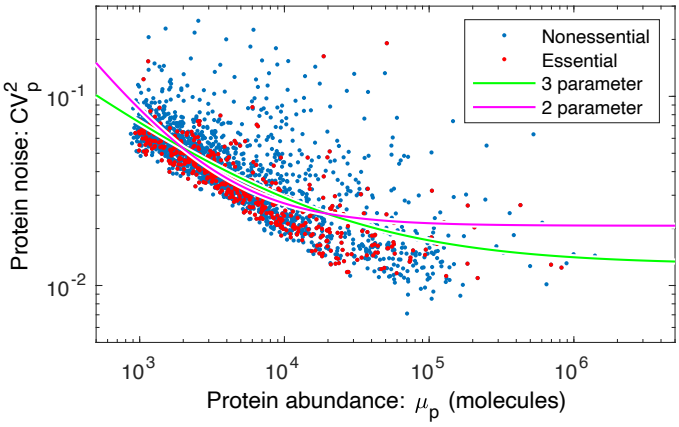



Figure 2.2: **Yeast noise from fit against canonical noise model, with a noise floor.** Yeast noise data from [1] fit with the 2- (null hypothesis $\mathrm{CV}^2_\mathrm{p} = \frac{\mathrm{b}}{\mu^1_\mathrm{p}} + \mathrm{c}$) and 3- parameter (alternative hypothesis $\mathrm{CV}^2_\mathrm{p} = \frac{\mathrm{b}}{\mu^\mathrm{a}_\mathrm{p}} + \mathrm{c}$) models.

where the parameter uncertainty has been estimated using the Fisher Information in the usual way using the MLE estimate of the variance.

The canonical model fails to fit the noise data for yeast as reported by Newman *et al.* [1]: The null hypothesis is rejected with p-value $p = 6 \times 10^{-36}$. The model fit to the data is shown in Fig. 2.4. The estimated scaling exponent for protein abundance in the alternative hypothesis is $a = 0.57 \pm 0.02$. As shown in Fig. 2.4, even from a qualitative perspective, the scaling of the yeast noise at low copy number is much closer to $\mu_p^{-1/2}$ than to canonical assumption $\mu_p^{-1}$ (Eq. 2.3). In particular, above the detection threshold, the noise is always larger than the low-translation efficiency limit (Eq. 2.4).

The noise model parameters were also determined for *E. coli*:

$$a = 1.22 \pm 0.01, \tag{2.9}$$

$$b = 1.27 \pm 0.02, \tag{2.10}$$

$$c = 0.154 \pm 0.002, \tag{2.11}$$

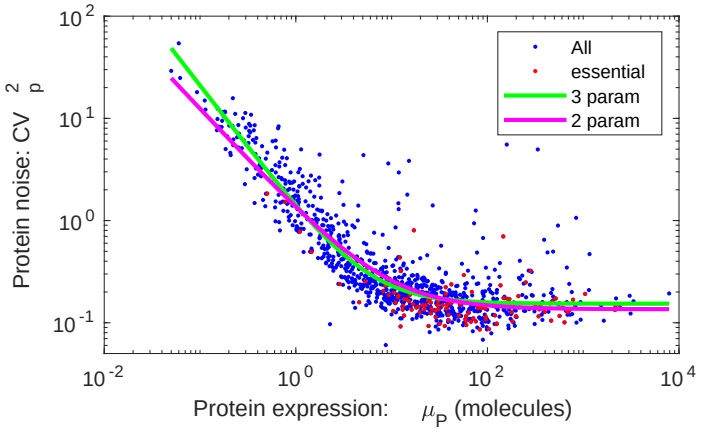



Figure 2.3: **Three-parameter fit to *E. coli* noise.** Noise as a function of protein abundance from [2] was fit to the 3-parameter noise model (Eq. 2.5). From the fit, protein noise scales proportionally with $\mu_p^{-1.22}$, which is a close result to the canonical model with $\mu_p^{-1}$.

with the corresponding fit shown in Fig. 2.3. Since $a$ is close to 1, the canonical model with $a = 1$ (Eq. 2.3) is a reasonable approximation for the noise in *E. coli*.

### *2.3.3 Stochastic kinetic model for central dogma.*

To understand the failure of the canonical assumptions, we revisit the underlying model. The central dogma describes multiple steps in the gene expression process[2]: Transcription generates messenger RNA (mRNA) messages [24] which are then translated to synthesize the protein gene products [24]. Both mRNA and protein are subject to degradation and dilution [25]. (See Fig. 2.5A.) At the single cell level, each of step of the central dogma is stochastic and expected to be a Poissonian process. We model these processes with the

[2]The central dogma describes the flow of information, and includes reverse transcription and DNA replication; however, we are not concerned with these processes in this analysis as they are not directly involved in gene expression.

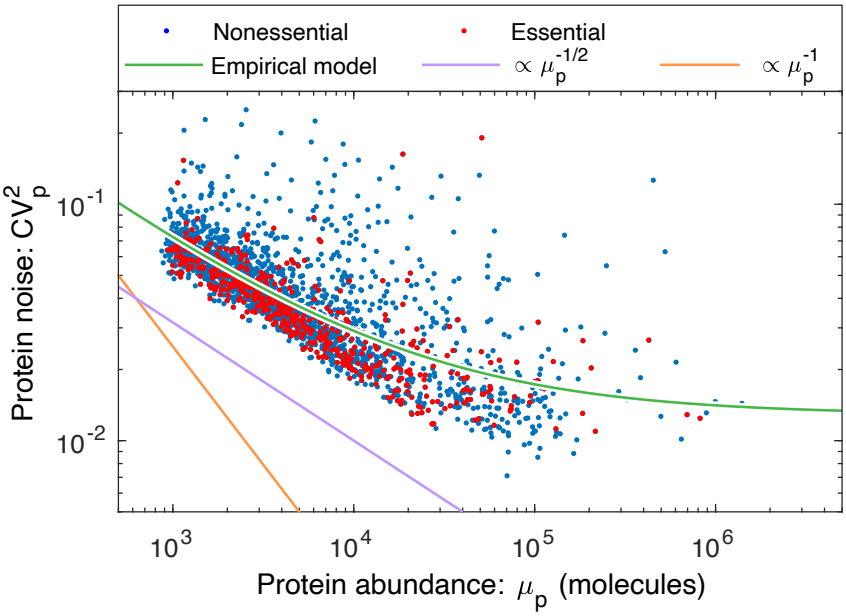



Figure 2.4: **A non-canonical scaling is observed for gene-expression noise in yeast.** The protein expression noise ($\mathrm{CV}_p^2$) for yeast scales like $\mu_p^{-1/2}$ (violet) rather than the canonical $\mu_p^{-1}$ (orange) for low-abundance proteins. (Data from Ref. [1].) An empirical noise model (Eq. 2.5, green) fit to the essential genes gives an estimate of the protein-abundance scaling of $\mu_p^{-0.57}$.

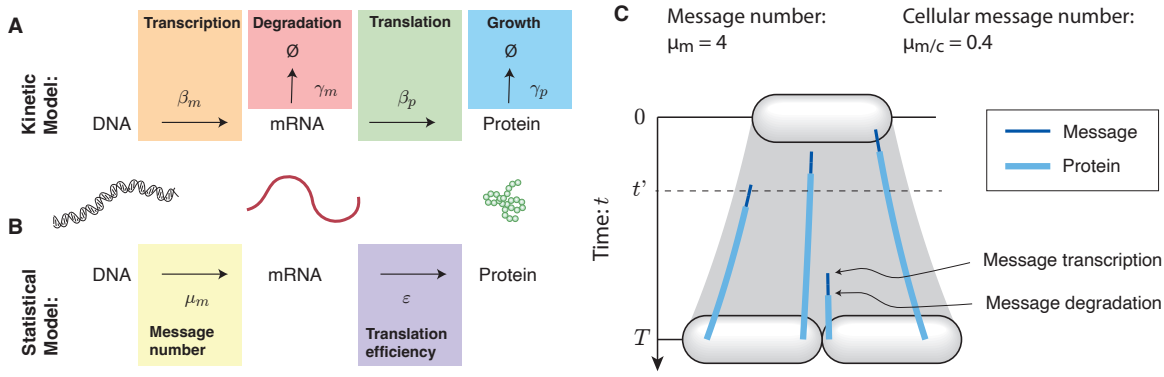



Figure 2.5: **Panel A: Kinetic model for the central dogma.** The central dogma is a stochastic-kinetic model for protein synthesis, described by four gene-specific rate constants: the transcription rate ($\beta_m$), the message degradation rate ($\gamma_m$), the translation rate ($\beta_p$), and the dilution rate ($\gamma_p$). **Panel B: Statistical model for the central dogma.** The predicted distribution in protein abundance is described by a gamma distribution, which is parameterized by two unitless constants: the shape parameter $\mu_m$, the mean number of messages transcribed per cell cycle, and the scale parameter $\varepsilon$, the mean number of proteins translated per message. **Panel C: Message number.** The *message number* ($\mu_m$) is defined as the mean total number of messages (dark blue) transcribed per cell cycle. Here, four total messages are transcribed and translated to protein (light blue); however, due to message degradation, at time $t'$, only one message is present in the cell. Cellular message number ($\mu_{m/c}$) is defined as the mean number of messages per cell at time $t$.

stochastic-kinetic scheme [24]:

$$\begin{array}{ccccc} \text{DNA} & \xrightarrow{\beta_m} & \text{mRNA} & \xrightarrow{\beta_p} & \text{Protein} \\ & & \gamma_m \downarrow & & \gamma_p \downarrow \\ & & \varnothing & & \varnothing, \end{array} \tag{2.12}$$

where $\beta_m$ is the transcription rate ($s^{-1}$), $\beta_p$ is the translation rate ($s^{-1}$), $\gamma_m$ is the message degradation rate ($s^{-1}$), and $\gamma_p$ is the protein effective degradation rate ($s^{-1}$). The message lifetime is $\tau_m \equiv \gamma_m^{-1}$. For most protein in the context of rapid growth, dilution is the dominant mechanism of protein depletion and therefore $\gamma_p$ is approximately the growth rate [26, 27, 2]: $\gamma_p = T^{-1} \ln 2$, where $T$ is the doubling time. We discuss an alternative description of the central dogma below.

### *2.3.4 Statistical model for protein abundance*

To study the stochastic dynamics of gene expression, we used a stochastic Gillespie simulation [28, 29]. In particular, we were interested in the explicit relation between the kinetic parameters $(\beta_m, \gamma_m, \beta_p, \gamma_p)$ and experimental observables. Assuming the lifetime of the cell cycle ($T_{cc} = 30$ min) [30], mRNA lifetime ($\tau_m = 2.5$ min) [31], and translation rate ($\beta_p \approx 500$ $hr^{-1}$), the protein distributions for several mean expression levels were numerically generated for exponential growth with 100,000 stochastic cell divisions, with protein partitioned at division following the binomial distribution.

The gamma distributions for each mean message number with scale and shape parameters determined by the corresponding translation efficiency and message number ($\theta = \varepsilon \ln 2$, $k = \frac{\mu_m}{\ln 2}$) as used for the Gillespie simulation were plotted with the protein distributions:

$$p(n|\theta, k) = \frac{1}{\Gamma(k)\theta^k} n^{k-1} e^{-\frac{n}{\theta}}, \tag{2.13}$$

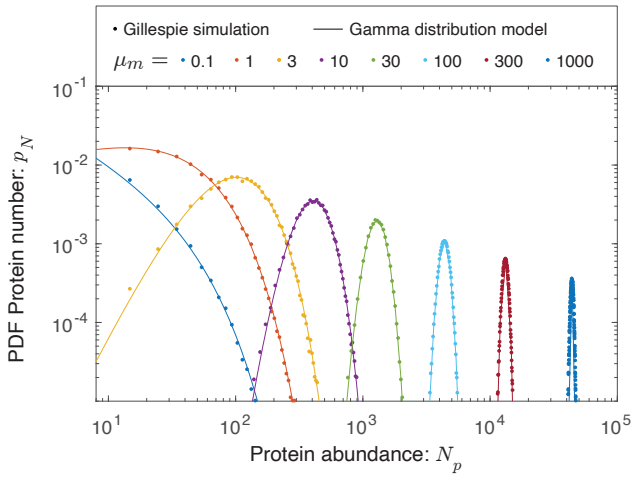



Figure 2.6: **The protein abundance is approximately gamma distributed.** Protein abundance was modeled for eight different transcription levels using a Gillespie simulation, including the stochastic partitioning of the proteins between daughter cells at cell division. The range in abundance matches the observed range of expression levels in the cell. We observed that the simulated protein abundances were well described by gamma distributions.

confirming that a gamma distribution well describes the simulation (see Fig. 2.6). We also considered other distributions such as Poisson, which did not match well with our simulation (see Fig. 2.7).

Consistent with previous reports [13, 19], our simulation confirms that the distribution of protein number per cell (at cell birth) was described by a gamma distribution: $N_p \sim \Gamma(\theta_\Gamma, k_\Gamma)$, where $N_p$ is the protein number at cell birth and $\Gamma$ is the gamma distribution which is parameterized by a scale parameter $\theta_\Gamma$ and a shape parameter $k_\Gamma$. The relation between the four kinetic parameters and these two statistical parameters has already been reported, and have clear biological interpretations [19]: The scale parameter:

$$\theta_\Gamma = \varepsilon \ln 2, \tag{2.14}$$

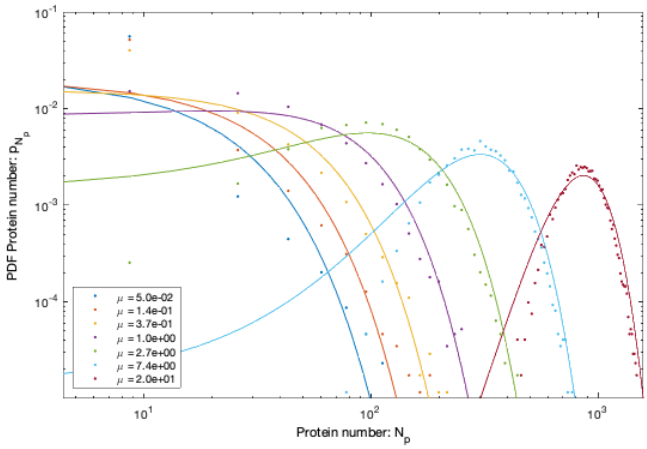



Figure 2.7: **The protein abundance is *not* Poisson distributed.** The Poisson distribution does not accurately model protein abundance.

is proportional to the translation efficiency:

$$\varepsilon \equiv \frac{\beta_p}{\gamma_m}, \tag{2.15}$$

where $\beta_p$ is the translation rate and $\gamma_m$ is the message degradation rate. $\varepsilon$ is understood as the mean number of proteins translated from each message transcribed. The shape parameter $k_\Gamma$ can also be expressed in terms of the kinetic parameters [19]:

$$k_\Gamma = \frac{\beta_m}{\gamma_p}; \tag{2.16}$$

however, we will find it more convenient to express the scale parameter in terms of the cell-cycle message number:

$$\mu_m \equiv \beta_m T = k_\Gamma \ln 2, \tag{2.17}$$

which can be interpreted as the mean number of messages transcribed per cell cycle. Forthwith, I abbreviate this quantity *message number* or *messages per cell cycle*.

| Model organism | Growth condition | Doubling time: $T$ | Message lifetime: $\tau_m$ | Message recycling ratio: $T/\tau_m$ | Total number of messages /cell: $N_{m/c}^{\text{tot}}$ | Total number of messages /cell-cycle: $N_m^{\text{tot}}$ | Total number of proteins: $N_p^{\text{tot}}$ | Average translation efficiency: $\varepsilon$ | Average translation rate: $\beta_p$ ($\text{h}^{-1}$) |
|---|---|---|---|---|---|---|---|---|---|
| *Escherichia coli* (*E. coli*) | LB | 30 min | 2.5 min | 12 | $7.8 \times 10^3$ | $9.4 \times 10^4$ | $3 \times 10^6$ | 32 | 770 |
| | M9 | 90 min | 2.5 min | 36 | $2.4 \times 10^3$ | $8.6 \times 10^4$ | $3 \times 10^6$ | 35 | 833 |
| *Saccharomyces cerevisiae (Yeast–haploid)* | YEPD | 90 min | 22 min | 4 | $2.9 \times 10^4$ | $1.2 \times 10^5$ | $5 \times 10^7$ | 420 | 1100 |
| *Homo sapiens (Human)* | Tissue | 24 h | 14 h | 1.7 | $3.6 \times 10^5$ | $6.2 \times 10^5$ | $2 \times 10^9$ | $3.2 \times 10^3$ | 230 |

Table 2.1: **Central dogma parameters for three model organisms.** Columns three through seven hold representative values for measured central-dogma parameters for the model organisms described in the paper.

In terms of two gamma parameters, the mean and the squared coefficient of variation are:

$$\mu_p = k_\Gamma \theta_\Gamma = \mu_m \varepsilon \tag{2.18}$$

$$\text{CV}_p^2 = \frac{1}{k_\Gamma} = \frac{\ln 2}{\mu_m}, \tag{2.19}$$

where the noise depends on the message number ($\mu_m$), not the mean protein number ($\mu_p$). (Eq. 2.19 only applies when $\varepsilon \gg 0$ [13, 19].) Are these theoretical results consistent with the canonical model (Eq. 2.3)? We can rewrite the noise in terms of the protein abundance and translation efficiency:

$$\text{CV}_p^2 = \frac{\varepsilon \ln 2}{\mu_p}, \tag{2.20}$$

which implies that the canonical model only applies when the translation efficiency ($\varepsilon$) is independent of expression ($\mu_p$).

### *2.3.5 Measuring the message number*

The prediction for the noise (Eq. 2.19) depends on the messages per cell cycle ($\mu_m$). However, mRNA abundance is typically characterized in literature by a closely related, but distinct quantity: quantitative RNA-Seq and methods that visualize fluorescently-labeled

mRNA molecules typically measure the *instantaneous* number of messages per cell [16]. We will call the mean of this number the *cellular message number* $\mu_{m/c}$. In the stochastic-kinetic model, these different message abundances are related:

$$\mu_m = \frac{T}{\tau_m}\mu_{m/c}, \tag{2.21}$$

by the message recycling ratio, $T/\tau_m$, which can be interpreted as the average number of times messages are recycled during the cell cycle. Thus, message number $\mu_m$ represents the instantaneous cellular message number time-averaged over the cell cycle. To estimate the message number, we will scale the observed cellular message number $\mu_{m/c}$ by the message recycling ratio, using the mean message lifetime. Fig. 2.5C illustrates the difference between the message number and the cellular message number. The mean lifetimes, message recycling ratios, as well as the total message number for three model organisms are shown in Tab. 2.1.

### *2.3.6 Construction of an empirical model for protein number*

To model the noise as a function of protein abundance ($\mu_p$), we will determine the empirical relation between mean protein levels and message abundance by fitting to Eq. 2.18. Note that the objective here is only to estimate $\mu_m$ from $\mu_p$, not to model the process mechanistically (*e.g.* [32].) The message numbers are estimated from RNA-Seq measurements, scaled as described above (Eq. 2.21). The protein abundance numbers come from fluorescence and mass-spectrometry based assays [1, 3], with overall normalization chosen to match reported total cellular protein content.

The protein abundance data for yeast grown in rich YEPD media and measured with flow cytometry fluorescence [1] were given in arbitrary units (AU). In order to convert from AU to protein number, the fluorescence values were rescaled by comparing with mass-spectrometry protein abundance data for yeast grown in rich YNB media [3]. Since the protein abundance from mass-spectrometry was given in terms of Intensity, the Intensity values were first rescaled by the total number of proteins in yeast, $5 \times 10^7$. The mass-spectrometry protein

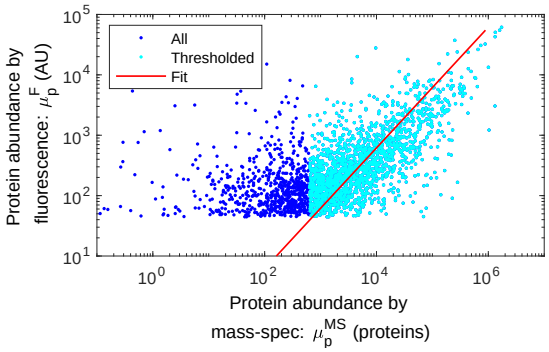



Figure 2.8: **Fit to rescale fluorescence intensity to protein number.** Protein abundance from flow cytometry fluorescence [1] as a function of mass-spectrometry scaled abundance [3]. The mass-spectrometry data was thresholded at 10 proteins, and then a linear fit was performed to find the offset of 3.9, which was used to convert protein fluorescence AU to number.

data was thresholded at 10 proteins, based on the assumption that the noise of the data for 10 and fewer proteins makes the data unreliable. Next, the log of the fluorescence protein abundance in AU as a function of the log of thresholded mass-spectrometry protein abundance was fit as a linear function with an assumed slope of 1 to find the offset, 3.9, (Fig. 2.8) which corresponds to a multiplicative scaling factor (Eqn. 2.23). We then used that offset value to rescale the fluorescence data from AU to protein number. We also compared to yeast grown in SD media [1] and found a similar offset result.

$$\log \mu_P^{\mathrm{F}} = m \log \mu_P^{\mathrm{MS}} + b \tag{2.22}$$

$$\mu_P^{\mathrm{F}} = b(\mu_P^{\mathrm{MS}})^m \tag{2.23}$$

The resulting fit generates our empirical translation model for yeast:

$$\mu_p \;\; = \;\; 8.0 \; \mu_m^{2.1}, \tag{2.24}$$

where both means are in units of molecules. The data and model are shown in Fig. 2.9A.

### *2.3.7 Prediction of the noise scaling with abundance.*

Now that we have fit an empirical model that relates $\mu_p$ and $\mu_m$, we return to the problem of predicting the yeast noise. We apply the relation (Eq. 2.24) to Eq. 2.19 to make a parameter-free prediction of the noise as a function of protein abundance:

$$\mathrm{CV}_p^2 = 1.9\ \mu_p^{-0.48}. \tag{2.25}$$

Our noise model (Eq. 2.25) makes both a qualitative and quantitative prediction: (i) From a qualitative perspective, the model suggests that the $\mu_p$ exponent should be roughly $\frac{1}{2}$ for yeast, rather than the canonically assumed scaling exponent of 1. (ii) From a quantitative perspective, the model also predicts the coefficient of proportionality if the empirical relation between protein and message abundances is known (Eq. 2.24).

### *2.3.8 Observed noise in yeast matches the predictions of the empirical model.*

Newman *et al.* have characterized protein noise by flow cytometry of strains expressing fluorescent fusions expressed from their endogenous promoters [1]. The comparison of this data to the prediction of the statistical expression model (Eq. 2.25) are shown in Fig. 2.9. From a qualitative perspective, the predicted scaling exponent of $-0.48$ comes very close to capturing the scaling of the noise, as determined by the direct fitting of the empirical noise model (Eq. 2.5 and Fig. 2.4). From a quantitative perspective, the predicted coefficient of Eq. 2.25 also fits the observed noise.

From both the statistical analysis (Eq. 2.5) and visual inspection (Fig. 2.10C), it is clear that the noise in yeast does not obey the canonical model (Eq. 2.3). However, the noise in *E. coli* does obey the canonical model for low copy messages [2]. (See Fig. 2.10C.) Why does the noise scale differently in the two organisms? The key difference is that the empirical relation between the protein and message numbers are different. In *E. coli*, $\mu_p \propto \mu_m^1$ [33].

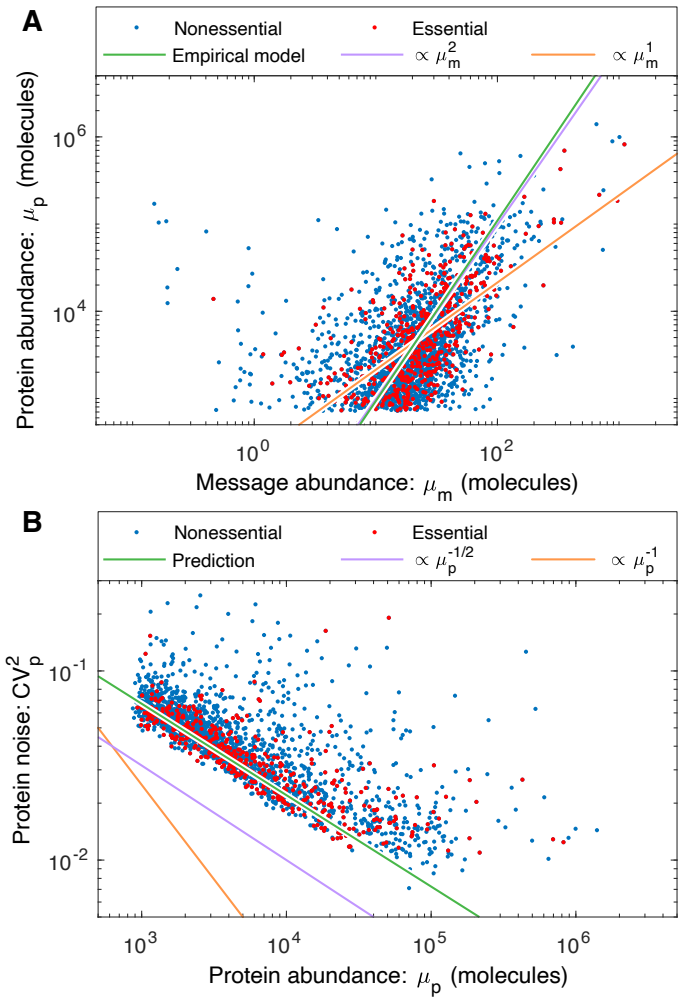



Figure 2.9: **Panel A: An empirical model for protein number $\mu_p$ in yeast.** The canonical noise model assumes constant translation efficiency, which would imply that protein number is proportional to the message number (orange); however, the empirical fit (green) shows that protein number scales close to the square of message number (violet): $\mu_p \propto \mu_m^2$. The protein abundance has a cutoff near $10^1$ due to the autofluorescence cutoff [1]. **Panel B: The statistical noise model predicts the observed noise.** The statistical noise model (Eq. 2.19) and empirical model for protein number (Eq. 2.24) make a parameter-free prediction of the noise (green). This prediction both closely matches the observed scaling ($\propto \mu_p^{-1/2}$, violet) relative to the canonical scaling ($\propto \mu_p^{-1}$, orange) and quantitatively estimates magnitude (vertical offset). This prediction does not include the contribution of noise floor, relevant for describing high-expression proteins.

Our analysis therefore predicts the canonical model (Eq. 2.3) should hold for *E. coli*, but not for yeast, as illustrated schematically in Fig. 2.10C.

In general, the central dogma predicts that the noise will have a coefficient of variation [13, 19]:

$$\mathrm{CV}_p^2 \approx \frac{1}{\mu_p} + \frac{\varepsilon \ln 2}{\mu_p}, \tag{2.26}$$

where the first term is not significant whenever the translation efficiency is much greater than 1, $\varepsilon \gg 1$. In both *E. coli* with average $\varepsilon \approx 30$ and yeast with average $\varepsilon \approx 420$, this would seem naively to be the case. However, since translation efficiency in yeast is not uniform, we must consider its variation for low-expression proteins. We estimate that the detection efficiency in yeast is roughly $10^3$ molecules. Using Eq. 2.24, we estimate that $\varepsilon \approx 100$ at the low-expression detection limit.

In *E. coli*, the situation is somewhat more complicated. Unlike yeast, the translation efficiency is roughly constant (at high to intermediate expression levels) with respect to expression level [33], and therefore both terms in Eq. 2.26 are expected to scale like the canonical model ($\propto \mu_p^{-1}$). However, it is clear that the translation efficiency must significantly decrease for the lowest abundance proteins. This is visible even in Ref. [33] Fig. 1B, where the data falls below the predicted protein abundance at low message number. Note that these mass-spec measurements are not as sensitive as fluorescence-based measurements (*e.g.* only 64% proteome could be detected [34]). Furthermore, fits to the *E. coli* noise (Eq. 2.10) are consistent only with low values of $\varepsilon$. At sufficiently high expression levels such that we are confident about the translation efficiency, the noise is already very close to the noise floor.

### *2.3.9 Implications of growth robustness for translation*

Before continuing with the noise analysis, we shift our focus on the significance of the empirical relationship between the protein and message numbers (Eq. 2.24). How can the cell counteract noise-induced reductions in robustness? Eq. 2.18 implies that gene expression can be thought of as a two-stage amplifier [24]: The first stage corresponds to transcription

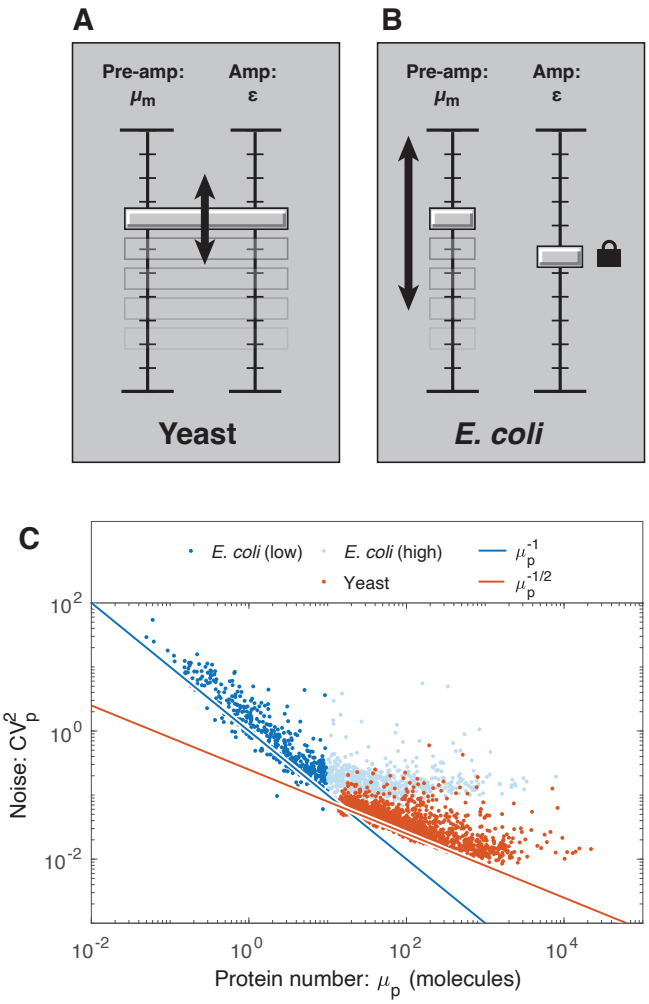



Figure 2.10: **Understanding the distinct central dogma strategies using the amplifier analogy. Panel A: Yeast.** High expression ($\mu_p$) is typically achieved by coordinated small increases in both transcription ($\mu_m$) and translation ($\varepsilon$), relative to low-expression genes. **Panel B: *E. coli*.** High expression ($\mu_p$) is typically achieved by a large increase in transcription ($\mu_m$) only, relative to low-expression genes. Translation ($\varepsilon$) is uncorrelated. **Panel C: Distinct noise scaling with gene expression.** Due to the coordinated changes in both transcription and translation in yeast, noise scaling is weaker than in *E. coli*, where only transcription changes. The noise of high-expression *E. coli* genes is determined by the noise floor.

with a gain of message number $\mu_m$, and the second stage corresponds to translation with a gain in translation efficiency $\varepsilon$. (See Fig. 2.10AB.) The noise is completely determined by the first stage of amplification, provided that $\varepsilon \gg 0$ [13, 19]. Genes with low transcription levels are the noisiest. For these genes, the cell can achieve the same mean gene expression ($\mu_p$) with lower noise by increasing the gain of the first stage (increasing message number) and decreasing the gain of the second stage (the translation efficiency) by the same factor. This is most clearly understood by reducing $\varepsilon$ at fixed $\mu_p$ in Eq. 2.20. Highly transcribed genes have low noise and can therefore tolerate higher translation efficiency in the interest of economy (decreasing the total number of messages) [15]. Growth robustness therefore predicts that the translation efficiency should grow with transcription level.

### *2.3.10 Translation efficiency increases with expression level in yeast*

The translation efficiency (Eq. 2.15) can be determined from the empirical translation model (Eq. 2.24):

$$\varepsilon \;=\; 8.0\ \mu_m^{1.1}, \tag{2.27}$$

as a function of message number. In yeast, the translation efficiency clearly has a strong dependence on message number $\mu_m$, and grows with the expression level, exactly as predicted by robustness arguments. We note the contrast to the translation efficiency in *E. coli*, which is observed to be roughly constant [33].

### *2.3.11 Implications of growth robustness for transcription.*

In addition to the prediction of translation efficiency depending on transcription, a second qualitative prediction of growth robustness is that essential gene expression should have a noise ceiling, or maximum noise level (Eq. 2.2), where noise above this level would be too great for robust growth. The fit between the statistical model and the observed noise has an important implication beyond confirming the predictions of the telegraph and statistical

models for noise: The identification of the message number, $\mu_m$, as the key determinant of noise allows us to use this quantity as a proxy for noise in quantitative transcriptome analysis.

To identify a putative transcriptional floor, we now broaden our consideration beyond yeast to characterize the central dogma in two other model organisms: the bacterium *Escherichia coli* and *Homo sapiens* (human). We will also analyze three different transcriptional statistics for each gene: transcription rate ($\beta_m$), cellular message number ($\mu_{m/c}$), and message number ($\mu_m$). Analysis of these organisms explores orders-of-magnitude differences in characteristics of the central dogma, including total message number, protein number, doubling time, message lifetime, and number of essential genes. (See Tab. 2.1.) In particular, as a consequence of these differences, the three statistics describing transcription: transcription rate, cellular message number and message number are all distinct. Genes with matching message numbers in two different organisms will not have matching transcription rates or cellular message numbers. We hypothesize that cells must express essential genes above some threshold message number for robust growth; however, we expect to see that non-essential genes can be expressed at much lower levels since growth is not strictly dependent on their expression. The signature of a noise-robustness mechanism would be the absence of essential genes for low message numbers.

### *2.3.12 No organism-independent threshold is observed for transcription rate or cellular message number*

Histograms of the per-gene transcription rate and cellular message number are shown in Fig. 2.11 for *E. coli*, yeast, and human. Consistent with existing reports, essential genes have higher expression than non-essential genes on average; however, there does not appear to be any consistent threshold in *E. coli* (even between growth conditions), yeast, or human transcription, either as characterized by the transcription rate ($\beta_m$) or the cellular message

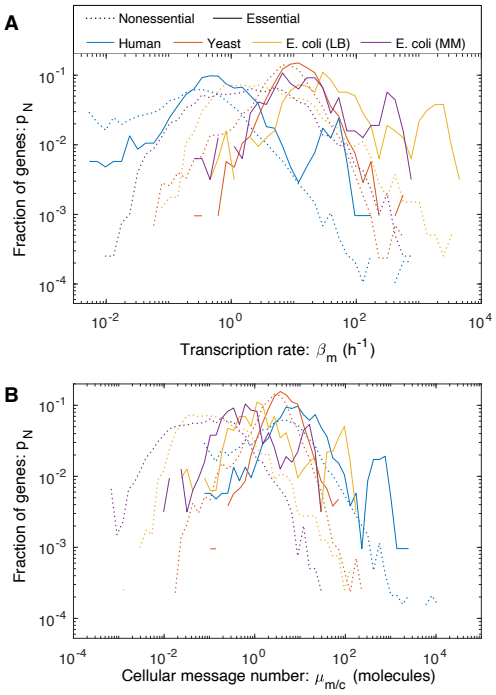



Figure 2.11: **Transcription in three model organisms.** We characterized different gene transcriptional statistics in three model organisms. In *E. coli*, two growth conditions were analyzed. **Panel A: The distribution of gene transcription rate.** The transcription rate varies by two orders-of-magnitude between organisms. **Panel B: The distribution of gene cellular message number.** There is also a two-order-of-magnitude variation between cellular message numbers.

number ($\mu_{m/c}$). For instance, the per gene rate of transcription is much lower in human cells than *E. coli* under rapid growth conditions, with yeast falling in between.

### *2.3.13 An organism-independent threshold is observed for message number for essential genes*

In contrast to the other two transcriptional statistics, there is a consistent lower limit, or floor, on message number ($\mu_m$) of somewhere between 1 and 10 messages per cell cycle for essential genes. (See Fig. 2.12.) Non-essential genes can be expressed at a much lower level. This floor is consistent not only between *E. coli*, growing under two different conditions,

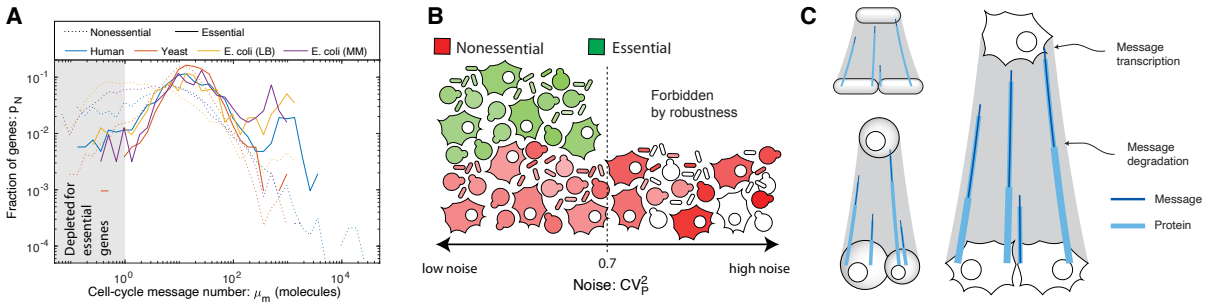



Figure 2.12: **Transcription in three model organisms. Panel A: The distribution of gene message number.** All organisms have roughly similar distributions of message number for essential genes, which are not observed for message numbers below a couple per cell cycle. However, non-essential genes can be expressed at much lower levels. **Panel B: Nonessential genes tolerate higher noise levels than essential genes.** The floor of message number is consistent with a noise ceiling of $\mathrm{CV}_p^2 = 0.7$ for essential genes (green). Nonessential genes (red) are observed with lower transcription levels. **Panel C: Conserved transcriptional program for essential genes.** The message number per gene (number of messages transcribed per cell cycle) is roughly identical in *E. coli*, yeast, and human. We show this schematically.

but also between the three highly-divergent organisms: *E. coli*, yeast and human. We will conservatively define the minimum message number as

$$\mu_m^{\min} \equiv 1, \tag{2.28}$$

and summarize this observation as the *one-message-per-cell-cycle rule* for essential gene expression.

In addition to the common floor for essential genes, there is a common gene expression distribution shape shared between organisms dependent on the message numbers, especially for low-expression essential genes. This is observed in spite of the significantly larger number of essential genes in human relative to *E. coli*. (See Fig. 2.12.) Interestingly, there is also a similarity between the non-essential gene distributions for *E. coli* and human, but not for

| Model organism | Maximum essential gene noise: $\max \mathrm{CV}_p^2$ | Estimated minimum essential gene messages /cell-cycle: $\mu_m^{\min}$ | messages /cell: $\mu_{m/c}^{\min}$ | transcription rate: $\beta_m^{\min}$ ($\mathrm{h}^{-1}$) | proteins: $\mu_p^{\min}$ |
|---|---|---|---|---|---|
| *E. coli* (LB) | 0.7 | 1 | 0.08 | 2 | 30 |
| (M9) | 0.7 | 1 | 0.03 | 0.7 | 30 |
| Yeast | 0.7 | 1 | 0.2 | 0.7 | 400 |
| Human | 0.7 | 1 | 0.6 | 0.04 | 3000 |

Table 2.2: **Estimates of threshold levels for the central dogma in three model organisms.** Estimates for the lower thresholds of transcription statistics as inferred from our analysis based on the *one-message-per-cell-cycle rule*.

yeast, which appears to have a much lower fraction of genes expressed at the lowest message numbers.

### *2.3.14 What genes fall below-threshold?*

We have hypothesized that essential genes should be expressed above a threshold value for robustness. It is therefore interesting to consider the function of genes that fall below this proposed threshold. Do functions of these genes give us any insight into essential processes that do not require robust gene expression?

Since our own preferred model system is *E. coli*, we focus here. Our essential gene classification was based on the construction of the Keio knockout library [35]. By this classification, 10 essential genes were below threshold. Our first step was to determine what fraction of these genes were also classified as essential using transposon-based mutagenesis [36, 37]. Of the 10 initial candidates, only one gene, *ymfK*, was consistently classified as an essential gene in all three studies, and we estimate that its message number is just below the threshold ($\mu_m = 0.4$). *ymfK* is located in the lambdoid prophage element e14 and is annotated as a CI-like repressor which regulates lysis-lysogeny decision [38]. In $\lambda$ phase, the CI repressor represses lytic genes to maintain the lysogenic state. A conserved function for *ymfK* is

consistent with it being classified as essential, since its regulation would prevent cell lysis. However, since *ymfK* is a prophage gene, not a host gene, it is not clear that its expression should optimize host fitness, potentially at the expense of phage fitness. In summary, closer inspection of below-threshold essential genes supports the threshold hypothesis.

### *2.3.15 Maximum noise for essential genes.*

The motivation for hypothesizing a minimum threshold for message number was noise-robustness, or the existence of a hypothesized noise ceiling above which essential gene expression is too noisy to allow robust cellular proliferation. With the *one-message-per-cell-cycle rule*, $\mu_m^{min} \equiv 1$, we can estimate the essential gene noise ceiling using Eq. 2.19:

$$\mathrm{CV}_p^2 \leq 0.7, \tag{2.29}$$

for essential genes. Since noise depends only on the message number, we expect to observe the same limit in all organisms if the message number floor is conserved.

### *2.3.16 Estimating the floor on central-dogma parameters.*

If message number floor is conserved, a limit can be estimated for the floor value on other transcriptional parameters. Using Eq. 2.21, we can estimate the floor on the cellular message number (as measured in RNA-Seq measurements):

$$\mu_{m/c}^{\min} = \tfrac{\tau_m}{T}, \tag{2.30}$$

for essential genes. Similarly, we can use Eq. 2.16 to estimate the minimum transcription rate:

$$\beta_m^{\min} = \tfrac{1}{T}, \tag{2.31}$$

for essential genes. Again, this result has an intuitive interpretation as the one-message-per-cell-cycle rule. Finally, we can estimate a floor on essential protein abundance, assuming a

constant translation efficiency using Eq. 2.18:

$$\mu_p^{\min} = \varepsilon, \tag{2.32}$$

for essential genes, where $\varepsilon$ is the translation efficiency (which we will assume is well approximated by the mean in the context of the estimate). All four floor estimates for each model organism are shown in Tab. 2.2.

## *2.4 Discussion*

### *2.4.1 Noise by the numbers.*

Although there has already been significant discussion of the scaling of biological noise with protein abundance [13, 19, 20, 1, 2], our study is arguably the first to test the predictions of the telegraph and statistical noise models against absolute measurements of protein and message abundances. This approach is particularly important for the message number ($\mu_m$), which determines the magnitude of the noise in protein expression, and facilitates direct comparisons of noise between organisms as well as identifying the common distributions of message number for genes, that are conserved from bacteria to human.

### *2.4.2 Noise scaling in E. coli versus yeast.*

A key piece of evidence for the significance of the message number was the observation of the non-canonical scaling of the yeast noise with protein abundance (Fig. 2.9); however, the canonical model (Eq. 2.3) does accurately describe the noise in *E. coli* (see Fig. 2.3). Why does the noise scale differently? In *E. coli*, the translation efficiency is only weakly correlated with the gene expression [33], and therefore the canonical model is a reasonable approximation. However, we also argued that translation efficiency should grow with expression level. Why is this not observed in *E. coli*? Due to the high noise floor in *E. coli*, nearly all essential genes are expressed at a sufficiently high expression level such that the noise is dominated

by the noise floor [2]. As a consequence, increasing the message number, while decreasing translation efficiency, does not decrease the noise even as it increases the metabolic load as a result of increased transcription. (A closely related point has recently been made in *Bacillus subtilis* [39], where Deloupy *et al.* report that the noise cannot be tuned by adjusting the message number due to the noise floor.) Our expectation is therefore that other bacterial cells will look similar to *E. coli*: They will have a higher noise floor and a similar scaling of noise with protein abundance.

In contrast, due to the lower noise floor, we expect eukaryotic cells to optimize the central dogma processes like yeast and as a result will have a similar non-canonical scaling of noise with protein abundance. Although this non-canonical scaling is clear from the abundance data (Fig. 2.9), there is an important qualification to emphasize: the mechanism that gives rise to the non-canonical scaling is due to the correlation between translation efficiency and transcription. Regulatory changes that effect only transcription (*i.e.* increase $\mu_m$) and not translation ($\varepsilon$) should obey the canonical noise model (Eq. 2.3). This scenario may help explain why Bar-Even *et al.* claim to observe canonical noise scaling in yeast [20], studying a subset of genes under a range of conditions resulting in differential expression levels. The failure of the canonical noise model (Eq. 2.3) at the proteome level in yeast (Eq. 2.25) is a consequence of genome-wide optimization of the relative transcription and translation rates.

### *2.4.3 Essential versus non-essential genes.*

What genes are defined as *essential* is highly context specific [40]. It is therefore important to consider whether the comparison between these two classes of genes is informative in the context of our analysis. We believe the example of *lac* operon in *E. coli* is particularly informative in this respect. The genes *lacZYA* are conditionally essential: they are required when lactose is the carbon source; however, these genes are repressed when glucose is the carbon source. Our expectation is that these conditionally essential genes will obey the one-message-per-cell-cycle rule when these genes are required; however, they need not obey

this rule when the genes are repressed. By analyzing essential genes, we are limiting the analysis to transcriptionally-active genes, whereas the non-essential category contains both transcriptionally-active and silenced genes.

### *2.4.4 Protein degradation and transcriptional bursting.*

Two important mechanisms can act to significantly increase the noise above the levels we predict: protein degradation and transcriptional bursting. Although the dominant mechanism of protein depletion is dilution in *E. coli*, protein degradation plays an important role in many organisms, especially in eukaryotic cells [41, 42]. If protein degradation depletes proteins faster than dilution, the shape parameter decreases below our estimate (Eq. 2.16), increasing the noise. Likewise, the existence of transcriptional bursting, in which the chromatin switches between transcriptionally active and quiescent periods, can also act to increase the noise [11, 43, 17]. Since the presence of both these mechanisms increases the noise beyond what is predicted by the message number, they do not affect our estimate of the minimum threshold for $\mu_m$.

### *2.4.5 The biological implications of noise*

What are the biological implications of gene expression noise? Many important proposals have been made, including bet-hedging strategies, the necessity of feedback in gene regulatory networks, *etc* [11]. Our analysis suggests that noise influences the optimal function of the central dogma process generically. Hausser *et al.* have already discussed some aspects of this problem and use this approach to place coarse limits on transcription versus translation rates [15]. The transcriptional floor for essential genes that we have proposed places much stronger limits on the function of the central dogma.

Although we describe our observations as a floor, a more nuanced description of the phenomenon is a common distribution of gene message numbers, peaked at roughly 15 messages per cell cycle and cutting off close to one message per cell cycle. Does this correspond to a

hard limit? We expect that this does not since there are a small fraction of genes, classified as essential, just below this limit; however, it does appear that virtually all essential genes have optimal expression levels above this threshold. The common distribution of message number clearly suggests that noise considerations shape the function of the central dogma for virtually all genes.

### *2.4.6 Adapting the central dogma to increased cell size and complexity*

Although core components of the central dogma machinery are highly-conserved, there has been significant complexification of both the transcriptional and translational processes in eukaryotic cells [44]. Given this increased regulatory complexity, it is unclear how the central dogma processes should be adapted in larger and more complex cells. An important clue to this adaptation comes from *E. coli* proliferating with different growth rates. Although there are very significant differences between the cellular message number as well as the overall transcription rate under the two growth conditions, there is very little difference in message number. In short, roughly the same number of messages are made during the cell cycle, but they are made more slowly under slow growth conditions.

How does this picture generalize in eukaryotic cells? Although both the total number of messages and the number of essential and non-essential genes are larger by orders of magnitude in both yeast and human cells, the distribution of the message number per gene is essentially the same as *E. coli* (Fig. 2.12). The conservation of the message number between organisms is consistent with all of these organisms being optimized with respect to the same trade-off between economy and robustness to noise.

## ***2.5 Conclusion: Noise is governed by transcription***

In this chapter, we have developed a framework to describe gene expression noise in both prokaryotes and eukaryotes. Using our framework, we confirm that the gamma distribution describes protein number at steady-state, and as a result, message number quantitatively

predicts noise. From data, it is observed that noise in *E. coli* has canonical scaling, while noise in yeast obeys non-canonical scaling due to the coupling between translation efficiency and transcription. Finally, we observe the one message rule describing a conserved transcriptional program where various organisms minimize the transcription per cell cycle. The following chapter explores the significance and rationale for the noise described by the mathematical framework, and makes predictions about fundamental principles that shape the central dogma.

# Chapter 3

# PROTEINS ARE OVERABUNDANT!

This chapter is a modified reproduction of ref. [45].

## *3.1 Author summary*

Now that we have established a mathematical framework describing gene expression noise, we create a model which predicts cellular growth robustness to the noise. For example, *E. coli* has over 600 essential proteins that each need to be expressed above certain minimum threshold levels to avoid failure of essential processes. Our model explores the tradeoff between growth robustness to noise, and increase in metabolic load from producing extra protein. A critical input to our model is an asymmetric single-cell fitness landscape, where the growth rate drops sharply when an essential protein falls below a minimum level, whereas the growth rate decreases slowly as protein number increases due to an increase in the metabolic load. A biologist might expect that the cell is a highly evolved organism, and over millions of years of generations has optimized the level of each protein needed. However, from the previous chapter, we know that gene expression is noisy! As a consequence, the overall protein expression level for the population of cells should be much higher than the minimum threshold level–overabundance.

## *3.2 Introduction*

What rationale determines the optimal transcription and translation level of a gene in the cell? Protein expression levels optimize cell fitness [46, 47]: Too low of an expression level of essential proteins slows growth by compromising the function of essential processes [48, 49], whereas the overexpression of proteins slows growth by increasing the metabolic

load [50]. This trade-off naïvely predicts that the cell maximizes its fitness by a Goldilocks principle in which cells express just enough protein for function [51]; however, achieving growth robustness is nontrivial, since all processes at the cellular scale are stochastic, including gene expression [11]. This biological noise leads to significant cell-to-cell variation in protein numbers, even for essential proteins that are required for growth [1, 2]. The optimal expression program must therefore ensure robust expression of hundreds of distinct essential gene products. This chapter explores the consequences of growth robustness on the central dogma regulatory program.

## *3.3 Results*

### *3.3.1 Defining the RLTO Model.*

To study the consequences of growth robustness on gene expression quantitatively, we propose and analyze a minimal model: the Robustness-Load Trade-Off (RLTO) Model. The model includes three critical components: (i) Protein levels are stochastic and the single-cell growth rate depends upon them, (ii) gene transcription and translation generate a metabolic load, and (iii) cell growth is dependent on a large number of essential genes. These model characteristics result in a highly-asymmetric fitness landscape. The optimization of expression on this asymmetric landscape predicts new phenomenology absent from previous models (*e.g.* [15]).

The protein number $N_p$ expressed from gene $i$ is the product of two sequential stochastic processes: transcription and translation [24], leading to cell-to-cell variation in protein number, which we will refer to as *noise*. In our analysis, we will model gene expression using the canonical steady-state noise model [13]. (See Fig. 3.1A.) In this model, the numbers of proteins $N_p$ for gene $i$ are predicted to be gamma-distributed [19], in close agreement with observation [2]. The distribution is described by two gene-specific statistical parameters: *message number* ($\mu_m$), defined as the mean number of messages transcribed per cell cycle for gene $i$, and the *translation efficiency* ($\varepsilon$), the mean number of proteins translated from each

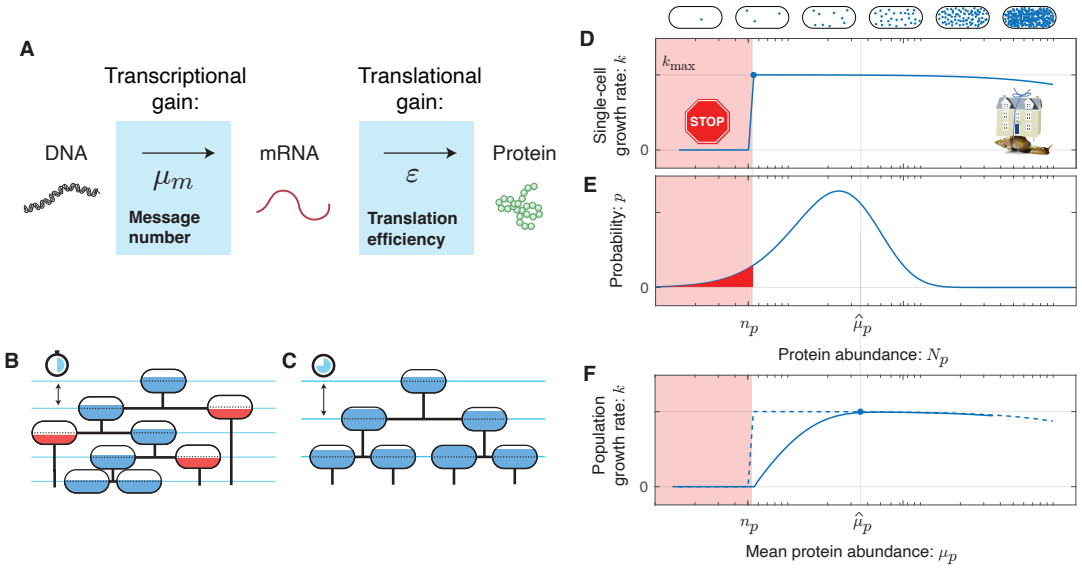



Figure 3.1: **The RLTO Model. Panel A: Gene expression processes are stochastic.** The central dogma describes a two-stage stochastic process where genes are first transcribed into $\mu_m$ messages per cell cycle, then translated to $\varepsilon$ proteins per message, on average. **Panel B & C:** A schematic cell lineage tree is shown during exponential growth. For a specific protein $i$, the cell fill represents protein number $N_p$ relative to its threshold number $n_p$ required for cell growth. **Panel B:** Reducing the mean expression level reduces doubling time; however, expression noise results in below-threshold cells (red fill) which arrest. **Panel C:** Increasing protein expression increases the doubling time. All cells are above threshold (blue fill). **Panel D: The fitness landscape is asymmetric.** Growth arrests for protein number $N_p$ smaller than the threshold level $n_p$ (red) due to the failure of essential processes. High expression levels decrease growth rate due to increasing metabolic cost. The relative metabolic cost of overabundance is small relative to the cost of growth arrest due to the large size of the total metabolic load $N_0$. **Panel E: Gene expression is stochastic.** There is significant cell-to-cell variation in protein abundance ($N_p$) around the mean level ($\mu_p$). Due to this noise, some cells fall below threshold (red). The distribution in protein number is modeled using a gamma distribution. **Panel F: The robustness-load tradeoff determines the optimal expression level.** The population growth rate depends on the distribution of the protein number. The asymmetry of the fitness landscape drives the optimal expression level far above the threshold level due to the high fitness cost of low protein abundance.

message transcribed for gene $i$. The mean protein abundance is their product: $\mu_p = \mu_m \varepsilon$. These parameters can be expressed in terms of ratios of the rates of the underlying gene expression processes, as described in Supplementary Material Sec. 3.7.1.

How should the effect of essential protein expression on growth rate be modeled in the context of the RLTO model? Much recent work has focused on cellular resource allocation to functional sectors (*e.g.* [52]). In this approach, an optimization is performed by the *coordinated* modulation of the abundance of all proteins in a particular sector, leading to a trade-off between functional capacities of the cell. However, in the RLTO model, the optimization is fundamentally different: We consider the *uncoordinated* modulation of the abundance of protein species $i$ due to noise. For these incoherent changes, we generically expect proteins to exhibit rate-limited kinetics: Increases in the protein number $N_p$ above a threshold level $n_p$ has minimal effect on the rate since other chemical species (proteins, metabolites, *etc.*) are rate limiting [53]. However, if the protein number $N_p$ falls below the threshold $n_p$, then protein species $i$ becomes rate limiting and leads to a significant slowdown of the growth rate. In the RLTO model, we coarse-grain the details of this growth slowdown as growth arrest. (See Fig. 3.1.) There is already some precedent for the use of this type of threshold (*e.g.* [54]), but we will demonstrate that the detailed form of the fitness landscape is not important. (See Supplementary Material Sec. 3.8.) Although sufficiently detailed knowledge of the relevant molecular and cellular biology could be used to predict the protein thresholds $n_p$, we will treat these as gene-specific unknown parameters.

As shown in Materials and Methods, the relative cellular fitness with respect to the expression of gene $i$ can be computed by combining the fitness losses associated with robustness (Eq. 3.6) and metabolic load (Eq. 3.8):

$$\frac{\Delta k}{k_0} = -(\Lambda + \tfrac{\varepsilon}{N_0})\mu_m - \tfrac{1}{\ln 2}\gamma(\tfrac{\mu_m}{\ln 2}, \tfrac{n_p}{\varepsilon \ln 2}), \tag{3.1}$$

where the first term represents fitness loss due to metabolic load of transcription and translation while the second term represents loss due to the arrest of essential processes. $\gamma$ is the

regularized incomplete gamma function and the central distribution function (CDF) of the gamma distribution. (See Supplementary Material Sec. 3.7.1 and 3.7.2.) In summary, the model depends only on a single global parameter: the relative metabolic load $\Lambda$ and three gene-specific parameters: the threshold number $n_p$, the message number $\mu_m$ and the relative translation efficiency $\varepsilon/N_0$. We propose that the cell is regulated to maximize the growth rate with respect to transcription (message number) and translation (translation efficiency). The fitness landscape predicted by the RLTO model for representative parameters is shown in Fig. 3.2A.

### *3.3.2 RLTO predicts protein overabundance.*

The optimal regulatory program ($\mu_m$ and $\varepsilon$ values) can be predicted analytically. They depend on only a single global parameter, the relative load $\Lambda$, and the gene-specific threshold number $n_p$. Since the threshold number is not directly observable experimentally, we will instead predict the optimal overabundance $o$, defined as the ratio of the mean protein number to the threshold number:

$$o \equiv \mu_p/n_p. \tag{3.2}$$

As shown in Fig. 3.2C, the RLTO model generically predicts that the optimal protein fraction is overabundant ($o > 1$); however, the overabundance is not uniform for all proteins. For highly-transcribed genes ($\mu_m \gg 1$) like ribosomal genes, the overabundance is predicted to be quite small ($o \approx 1$); however, for message numbers approaching unity, the overabundance is predicted to be extremely high ($o \gg 1$). At a quantitative level, the relation between optimal overabundance and message number depends on the relative load ($\Lambda$), but its phenomenology is qualitatively unchanged over orders of magnitude variation in $\Lambda$.

### *3.3.3 Understanding the rationale for overabundance*

To explore both the robustness of the protein overabundance prediction and to understand its mathematical rationale, we explored a collection of more complex models numerically.

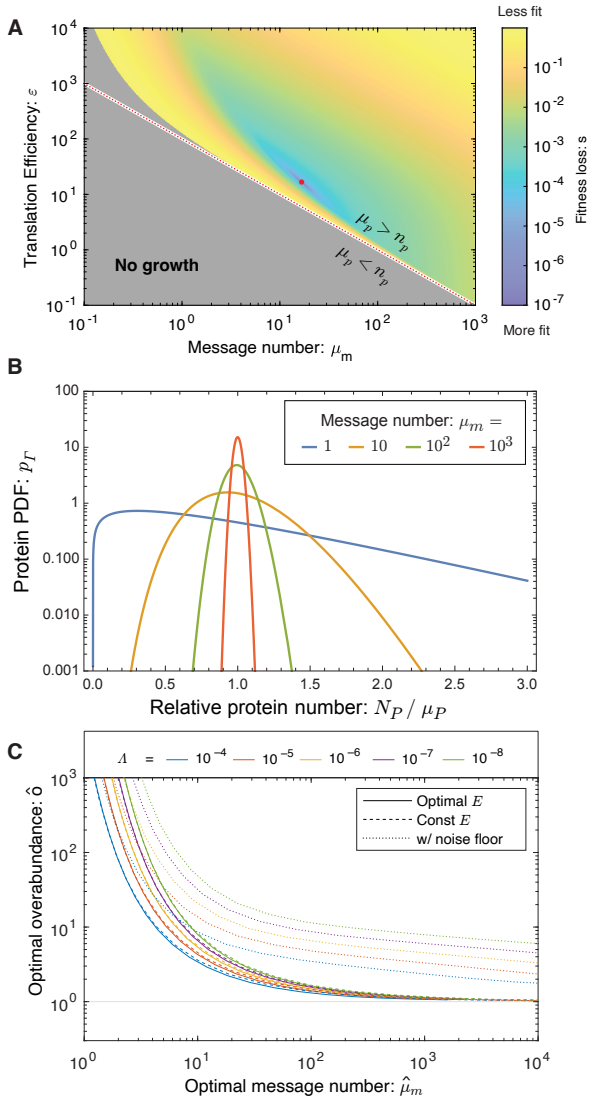



Figure 3.2: **The RLTO model predicts overabundance is optimal. Panel A: Fitness landscape determines optimal message number and translation efficiency.** The fitness loss ($s \equiv \ln k_{\max}/k$) is shown as a function of message number ($\mu_m$) and translation efficiency ($\varepsilon$). The red dotted curve represents programs where the mean protein number is equal to the threshold ($\mu_p = n_p$) and the red dot represents the optimal regulatory program ($\hat{\mu}_m$, $\hat{\varepsilon}$). **Panel B: Gene-expression noise.** Due to the stochasticity of gene expression processes at equilibrium, the protein number $N_p$ is gamma-distributed [2]. For high-expression genes, expression has low noise and the protein number is tightly distributed around its mean; however, for low-expression genes, expression is noisy and the distribution is extremely wide. **Panel C: Overabundance is optimal for all genes.** For high-expression genes, low overabundance is optimal ($\mu_p \approx n_p$); however, for low-expression genes, vast overabundance is optimal ($\mu_p \gg n_p$). From a quantitative perspective, overabundance depends on the relative load $\Lambda$; however, the qualitative dependence is invariant to over an orders-of-magnitude range of values.

The key mathematical feature that drives overabundance is not the assumption of growth arrest, but rather the strong asymmetry of the fitness landscape: the high cost of protein *underabundance* and the low cost of protein *overabundance*. (See Fig. 3.1EF.) The population growth rate (Panel F) can be understood qualitatively as the convolution of the single-cell growth rate (Panel D) with the probability density function (PDF) of the protein abundance (Panel E). In the RLTO model, this asymmetry is parameterized by the relative load ($\Lambda$), defined as the relative metabolic cost of transcribing an additional message. Since we estimate that $\Lambda < 10^{-5}$, this cost is very low relative to the total metabolic cost of the cell, therefore we expect this asymmetry, and the prediction of the RLTO model, to be robust.

### *3.3.4 Overabundance is observed in a range of experiments*

The RLTO Model predicts that all essential proteins are overabundant. In general, the RLTO model predicts that protein numbers have very significant robustness (*i.e.* buffering) to protein depletion. Although this result is potentially surprising, it is in fact consistent with many studies. For instance, Belliveau *et al.* have recently analyzed the abundance of a wide range of metabolic and other essential biological processes, and conclude that protein abundance appears to be in significant excess of what is required for function [51]. Likewise, CRISPRi approaches have facilitated the characterization of essential protein depletion. The qualitative results from these experiments are consistent with overabundance: Large-magnitude protein depletion is typically required to generate strong phenotypes [48, 55, 56]. In particular, Peters *et al.* engineered a complete collection of CRISPRi essential-gene depletion constructs in *Bacillus subtilis*. Importantly, when dCas9 is constitutively expressed, these constructs deplete essential proteins about three-fold below their endogenous expression levels [48]; however, roughly 80% grew without measurable fitness loss in log-phase growth despite the depletion. When grouped by functional category, only ribosomal proteins were found to have statistically significant reductions in fitness [48]. As shown in Fig. 3.2C, the RLTO model predicts that all but the highest expression proteins are expected to show

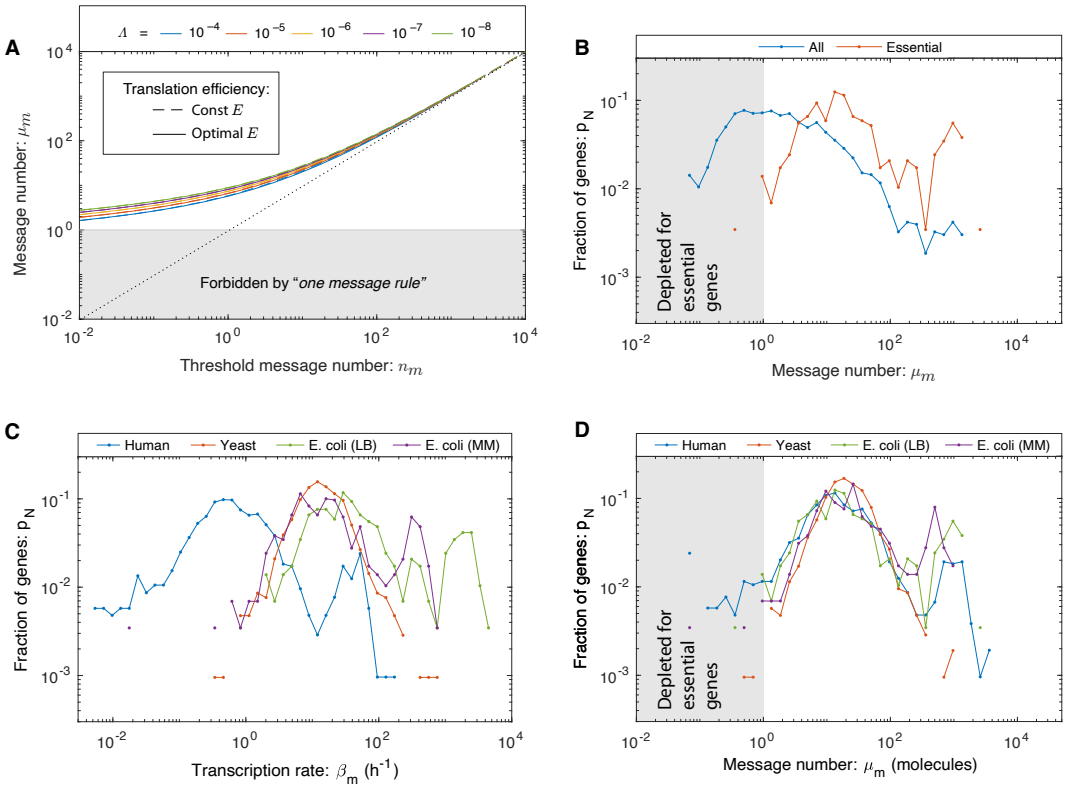



Figure 3.3: **A lower threshold for transcription: The one message rule. Panel A: RLTO predicts the one-message rule.** For high-expression genes, overabundance is low and the message number $\mu_m$ is predicted to be comparable to the threshold level $n_m$ (dotted line); however, for low-expression genes there is a lower threshold ($\mu_m \geq 1$) below which expression is too noisy for robust growth. The threshold is weakly dependent on relative load $\Lambda$. **Panel B: A one-message threshold is observed in *E. coli* for essential genes.** A histogram shows the distribution of gene message numbers for all genes (blue) versus essential genes (orange). As predicted by the RLTO model, virtually all essential genes are expressed above the one-message-per-cell-cycle threshold. **Panel C: The distribution of transcription rates for essential genes.** No alignment is observed between the distributions of transcription rates in three evolutionarily-divergent organisms. For instance, the per gene transcription rate is significantly lower in human cells relative to *E. coli*. **Panel D: The distribution of message numbers for essential genes in three evolutionarily-divergent organisms.** The alignment of distributions of message number per gene between human, yeast, and *E. coli* (under two distinct growth conditions) reveals a nontrivial commonality between central dogma regulatory programs. We propose that the rationale for this alignment is the one-message rule that predicts that all essential genes must be expressed above one message per cell cycle. Both yeast and *E. coli* come very close to satisfying this proposed threshold; however, a greater proportion of genes in human break the one-message threshold. We speculate that this is due in part to the *ad hoc* nature of the essential-gene classification in the context of complex multicellular organisms.

minimal fitness reductions in response to a three-fold depletion of essential enzymes. The optimality of protein overabundance explains the paradox of protein expression levels being simultaneously optimal [46] and in excess of what is required for function [51, 48, 49, 56]. Although this qualitative picture of essential protein overabundance is clear, there has yet to be a quantitative and detailed measurement of protein overabundance, and in particular, an analysis of the relationship between protein overabundance and message number.

### *3.3.5 RLTO predicts a one-message transcription threshold*

The RLTO model predicts protein overabundance, but is there a clear transcriptional signature? To analyze this question, we define the message threshold $n_m \equiv \mu_m/o$. (This parameterization is convenient since it is independent of the translation efficiency.) We can then analyze the relation between optimal message number and threshold message number, as shown in Fig. 3.3A. The model predicts that even for genes that have extremely small threshold message numbers (*e.g.* $n_m = 10^{-2}$), the optimal message number stays above one message transcribed per cell cycle. Qualitatively, expressing messages below this level is simply too noisy even for proteins needed at the lowest expression levels. (See the blue curve in Fig. 3.2B corresponding to the protein number distribution of $\mu_m = 1$.) The model therefore predicts a lower floor on transcription for essential genes of one message per cell cycle.

### *3.3.6 A lower threshold is observed for message number.*

To identify a putative transcriptional floor, we first analyzed the transcriptome in *Escherichia coli*. We hypothesize that cells must express essential genes above the one-message threshold for robust growth. The distinction between essential and nonessential genes is critical in this context, since nonessential genes can be inducibly expressed. For instance, in *E. coli*, the *lac* operon is repressed in the absence of lactose and therefore need not satisfy

the one-message threshold. The transcriptional threshold is only hypothesized to apply to genes whose products are required to maintain cell fitness under the measured conditions.

We generated histograms for *E. coli* growing rapidly on rich media for these two classes of genes. The message numbers for *nonessential* genes are widely distributed, with a significant fraction of genes falling below the one-message threshold; however, only one *essential* gene is expressed below the one-message threshold (0.3% of essential genes). (See Fig. 3.3B.) The threshold is not sharp, but rather a smooth depletion relative to a median of 18 messages per cell cycle. This observation is consistent with the predictions of the RLTO model.

To further test this prediction, we then analyzed *E. coli* transcription under slow-growth conditions. Since these cells are less transcriptionally active, we hypothesized that this analysis would constitute a more stringent test of the one-message rule. To our surprise, although the transcription rate is indeed reduced in slow growth, the essential gene message numbers still satisfy the one-message rule (with a two gene exception, 0.7%), again consistent with the predictions of the RLTO model. (See Fig. 3.3D.)

Next, we analyzed eukaryotic transcriptomes in *Saccharomyces cerevisiae* (yeast) and *Homo sapiens* (human). For yeast, there is a well-defined notion of essential genes [57]. As predicted, yeast essential genes obey the one-message threshold (with two exceptions, 0.2%). (See Fig. 3.3D.) The interpretation is less clear-cut in human cells: An essential gene classification has been generated in the context of proliferation in cell culture [58]. In order to try to capture a generic picture, we average the human transcriptome of cell types. We find that the vast majority of essential genes obey the one-message rule; however, there are significantly more genes that break the rule (81 genes, 8%) than in the other organisms.

### *3.3.7 Message number distribution is conserved*

To what extent is this human data consistent with the RLTO model? For human cells, our test of the one-message rule is too simplistic in two respects: (i) We ignore the significant transcriptional differences associated with distinct cell types and (ii) the essential gene

classification itself is defined by the ability of mutants or knockdowns to proliferate in cell culture; in marked contrast to the *in vivo* context where cell proliferation is tightly regulated [58]. Due to these subtleties, we decided to take a complementary approach: We considered the distribution of three different transcriptional statistics for each gene: transcription rate, cellular message number, defined as the average number of messages instantaneously, and message number ($\mu_m$), defined as the number of messages transcribed in a cell cycle. The RLTO model predicts a one-message threshold with respect to message number, but not the other two statistics. We therefore predict that the message number distributions in each organism (*E. coli*, yeast, and human) should align for low expression genes with respect to message number, but not for the other two transcriptional statistics. Consistent with the predictions of the RLTO model, there is a striking alignment of message number for essential genes between all three model organisms and growth conditions for message number. (See Fig. 3.3D.) This alignment is non-trivial: It is not observed with respect to other transcriptional statistics (Fig. 3.3C).

What is the significance of the similarity in the distributions of message number between organisms? Another strategy for satisfying the one message rule would be for transcription to be increased. For instance, mammalian cells have about 1000 times the number of proteins relative to bacterial cells. (See Tab. 3.2.) One might therefore naively predict that the message number should be increased 1000-fold as well. This is not observed. In fact, the message number distributions of all the model organisms analyzed abut the one message threshold. The proximity to the threshold suggests that organisms do as little transcription as possible while satisfying the one message rule. This appears to be a conserved transcriptional regulatory strategy from *E. coli* to human.

### *3.3.8 Translation efficiency is predicted to increase with transcription.*

What does the RLTO model predict about how the cell should balance the gene expression process between transcription and translation? Minimizing transcription (at fixed protein

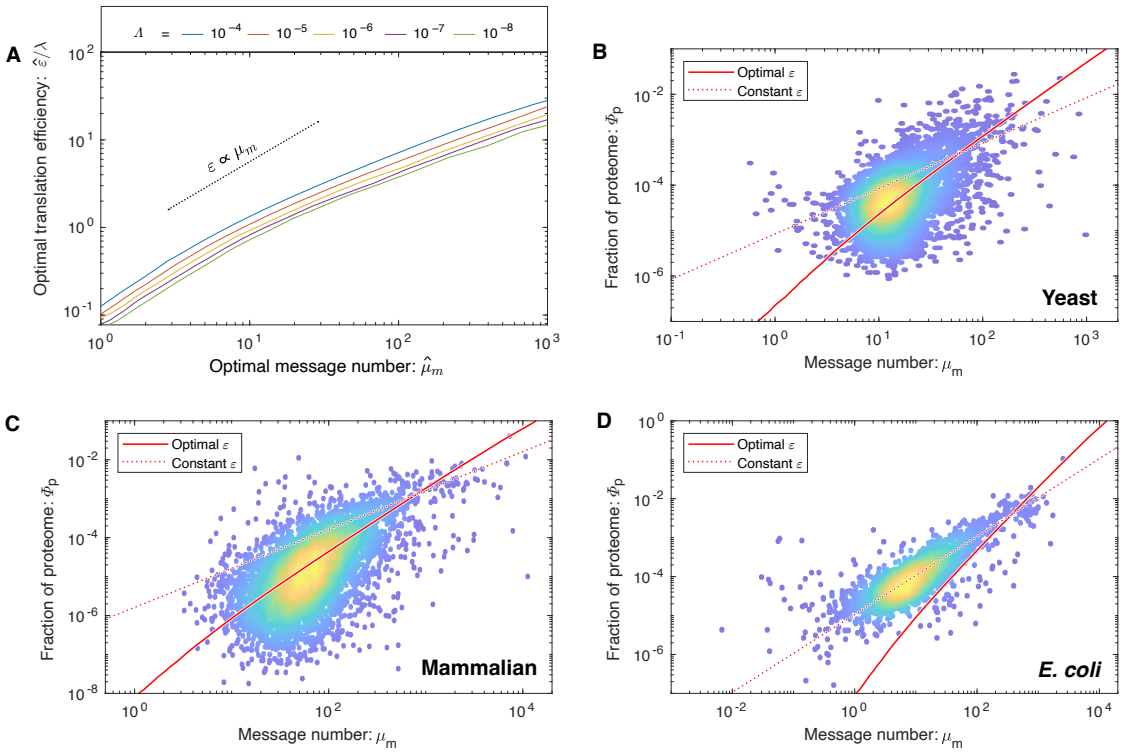



Figure 3.4: **How are transcription and translation balanced? Panel A: The RLTO model predicts load balancing.** The ratio of the optimal translation efficiency ($\hat{\varepsilon}$) to the message cost ($\lambda$) is roughly independent of the relative load ($\Lambda$). The translation efficiency $\varepsilon$ is predicted to be roughly proportional message number $\mu_m$. **Panel B: RLTO predicts the protein-message-abundance relation in yeast.** The observed proteome fraction is compared to two models: the RLTO optimal model (solid red line) and constant-translation-efficiency model (dotted red line). Both models make parameter-free predictions. The RLTO optimum predicts the global trend. (Data from Ref. [4].) **Panel C: Mammalian proteome fraction.** The RLTO prediction (solid) is superior to the constant-translation-efficiency prediction (dashed). **Panel D: *E. coli* proteome fraction.** In contrast, the constant-translation-efficiency prediction (dashed) is superior to RLTO prediction (solid).

abundance) reduces the metabolic load; however, it decreases robustness. Growth rate maximization balances these two costs. Quantitatively, the maximization of the growth rate (Eq. 3.1) with respect to the translation efficiency can be performed analytically, predicting the optimal translation efficiency, shown in Fig. 3.4A. We provide an exact expression in the Supplementary Material Sec. 3.7.4; however, an approximate expression for the translation efficiency is more clearly interpretable:

$$\hat{\varepsilon} \approx 0.1\lambda\hat{\mu}_m. \tag{3.3}$$

The optimal translation efficiency has two important qualitative features for central dogma regulation. The first prediction is that as the message cost ($\lambda$) rises, the optimal translation efficiency ($\hat{\varepsilon}$) increases in proportion while the message number decreases. We present evidence for this prediction in the Supplementary Material Sec. 3.7.11.

The second prediction is that the optimal translation efficiency is also approximately proportional to message number ($\hat{\varepsilon} \propto \mu_m$). Therefore, the RLTO model predicts that low expression levels should be achieved with low levels of transcription and translation, whereas high-expression genes are achieved with high levels of both. We call this relation between optimal transcription and translation the *load balancing principle*. The most direct test of load balancing is measuring the protein-message abundance relation. Due to load balancing, the RLTO model predicts protein number (and proteome fraction) to scale like:

$$\hat{\mu}_p \propto \hat{\mu}_m^2, \tag{3.4}$$

whereas a constant-translation-efficiency model has linear scaling ($\mu_p \propto \mu_m$). Computing proteome fraction, rather than protein number, results in a parameter-free prediction. (See Supplemental 3.7.12.)

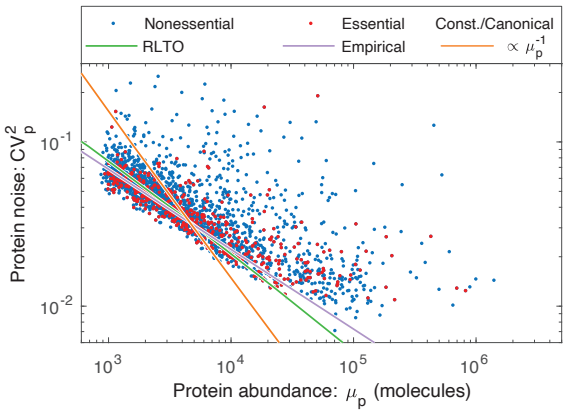



Figure 3.5: **RLTO predicts the magnitude of noise in yeast.** The observed gene expression noise is yeast is shown for essential and nonessential genes. Two protein-message abundance models are compared to the data: The RLTO model (green) versus the constant-translation-efficiency (canonical model, orange). The RLTO model predicts both the magnitude of the noise, as well as its scaling with protein abundance. The reduced slope of the RLTO model is the consequence of load balancing, which reduces the noise for the noisiest, low-expression genes. (Data from Ref. [1].)

### *3.3.9 Load balancing is observed in eukaryotic cells.*

To test the RLTO predictions, we compare observed proteome measurements in three evolutionarily divergent species, *E. coli* [33], yeast [4] and mammalian cells [59], to two models: the RLTO and the constant-translation-efficiency models. The results of the parameter-free predictions are shown in Fig. 3.4BCD for each organism. The RLTO model clearly captures the global trend in the proteome-fraction message-number relation in eukaryotic cells and a direct fit to a power law with an unknown exponent is consistent with Eq. 3.4 (Supplementary Material 3.10.4).

In *E. coli*, the constant-translation-efficiency model better describes the data. Why does this organism appear not to load balance? In the supplementary material, we demonstrate that the observed translation efficiency is consistent with the RLTO model, augmented by a ribosome-per-message limit. Hausser *et al.* have proposed just such a limit, based on the ribosome footprint on mRNA molecules [15]. (See Supplementary Material 3.7.17.)

Although this augmented model is consistent with central dogma regulation in *E. coli*, it is not a complete rationale. This proposed translation-rate limit could be circumvented by increasing the lifetime of *E. coli* messages, which would increase the translation efficiency. A more in-depth analysis specific to *E. coli* is needed to understand why the observed message lifetime is so short.

### *3.3.10 RLTO model predicts observed noise in yeast.*

Although the protein fraction measurements support the RLTO predictions for the translation efficiency in eukaryotic cells, these measurements do not provide a compelling rationale for why load balancing maximizes the growth rate. To understand its rationale, we explore its implications for noise.

In a typical biological context, $\mu_m \ll \varepsilon$ and as a result, noise production is dominated by the transcription step of the gene expression process [13, 19]. (A table of central dogma parameters for each model organism appears in the Supplementary Material Tab. 3.2.) Quantitatively, the canonical steady-state noise model predicts that the noise should be inversely related to the message number [13, 19]:

$$\mathrm{CV}_p^2 = \frac{\ln 2}{\mu_m}, \tag{3.5}$$

however, it is the relation between mean protein abundance $\mu_p$ and noise ($\mathrm{CV}_\mathrm{p}^2$) which is typically reported [2, 1]. Based on the scaling of the optimal translation efficiency with the message number in eukaryotic cells (Eq. 3.3), we find the protein number to scale with message number (Eq. 3.4), which predicts that noise should scale with protein abundance $\mathrm{CV}_\mathrm{p}^2 \propto \mu_\mathrm{p}^{-1/2}$ in yeast (see Supplementary Material 3.10.1); however, due to the observed absence of translation-efficiency scaling in bacteria, the noise should scale as $\mathrm{CV}_\mathrm{p}^2 \propto \mu_\mathrm{p}^{-1}$ in bacteria, as observed [2]. Does the yeast noise show the predicted scaling? The parameter-free RLTO noise prediction closely matches the observed noise in both magnitude and scaling, as shown in Fig. 3.5.

### *3.3.11 Reducing noise is the rationale for load balancing.*

This noise analysis also provides a conceptual insight into the rationale for load balancing. The load balanced (RLTO-green) and constant-translation-efficiency (orange) predictions for the noise are shown in Fig. 3.5. Load balancing results in decreased noise for low-expression, noisy genes over what is achieved with constant translation efficiency. This decreased noise is predicted to increase growth robustness. In principle, the noise could be reduced further by tipping the balance even more towards transcription; however, the RLTO model predicts that this approach is too metabolically costly, and the optimal strategy is that observed for noise scaling in yeast.

## *3.4 Discussion*

### *3.4.1 What are the biological implications of noise?*

Many important proposals have been made, including bet-hedging strategies, the necessity of feedback in gene regulatory networks, *etc.* [11]. Our model suggests that overcoming cell-to-cell variation may fundamentally reshape the metabolic budget: Typically, proteins constitute 50-60% of the dry mass of the cell [50] and therefore overabundance could increase the overall protein budget by a significant factor. Why does the cell tolerate this significant increase in metabolic load above what would be predicted by a resource allocation analysis (*e.g.* [60])? This strategy dramatically reduces the consequence of stochastic expression of proteins on the rate of single-cell proliferation.

A second source of stochasticity, environmental fluctuations, has been proposed as a rationale for overabundance [61], especially in the context of metabolic genes [62]. In short, cells express protein to hedge against starvation [61] or changes in the carbon source, *etc.* [62]. How does this hypothesis compare to our growth robustness hypothesis? There are some similarities between these environmental-fluctuation models and the RLTO model: In both models, it is a fluctuations-based mechanism that drives overabundance; however, there are important distinctions between the model predictions. In the environmental fluctuation

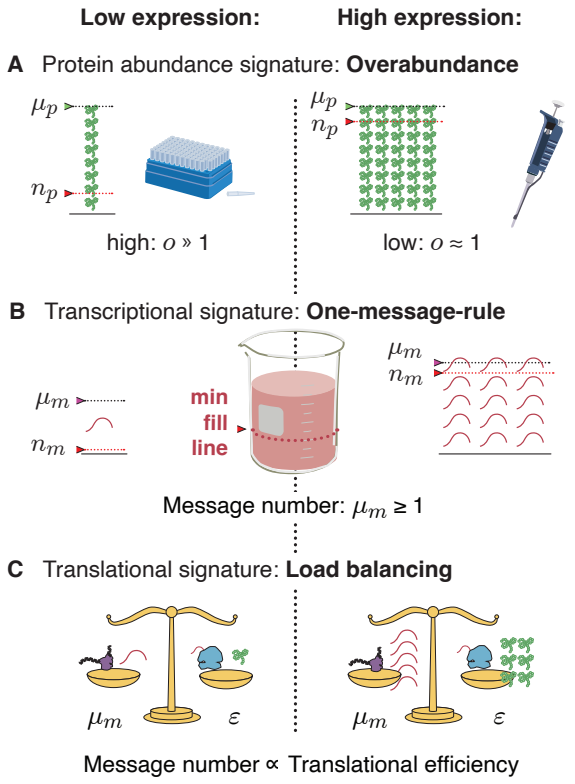



Figure 3.6: **Central dogma regulatory principles. Panel A: Overabundance.** Low-expression essential genes are expressed with high overabundance; whereas, high-expression essential genes are expressed with low overabundance. Lab supply analogy: Low-cost items that are used stochastically (*e.g.* pipette tips) are purchased in great excess, while the higher cost items that are less stochastic (*e.g.* pipette) are purchased as needed. **Panel B: One-message rule.** Robust expression of essential genes requires them to be transcribed above a threshold of one message per cell cycle. **Panel C: Load balancing.** In eukaryotic cells, optimal fitness is achieved by balancing transcription and translation: The optimal message number is proportional to the optimal translation efficiency. High (low) expression levels are achieved by high (low) levels of transcription followed by high (low) levels of translation per message.

model, there is a trade-off between log-phase fitness and the rapidity of adaptation [61]; whereas in the RLTO model, overabundance corresponds to the log-phase optimum. Organisms experiencing prolonged periods of balanced growth would therefore be expected to reduce overabundance. Furthermore, the environmental fluctuation model most naturally explains overabundance for proteins related to metabolic processes, whereas the RLTO model predicts overabundance generically, dependent only on message number, which appears to be much more consistent with experiments exploring essential-protein depletion [49].

### *3.4.2 Implications for nonessential genes.*

In our analysis, we have focused on essential genes in order to motivate the growth-threshold in the RLTO model. To what extent do nonessential genes share the same optimization? In support of the proposal that RLTO optima describe nonessential genes is the success of the model in predicting the translation efficiency for all genes, not just essential genes. (See Fig. 3.4 and Fig. 3.5.) Furthermore, the definition of a gene as *essential* depends on context: For instance, in the context of *E. coli* growth on lactose, the gene *lacZ* is essential, although it is nonessential on other carbon sources [63]. Under growth conditions where the *lacZ* gene is essential, we predict that LacZ should be overabundant, consistent with observation [62]. Finally, our modeling suggested that RLTO model phenomenology is the results of asymmetry of the cost of under versus overabundance. For nonessential genes whose activity significantly increases fitness, we still expect fitness asymmetry due to the low relative metabolic cost of increased expression. We therefore expect all gene products, most especially those with low expression, to be overabundant, under conditions where their activity increases fitness.

### *3.4.3 Implications of overabundance for inhibitors.*

The generic nature of overabundance, especially for low-expression proteins, has important potential implications for the targeting of these proteins with small-molecule inhibitors (*e.g.*

drugs). For the highest expression proteins, like the constituents of the ribosome, relatively small decreases in the active fraction (*e.g.* a three-fold reduction) are expected to lead to growth arrest [48]. This may help explain why inhibitors targeting translation make such effective antimicrobial drugs. However, we predict that the lowest expression proteins require a much higher fraction of the protein to be inactivated, with the lowest-expression proteins expected to need more than a 100-fold depletion. This predicted robustness makes these proteins much less attractive drug targets [64].

### *3.4.4 The principles that govern central dogma regulation.*

We propose that robustness to noise fundamentally shapes the central dogma regulatory program for all genes and predicts a number of key regulatory principles. (See Fig. 3.6.) For high-expression genes, load balancing implies that gene expression consists of both high-amplification translation and transcription. The resulting expression level has low overabundance relative to the threshold required for function. In contrast, for essential low-expression genes, a three-fold strategy is implemented: (i) overabundance raises the mean protein levels far above the threshold required for function, (ii) load balancing, and (iii) the one-message rule ensures that message number is sufficiently large to lower the noise of inherently-noisy, low-expression genes. We anticipate that these regulatory principles, in particular protein overabundance, will have important implications, not only for our understanding of central dogma regulation specifically, but for understanding the rationale for protein expression level and function in many biological processes.

## ***3.5 Methods***

### *3.5.1 RLTO model.*

The effect of stochastic cell arrest can be implemented analytically as follows: The probability of growth is the probability that all essential proteins are above threshold, $P_+$. The

population growth rate $k$ is [65]:

$$\frac{k}{k_0} = 1 + \frac{1}{\ln 2} \ln P_+, \tag{3.6}$$

for a population of cells subject to stochastic arrest with probability $1-P_+$ per cell cycle where $k_0$ is the growth rate of the non-arrested cells. For each gene $i$, the canonical steady-state noise model predicts the protein number CDF in terms of message number $\mu_m$ and translation efficiency $\varepsilon$ [13]. Assuming the below-threshold probability is small, the probability that the cell is below threshold for gene $i$ is:

$$\ln P_{+,i} = -\gamma(\tfrac{\mu_m}{\ln 2}, \tfrac{n_p}{\varepsilon \ln 2}), \tag{3.7}$$

where $\gamma$ is the regularized incomplete gamma function and the CDF of the gamma distribution. (See Supplementary Material Sec. 3.7.2.)

While protein underabundance slows cell growth by the arrest of essential processes, protein overabundance slows growth by increasing the metabolic load. To implement the metabolic-load contribution to cell fitness, we use a minimal model that realizes the metabolic cost of both transcription and translation that is analogous to those previously used in the context of resource allocation (*e.g.* [52]). The metabolic load of transcription and translation of gene $i$ is:

$$\frac{k}{k_0} = 1 - \frac{\lambda+\varepsilon}{N_0}\mu_m, \tag{3.8}$$

where $k_0$ is the growth rate in the absence of the metabolic load of gene $i$, $N_0$ is the total cellular metabolic load, and $\lambda$ is the metabolic message cost. (See the Supplementary Material Sec. 3.7.2 for a detailed development of the model.) The $\lambda$-term represents the metabolic cost of transcription and the $\varepsilon$-term represents the metabolic cost of translation of gene $i$. We define the relative load as $\Lambda \equiv \lambda/N_0$ as the ratio of the metabolic load of a single message to the total metabolic cost of the cell. In *E. coli*, we estimate that $\Lambda$ is roughly $10^{-5}$ and it is smaller still for eukaryotic cells.

## 3.6 Conclusion: The cell produces protein in overabundance for growth robustness to noise

In this chapter, we developed the Robustness-Load Trade-Off (RLTO) model. First, we input the trade-off from the probability of cell arrest and the metabolic load to compute the relative growth rate. The model predicts large overabundance for low expression proteins (subject to high noise, with low cost to produce in excess) and small overabundance for high expression proteins (subject to low noise, with high cost to produce in excess). Next, upon optimizing growth rate as a function of message number in the model, we again find the one message rule predicted from the noise framework. Finally, upon optimizing growth rate as a function of translation efficiency, the model predicts translation efficiency to change with transcription, agreeing with our transcription-translation load-balancing principle from the previous chapter. In the following chapter, I discuss a pipeline tool used to analyze timelapses of cellular growth. Imaging experiments involving the knockout of essential genes were performed to demonstrate overabundance; however, essential gene knockouts often result in unusual cell morphologies. The advances introduced by the pipeline tool allowed for the quantification of these mutant cells and provided evidence for the overabundance of essential proteins.

## 3.7 Supplemental Material: Detailed development of the RLTO model

In this section, we provide a detailed development of the RLTO model. First, we describe the stochastic kinetic model for the central dogma, which introduces key quantities for the RLTO model (Sec. 3.7.1). Next, we provide a derivation of the growth rate as a function of the model parameters (Sec. 3.7.2) as well as other methods (Secs. 3.7.4, 3.7.12, 3.7.15, and 3.7.18). For each of the results discussed in the main paper, we provide more detailed analyses, which include both supplemental results (Secs. 3.7.8, 3.7.9, 3.7.11, 3.7.13, and 3.7.16) that support the story described in the main paper, as well as supplemental discussions (Secs. 3.7.3, 3.7.5, 3.7.6, 3.7.7, 3.7.14, and 3.7.17).

### *3.7.1 Methods: Detailed description of the noise model*

*Stochastic kinetic model for the central dogma.*

The canonical steady-state noise model for the central dogma describes multiple steps in the gene expression process [13, 19, 2]: Transcription generates mRNA messages. These messages are then translated to synthesize the protein gene products [24]. Both mRNA and protein are subject to degradation and dilution [25]. At the single cell level, each of these processes are stochastic. We will model these processes with the stochastic kinetic scheme [24]:

$$\begin{array}{ccccc} \text{DNA} & \xrightarrow{\beta_m} & \text{mRNA} & \xrightarrow{\beta_p} & \text{Protein} \\ & & \gamma_m \downarrow & & \gamma_p \downarrow \\ & & \varnothing & & \varnothing, \end{array} \tag{3.9}$$

where $\beta_m$ is the transcription rate ($\text{s}^{-1}$), $\beta_p$ is the translation rate ($\text{s}^{-1}$), $\gamma_m$ is the message degradation rate ($\text{s}^{-1}$), and $\gamma_p$ is the protein effective degradation rate ($\text{s}^{-1}$). The message lifetime is $T_m \equiv \gamma_m^{-1}$. For most proteins in the context of rapid growth, dilution is the dominant mechanism of protein depletion and therefore $\gamma_p$ is approximately the growth rate [26, 27, 2]: $\gamma_p = T^{-1} \ln 2$, where $T$ is the doubling time.

*Statistical model for protein abundance.*

To study the stochastic dynamics of gene expression, we used a stochastic Gillespie simulation [28, 29]. (See Sec. 3.7.1.) In particular, we were interested in the explicit relation between the kinetic parameters $(\beta_m, \gamma_m, \beta_p, \gamma_p)$ and experimental observables. This framework was largely developed in the first chapter.

Consistent with previous reports [13, 19], we find that the distribution of protein number per cell (at cell birth) was described by a gamma distribution: $N_p \sim \Gamma(a, \theta)$, where $N_p$ is the protein number at cell birth and $\Gamma$ is the gamma distribution which is parameterized by a scale parameter $\theta$ and a shape parameter $a$. (See Sec. 3.7.1.) We refer to this distribution as the *canonical steady-state noise model*; The relation between the four kinetic parameters

and these two statistical parameters has already been reported, and have clear biological interpretations [19]: The scale parameter:

$$\theta = \varepsilon \ln 2, \tag{3.10}$$

is proportional to the translation efficiency:

$$\varepsilon \equiv \frac{\beta_p}{\gamma_m}, \tag{3.11}$$

where $\beta_p$ is the translation rate and $\gamma_m$ is the message degradation rate. $\varepsilon$ is understood as the mean number of proteins translated from each message transcribed. The shape parameter $a$ can also be expressed in terms of the kinetic parameters [19]:

$$a = \frac{\beta_m}{\gamma_p}; \tag{3.12}$$

however, we will find it more convenient to express the scale parameter in terms of the cell-cycle message number:

$$\mu_m \equiv \beta_m T = a \ln 2, \tag{3.13}$$

which can be interpreted as the mean number of messages transcribed per cell cycle. Forthwith, we will abbreviate this quantity *message number* in the interest of brevity.

*Gene expression statistics.*

Assuming the gamma-distributed model, the statistics of the protein distribution can be written in terms of the gamma distribution parameters. The mean protein number is:

$$\mu_p = a\theta = \mu_m \varepsilon, \tag{3.14}$$

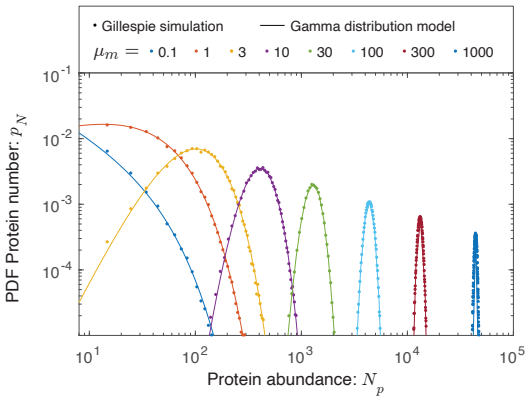



Figure 3.7: **The protein abundance is approximately gamma distributed.** Protein abundance was modeled for eight different transcription rates using a Gillespie simulation, including the stochastic partitioning of the proteins between daughter cells at cell division. The range in abundance matches the observed range of expression levels in the cell. We observed that the simulated protein abundances were well fit by gamma distributions.

or the product of the message number and translation efficiency. The variance in protein number is:

$$\sigma_p^2 = a\theta^2 = \mu_m \varepsilon^2 \ln 2, \tag{3.15}$$

and the coefficient of variation is:

$$\begin{aligned} \mathrm{CV}_p^2 &\equiv \frac{\sigma_p^2}{\mu_p^2}, & (3.16)\\ &= \frac{1}{a} = \frac{\ln 2}{\mu_m}, & (3.17) \end{aligned}$$

which depends on the message number alone.

*Gillespie simulation of stochastic kinetic scheme*

Protein distributions based on the kinetic scheme defined in Sec. 3.7.1 were simulated with a Gillespie algorithm, with specific parameter values for *E. coli*. Assuming the lifetime of the cell cycle ($T = 30$ min) [30], mRNA lifetime ($T_m = 2.5$ min) [31], and translation rate

($\beta_p \approx 500$ hr$^{-1}$), the protein distributions for several mean expression levels were numerically generated for exponential growth with 100,000 stochastic cell divisions, with protein partitioned at division following the binomial distribution.

The gamma distributions for each mean message number with scale and shape parameters determined by the corresponding translation efficiency and message number ($\theta = \varepsilon \ln 2$, $a = \frac{\mu_m}{\ln 2}$) as used for the Gillespie simulation were also plotted with the protein distributions. We observe an excellent match between these Gillespie simulations and the canonical steady-state statistical noise model (*i.e.* gamma function) as shown in Fig. 3.7.

*Gamma function and distribution conventions*

There are a number of conflicting conventions for the gamma function and distribution arguments. We will use those defined on Wikipedia. The gamma distributed random variable $X$ will be written:

$$X \sim \Gamma(a, \theta), \tag{3.18}$$

where $a$ is the shape parameter and $\theta$ is the scale parameter. The PDF of the distribution is:

$$p_\Gamma(x|a,\theta) \equiv \tfrac{x^{a-1}}{\theta^a \Gamma(a)} e^{-x/\theta}, \tag{3.19}$$

where $\Gamma(a)$ is the gamma function. The CDF is therefore:

$$\begin{aligned} P_\Gamma(x|a,\theta) &= \int_0^x \mathrm{d}x'\ p_\Gamma(x'|a,\theta), && (3.20)\\ &= P_\Gamma(\tfrac{x}{\theta}|a,1), && (3.21)\\ &= \int_0^{x/\theta} \mathrm{d}x'\ \tfrac{x^{a-1}}{\Gamma(a)} e^{-x}, && (3.22)\\ &= \gamma(a, x/\theta), && (3.23) \end{aligned}$$

where $\gamma$ is the regularized incomplete gamma function.

### *3.7.2 Methods: The derivation of the RLTO growth rate*

*The metabolic load of protein and the resource allocation model*

To model the effect of the metabolic load on cell growth, we will expand on a model used by Hwa and co-workers [52]. For conciseness, we will call the original model the *resource allocation* model.

Consider a cell where the total number of proteins in the cell is $N$. The synthesis of these proteins requires two sets of processes: (i) the metabolic processes responsible for synthesizing the precursors (*i.e.* amino acids, *etc*) and (ii) the translation process. Proteins involved in the metabolic processes be referred to as the $P$ sector and number $N_P$. The proteins involved in the translational process will be referred to as the $R$ sector and number $N_R$. In addition to the P and R sectors, there is a third Q sector with protein number $N_Q$. The total protein number per cell is therefore:

$$N = N_R + N_P + N_Q. \tag{3.24}$$

The proteome fractions are defined $\Phi_X \equiv N_X/N$ and have the normalization condition:

$$1 = \Phi_R + \Phi_P + \Phi_Q. \tag{3.25}$$

The key assumption in the resource allocation model is that the abundances of the R and P sectors can change in size to accommodate changes in the nutrient quality and translation load associated with a particular growth condition [52]. In contrast, the Q sector has a fixed proteome fraction, $\Phi_Q$, irrespective of growth conditions. In the resource allocation model, the size of these adjustable R and P sectors are chosen to optimize the growth rate $k$.

The condition for balanced growth requires that the overall protein output of the translation process match the growth rate:

$$kN = k_R N_{*R}, \tag{3.26}$$

where $k_R$ is the effective translation rate per protein and $N_{*R}$ is the number of productive R sector proteins, which is subset of the total number $N_R$:

$$N_R = N_{*R} + N_{0R}, \tag{3.27}$$

and $N_{0R}$ represents unproductive R sector protein. We can rewrite Eq. 3.26 in terms of the proteome fraction:

$$k = k_R(\Phi_R - \Phi_{0R}). \tag{3.28}$$

In the expression above, we will assume that the parameters $k_R$ and $\Phi_{0R}$ are fixed, but the total fraction $\Phi_R$ is chosen to optimize the growth rate $k$.

For any productive sectors $i$, we will write analogous equations to Eq. 3.28 linking sector fraction size $\Phi_i$ to function:

$$k = k_i(\Phi_i - \Phi_{0i}), \tag{3.29}$$

where, as before, $\Phi_{0i}$ represents a fixed-size fraction of unproductive protein.

To determine the unknown optimum growth rate and sector sizes, Eq. 3.29 can be re-written:

$$\Phi_{*i} = \Phi_i - \Phi_{0i} = \tfrac{k}{k_i}, \tag{3.30}$$

and then summed over all sectors (excluding Q):

$$\Phi_* = k \sum_{i \neq Q} k_i^{-1}, \tag{3.31}$$

where we define the total productive fraction of the proteome:

$$\Phi_* \equiv \sum_{i \neq Q} \Phi_{*i} = 1 - \Phi_0, \tag{3.32}$$

and $\Phi_0$ represents the total fraction of unproductive protein:

$$\Phi_0 = \Phi_Q + \sum_{i \neq Q} \Phi_{0i}, \tag{3.33}$$

including the entire Q sector. Since $\Phi_Q$ and the $\Phi_{0i}$ are all assumed to be fixed, Eq. 3.31 determines the growth rate. (In *E. coli*, Hwa and coworkers estimate that $\Phi_* \approx 0.55$.)

To understand the meaning of Eq. 3.31, we first define an ideal growth rate as

$$k_{\text{ideal}}^{-1} = \sum_{i \neq Q} k_i^{-1}, \tag{3.34}$$

which would be the growth rate in the absence of unproductive protein; however, due to the presence of the unproductive protein, the growth rate is proportional to the productive fraction:

$$k = k_{\text{ideal}} \Phi_*. \tag{3.35}$$

The optimal protein fractions can be determined using Eq. 3.30:

$$\Phi_i = \tfrac{k_{\text{ideal}}}{k_i} \Phi_* + \Phi_{0i}. \tag{3.36}$$

How does the growth rate change when the unproductive protein fraction is changed by $\delta\Phi$? The productive fraction is reduced:

$$\Phi_* \to \Phi'_* = \Phi_* - \delta\Phi. \tag{3.37}$$

The ratio of the new growth rate $k'$ to the original is therefore:

$$\tfrac{k'}{k} = 1 - \tfrac{\delta\Phi}{\Phi_*}. \tag{3.38}$$

Note that our generalized resource allocation model is written for arbitrary number of functional sectors $i \neq Q$ and the key determinant of the change in growth rate is the fraction of productive protein $\Phi_*$. This equation will be used to model the fitness cost of the metabolic load.

*The metabolic load of mRNA*

What is the cost of transcription? It is perhaps useful to first consider the estimates of biosynthetic cost of macromolecules in the cell in descending order in *E. coli* [12]:

| **Macromolecule** | **Biosynthetic cost** ($10^9$ ATP) |
|---|---|
| Protein | 4.5 |
| Phospholipid | 3.2 |
| RNA | 1.6 |
| Lipopolysaccharide | 3.8 |
| DNA | 0.35 |
| Peptidoglycan | 0.17 |
| Glycogen | 0.03 |

So clearly the cost of RNA is itself not insignificant. Although a significant fraction of the RNA is rRNA rather than mRNA, the mRNA itself in *E. coli* undergoes multiple rounds of transcription due to its short lifetime, increasing its cost to what is required to synthesize the molecules observed in the *E. coli* cell at any time $t$. Furthermore, transcription is dependent on protein enzymes, which themselves must be synthesized. We therefore conclude from this estimate that the cost of transcription is likely a significant determinant of the metabolic load.

Experimentally, Kafri and coworkers have measured the fitness cost of transcription and translation independently using the DAmP (Decreased Abundance by mRNA Perturbation) system in *Saccharomyces cerevisiae* [66]. As expected, they report that the metabolic cost

of transcription is comparable to translation and that the reduction in growth rate is linear in transcription, in close analogy to Eq. 3.38.

*Metabolic load in the RLTO model*

To produce a minimal model to study the trade-off between robustness and metabolic load, we must consider both the metabolic cost of transcription and translation. We will write that the metabolic load (in protein equivalents) associated with gene $i$ is:

$$\delta N_i = \lambda \mu_{m,i} + \mu_{p,i}, \tag{3.39}$$

where $\lambda$ is the message cost, the metabolic load associated with an mRNA molecule relative to a single protein molecule of the gene product. $\mu_{m,i}$ is the mean number of messages transcribed per cell cycle (mRNA molecules per cell cycle) for gene $i$. $\mu_{p,i}$ is the mean number of protein translated per cell cycle for gene $i$. We will describe the mean protein number in terms of the translation efficiency $\varepsilon_i$, the number of proteins translated per message:

$$\mu_{p,i} = \varepsilon_i \mu_{m,i}. \tag{3.40}$$

How does the cell growth rate change due to the metabolic load associated with the expression of gene $i$? The change in the metabolic load is:

$$\delta \Phi_i = \frac{\delta N_i}{N}, \tag{3.41}$$

where $N$ represents the total metabolic load of all components of the cell, in units of protein equivalents. Using Eq. 3.38, the resulting change in growth rate is:

$$\frac{k}{k_0} = 1 - \frac{(\lambda+\varepsilon_i)\mu_{m,i}}{\Phi_* N}, \tag{3.42}$$

where $k_0$ is the growth rate in the absence of the metabolic load of gene $i$.

In our analysis, the exact size of the total metabolic load $N$ will not be important. In the interest of simplicity we will therefore adsorb the productive fraction $\Phi_*$ into an effective total metabolic load:

$$N_0 \equiv \Phi_* N, \tag{3.43}$$

and write a concise relation between the load from gene $i$ and the growth rate:

$$\frac{k}{k_0} = 1 - \frac{(\lambda+\varepsilon_i)\mu_{m,i}}{N_0}. \tag{3.44}$$

This equation has an intuitive interpretation: growth slows in proportion to the relative added metabolic load. Since $\Phi_*$ is order unity, we will ignore the distinction between the $N$ and $N_0$ quantities hence forth.

Although the global parameters $N_0$ and $\lambda$ provide an intuitive representation of the model, the relative growth rate depends on fewer parameters. Let $k$ and $k_0$ be the growth rates in the presence and absence of the metabolic load of gene $i$. The relative growth rate is:

$$\frac{k}{k_0} = 1 - (\Lambda + E_i)\mu_{m,i}, \tag{3.45}$$

where we have introduced two new reduced parameters: the relative load, defined as $\Lambda \equiv \lambda/N_0$, represents the ratio of the metabolic load of a single message to the total load and the relative translation efficiency, defined $E_i \equiv \varepsilon_i/N_0$, which is the ratio of the number of proteins translated per message to the total metabolic load $N_0$. (Note that due to the high multiplicity $N_0 \gg (\lambda + \varepsilon_i)\mu_{m,i}$, we can ignore the distinction between $N_0$ and $N_0'$ in the denominator.) If we neglect the difference between the total metabolic load and the number of proteins, the proteome fraction for gene $i$ is:

$$\Phi_i = E_i \mu_{m,i}. \tag{3.46}$$

Both reduced parameters, $\Lambda$ and $E_i$ are extremely small. In *E. coli*, we estimate that both $\Lambda$ and $E_i$ are roughly $10^{-5}$ and they are smaller still for eukaryotic cells. (See Sec. 3.7.18.)

*Growth rate with stochastic arrest*

For completeness, we provide a derivation of the growth rate with stochastic cell-cycle arrest that we have previously described [65]. Starting from the exponential mean expression for the population growth rate [65]:

$$k = \frac{\ln 2}{\overline{T}}, \tag{3.47}$$

where $k$ is the population growth rate and

$$\overline{T} \equiv -\frac{1}{k} \ln \mathbb{E}_T \exp(-kT), \tag{3.48}$$

is the exponential mean, where $T$ is the stochastic cell cycle duration and $\mathbb{E}$ is the expectation operator [65]. We take a coarse-grained model which considers changes in growth rate due to fluctuations in protein number to be negligible. Note that Eq. 3.48 is equivalent to the *Euler-Lotka* equation [67, 68].

Let $P_+$ be the probability of growth. When the cells are growing, the cell cycle duration $\tau$ is determined by the metabolic load predictions (Eq. 3.44). The probability mass function is therefore:

$$p_T(t) = \begin{cases} P_+, & t = \tau \\ (1 - P_+), & t \to \infty \end{cases}. \tag{3.49}$$

Evaluating the expectation in Eq. 3.48 gives:

$$\overline{T} = -k \ln P_+ + \tau. \tag{3.50}$$

Using Eq. 3.47, we can solve for the growth rate $k$:

$$k = \tau^{-1} \ln(2P_+). \tag{3.51}$$

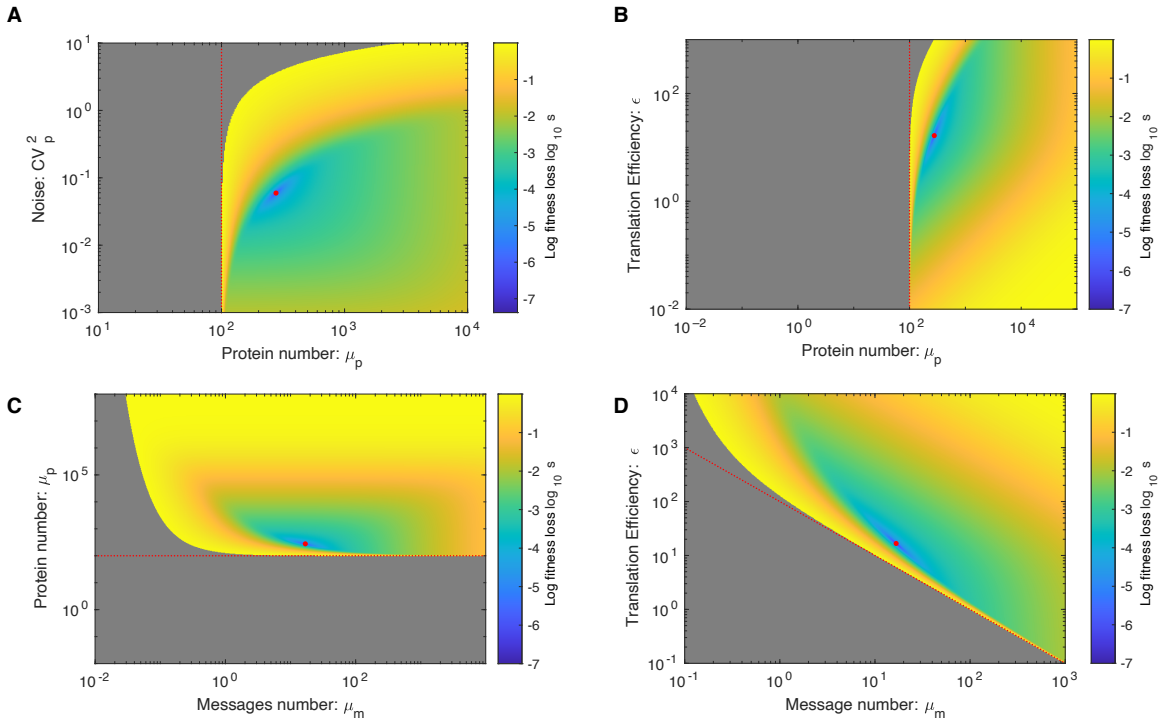



Figure 3.8: **Four perspectives on the fitness landscape.** In each landscape, the red circle represents the fitness optimum. The red dotted line represents the mean protein number equal to the protein threshold $n_p = 10^2$. Here, fitness is quantified by the log growth rate: $s = \ln k_{\max}/k$. **Panel A: Mean protein number versus noise. Panel B: Mean protein number versus translation efficiency. Panel C: Mean message versus protein numbers. Panel D: Message number versus translation efficiency.**

As expected, the growth rate goes down as the probability of growth $P_+$ decreases, stopping completely at $P_+ = \frac{1}{2}$. We can then compute the ratio of the growth with $(k)$ and without arrest $(k_0)$:

$$\frac{k}{k_0} = 1 + \frac{1}{\ln 2} \ln P_+. \tag{3.52}$$

where $k_0$ is computed by evaluating Eq. 3.51 at $P_+ = 1$.

*RLTO growth rate*

In the RLTO model, we will assume the probability of growth is the probability that all essential protein numbers are above threshold. We will further assume that each protein number is independent, and therefore:

$$P_+ = \prod_{i\in\mathcal{E}} \Pr\{N_{p,i} > n_{p,i}\}, \tag{3.53}$$

where $\mathcal{E}$ is the set of essential genes. Clearly, this assumption of independence fails in the context of polycistronic messages. We will discuss the significance of this feature of bacterial cells elsewhere, but we will ignore it in the current context.

As we will discuss, the probability of arrest of any protein $i$ to be above threshold is extremely small. It is therefore convenient to work in terms of the CDFs which are very close to zero:

$$\begin{aligned} \ln P_+ &= \sum_{i\in\mathcal{E}} \ln\left(1 - \Pr\{N_{p,i} < n_{p,i}\}\right), && (3.54)\\ &\approx -\sum_{i\in\mathcal{E}} \Pr\{N_{p,i} < n_{p,i}\}, && (3.55)\\ &= -\sum_{i\in\mathcal{E}} \gamma(\tfrac{\mu_{m,i}}{\ln 2}, \tfrac{n_{p,i}}{\varepsilon_i \ln 2}), && (3.56) \end{aligned}$$

where $\gamma$ is the regularized incomplete gamma function and the CDF of the gamma distribution (see 3.7.1 and 3.7.1).

*When the ln approximation is avoided...*

The approximation discussed in the previous section is extremely well justified at the optimal central dogma parameters; however, there are a set of figures where we cannot use it. In the fitness landscape figures (Fig. 3.8), we compute the fitness not just at the optimal values but far from them. When cell arrest has a large effect on the growth rate, we cannot

approximate the natural log with a series expansion, and we must use the full expression in Eq. 3.54.

*Single-gene equation*

By summing the fitness losses from the metabolic load and cell arrest (Eqs. 3.44, 3.52, and 3.56), we can write an expression for the growth rate including contributions from essential gene $i$:

$$\begin{aligned} \frac{k}{k_0} &= 1 - \frac{1}{N_0}(\lambda + \varepsilon_i)\mu_{m,i} + ... \\ &\quad -\frac{1}{\ln 2}\gamma(\frac{\mu_{m,i}}{\ln 2}, \frac{n_{p,i}}{\varepsilon \ln 2}), \end{aligned} \tag{3.57}$$

where the second term on the RHS represents the fitness loss due to the metabolic load and the third term represents the fitness loss due to stochastic cell arrest due to protein $i$ falling below threshold. The fitness landscape for different gene expression parameters is shown from four different perspectives in Fig. 3.8. From this point forward, we will drop the subscript $i$ for the sake of brevity unless otherwise noted.

*Summary of RLTO parameter values for figures*

The parameter values for the RLTO model used for each figure in the paper are shown in Tab. 3.1.

### *3.7.3 Discussion: The fitness landscape of a trade-off.*

The fitness landscape predicted by the RLTO model for representative parameters is shown in Fig. 3.8. The figure displays a number of important model phenomena: There is no growth for mean protein number $\mu_p$ below the threshold number $n_p$, and for high noise, $\mu_p$ must be in significant excess of $n_p$. Rapid growth can be achieved by the two mechanisms: (i) high expression levels ($\mu_p$) are required for high noise amplitude ($\mathrm{CV}_\mathrm{p}^2$) or (ii) lower expression

| Figure number and panel: | Relative load: $\Lambda$ (No units) | Protein cost: $\lambda$ (molecules$^{-1}$) | Protein threshold: $n_p$ (molecules) | Message number: $\mu_m$ (molecules/cell cycle) | Translation efficiency: $\varepsilon$ (No units) | Noise floor $C_0$ (No units) |
|---|---|---|---|---|---|---|
| 3.2A | $10^{-5}$ | 10 | $10^2$ | Range | Range | 0 |
| 3.2C | Range | NA | Range | Local optimum | Local optimum | 0 |
| 3.4A | Weak dependence ($10^{-5}$) | NA | Range | Local optimum | Local optimum | 0 |
| 3.4B | Weak dependence ($10^{-5}$) | NA | Range | Local optimum | Local optimum | 0 |
| 3.4C | Weak dependence ($10^{-5}$) | NA | Range | Local optimum | Local optimum | 0 |
| 3.4D | Weak dependence ($10^{-5}$) | NA | Range | Local optimum | Local optimum | 0 |
| 2.9 | Weak dependence ($10^{-5}$) | NA | Range | Local optimum | Local optimum | 0 |
| 3.8A | $10^{-5}$ | 10 | $10^2$ | Range | Range | 0 |
| 3.8B | $10^{-5}$ | 10 | $10^2$ | Range | Range | 0 |
| 3.8C | $10^{-5}$ | 10 | $10^2$ | Range | Range | 0 |
| 3.8D | $10^{-5}$ | 10 | $10^2$ | Range | Range | 0 |
| 3.9A | Range | $10^2$ | Range | Local optimum | Local optimum | 0 |
| 3.9B | Range | $10^2$ | Range | Local optimum | Local optimum | 0 |
| 3.9C | Range | $10^2$ | Range | Local optimum | Local optimum | 0 |
| 3.9D | Range | $10^2$ | Range | Local optimum | Local optimum | 0 |
| 3.10A | Range | $10^2$ | Range | Local optimum | Local optimum | 0 |
| 3.10B | Range | $10^2$ | Range | Local optimum | Local optimum | 0 |
| 3.10C | Range | $10^2$ | Range | Local optimum | Local optimum | 0 |
| 3.10D | Range | $10^2$ | Range | Local optimum | Local optimum | 0 |
| 3.11 | $10^{-5}$ | NA | Range | Local optimum | 30 | 0.1 |
| 3.12A | Range | NA | Range | Local optimum | Local optimum | 0 |
| 3.13 | Weak dependence ($10^{-5}$) | NA | Range | Local optimum | Local optimum | 0 |
| 3.15A | Range | NA | Range | Local optimum | Local optimum | 0 |
| 3.18B | Weak dependence ($10^{-5}$) | NA | Range | Local optimum | Local optimum | 0 |

Table 3.1: **RLTO model parameters by figure.** *Range* appears if a range of parameters is used. *NA* appears if the parameter value is irrelevant. *Local optimum* appears if the parameter values in optimized to maximize the growth rate.

levels coupled with lower noise. This trade-off leads to a ridge-like feature of nearly optimal models. The optimal fitness corresponds to a balance between increasing the mean protein number ($\mu_p$) and decreasing noise ($\mathrm{CV}_p^2$). This optimal central dogma program strategy leads to significant overabundance.

### *3.7.4 Methods: Central dogma optimization*

*Optimization of transcription and translation (eukaryotes)*

The relative growth rate is:

$$\frac{\Delta k}{k_0} = -(\Lambda + \tfrac{\varepsilon}{N_0})\mu_m - \tfrac{1}{\ln 2}\gamma(\tfrac{\mu_m}{\ln 2}, \tfrac{n_p}{\varepsilon \ln 2}), \tag{3.58}$$

where $\gamma$ is the regularized incomplete gamma function, which is the CDF of the gamma distribution and represents the probability of arrest due to gene $i$. (Note that this equation is identical to Eq. 3.57 but with the gene subscript $i$ implicit.) We set the partial derivative of the growth rate with respect to message number equal to zero:

$$0 = -\tfrac{\lambda+\hat{\varepsilon}}{N_0} - \tfrac{1}{(\ln 2)^2}\gamma_{,1}(\tfrac{\hat{\mu}_m}{\ln 2}, \tfrac{n_p}{\hat{\varepsilon} \ln 2}), \tag{3.59}$$

where we use the canonical comma notation to show which argument of $\gamma$ has been differentiated. Next we differentiate with respect to the translation efficiency to generate a second optimization condition:

$$0 = -\tfrac{\hat{\mu}_m}{N_0} + \tfrac{n_p}{\hat{\varepsilon}^2(\ln 2)^2}\gamma_{,2}(\tfrac{\hat{\mu}_m}{\ln 2}, \tfrac{n_p}{\hat{\varepsilon} \ln 2}). \tag{3.60}$$

We will work in the large multiplicity limit where the overall metabolic load is much smaller than the metabolic load associated with any single gene: $N_0 \gg (\lambda+\hat{\varepsilon})\hat{\mu}_m$. Next, we eliminate the threshold $n_p$ in favor of the optimal overabundance:

$$\hat{o} \equiv \tfrac{\hat{\mu}_p}{n_p} = \tfrac{\hat{\varepsilon}\hat{\mu}_m}{n_p}, \tag{3.61}$$

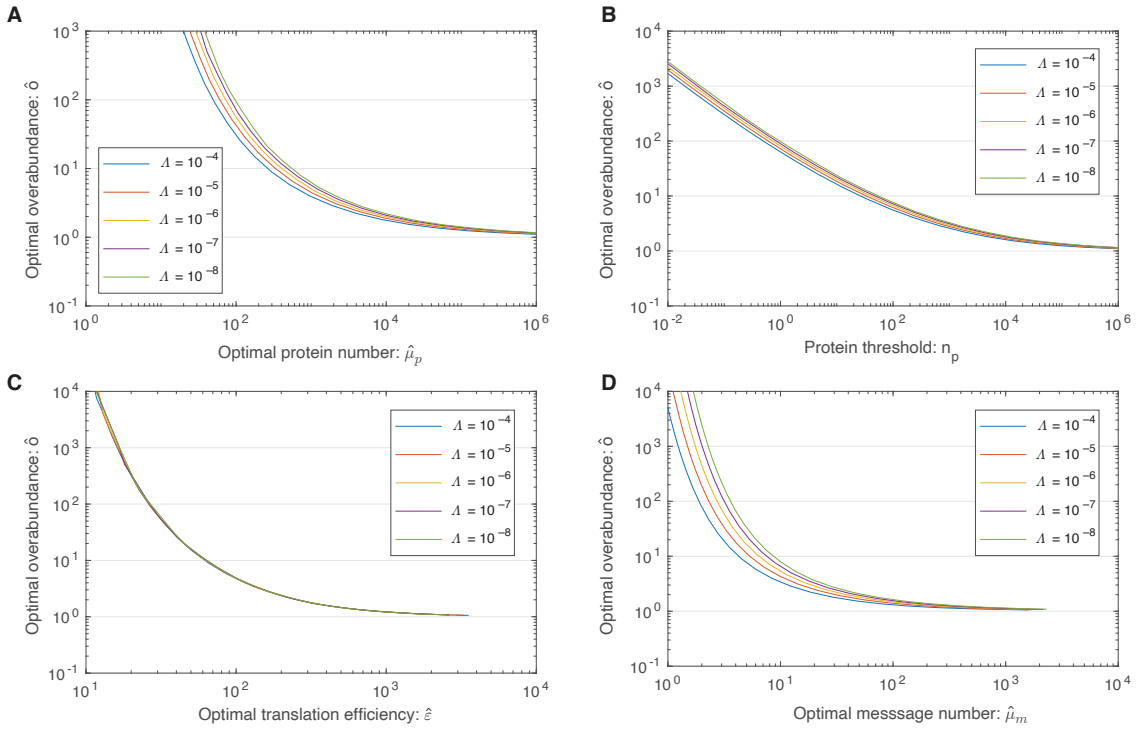



Figure 3.9: **Four perspectives on the dependence of optimal overabundance on relative load $\Lambda$.** All these calculations are performed for protein cost $\lambda = 100$ in order to give real numbers in molecules per cell. **Panel A: Overabundance as a function of protein number.** Overabundance decreases as protein number increases. These calculations are $\lambda$ *dependent*. **Panel B: Overabundance as a function of the protein threshold.** Overabundance decreases as protein threshold increases. These calculations are $\lambda$ *dependent*. **Panel C: Overabundance as a function of translation efficiency.** Overabundance decreases as translation efficiency increases. These calculations are $\lambda$ *dependent*. **Panel D: Overabundance as a function of message number.** Overabundance decreases as message number increases. These calculations are $\lambda$ *independent*.

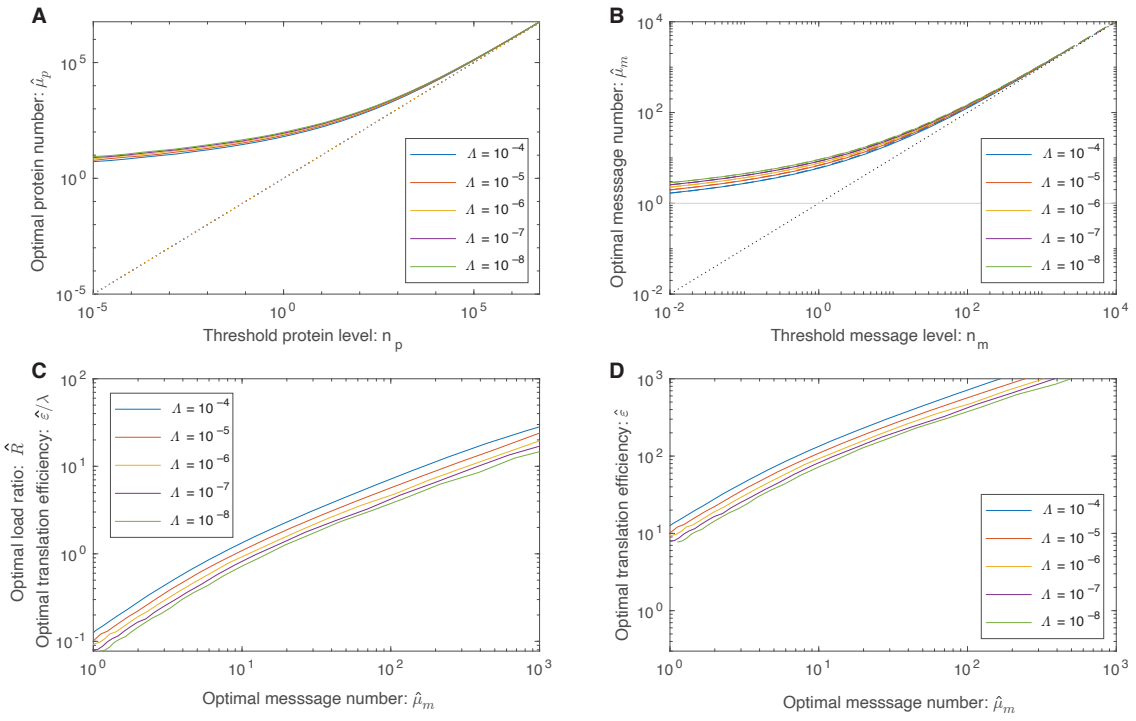



Figure 3.10: **Four perspectives on load balancing.** All these calculations are performed for protein cost $\lambda = 100$ in order to give real numbers in molecules per cell. **Panel A: Protein number versus protein threshold.** At high expression levels, the protein number tracks the protein threshold; however, the one message rule forces the protein number to threshold for low expression levels. These calculations are $\lambda$ *dependent*. **Panel B: Message number versus message threshold.** At high expression levels, the message number tracks the message threshold; however, the one message rule forces the message number to a threshold close to $\mu_m = 1$ for low expression levels. These calculations are $\lambda$ *independent*. **Panel C & D: Message number versus translation efficiency.** The optimal translation efficiency grows almost linearly with the optimal message number. The scaled translation efficiency ($\hat{\varepsilon}/\lambda$) is independent of $\lambda$ while the translation efficiency ($\hat{\varepsilon}$) is dependent on $\lambda$. The ratio $\varepsilon/\lambda$ has a second interpretation: the load ratio $R$. $R$ is defined as the metabolic cost of translation over transcription of the gene.

in both Eqs. 3.59 and 3.60. Eq. 3.60 can now be solved for the optimal translation efficiency:

$$\hat{\varepsilon} = \frac{N_0}{\hat{o}(\ln 2)^2}\gamma_{,2}(\frac{\hat{\mu}_m}{\ln 2}, \frac{\hat{\mu}_m}{\hat{o}\ln 2}). \tag{3.62}$$

If we reinterpret $\gamma$ as the CDF of the gamma distribution, we can rewrite this equation in terms of the gamma distribution PDF:

$$\hat{\varepsilon} = \frac{N_0}{\ln 2}p_\Gamma(\mu_m|\frac{\hat{\mu}_m}{\ln 2}, \hat{o}\ln 2), \tag{3.63}$$

which will be the optimization equation for the translation efficiency.

To derive the optimization condition for the message number $\mu_m$, we substitute Eq. 3.62 into Eq. 3.59:

$$\frac{\lambda \ln 2}{N_0} = -\frac{1}{\hat{o}\ln 2}\gamma_{,2}(\frac{\hat{\mu}_m}{\ln 2}, \frac{\hat{\mu}_m}{\hat{o}\ln 2}) - \frac{1}{\ln 2}\gamma_{,1}(\frac{\hat{\mu}_m}{\ln 2}, \frac{\hat{\mu}_m}{\hat{o}\ln 2}). \tag{3.64}$$

The two terms on the RHS can now be collected as the single partial derivative of message number $\mu_m$:

$$\Lambda \ln 2 = -\partial_{\hat{\mu}_m}\gamma(\frac{\hat{\mu}_m}{\ln 2}, \frac{\hat{\mu}_m}{\hat{o}\ln 2}), \tag{3.65}$$

where the relative load is $\Lambda \equiv \lambda/N_0$.

The two optimization equations are summarized below:

$$\Lambda \ln 2 = -\partial_{\hat{\mu}_m}\gamma(\frac{\hat{\mu}_m}{\ln 2}, \frac{\hat{\mu}_m}{\hat{o}\ln 2}), \tag{3.66}$$

$$\frac{\hat{E}}{\Lambda} = \frac{\hat{\varepsilon}}{\lambda} = \frac{1}{\Lambda \ln 2}p_\Gamma(\hat{\mu}_m|\frac{\hat{\mu}_m}{\ln 2}, \hat{o}\ln 2). \tag{3.67}$$

The optimal overabundance is shown for a range of relative loads in Fig. 3.9. The optimal translation efficiency and scaled translation efficiency are shown for a range of relative loads in Fig. 3.10.

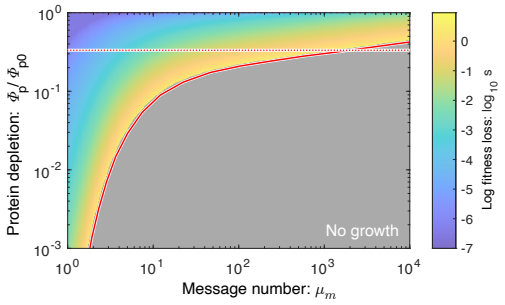



Figure 3.11: **Optimal expression levels are buffered.** The predicted fitness loss as a function of protein depletion level and message number for bacterial cells (including the noise floor). Due to the overabundance phenomenon, all proteins are buffered against depletion, but low-expression genes are particularly robust due to higher overabundance. The solid red line represents $1/o$, and predicts the range of depletion values for which cell growth is predicted. The dotted red line represents a three-fold depletion.

*Optimization of message number only*

Consider the special case of optimizing the message number only at fixed translation efficiency. Eq. 3.59 is the condition; however, in this case it makes sense to adsorb both the message and protein metabolic load into a single metabolic load. The optimum message number satisfies the equation:

$$\frac{(\lambda+\varepsilon)\ln 2}{N_0} = -[\partial_{\hat{\mu}_m}\gamma(\hat{\mu}_m, \hat{n}_m)]_{\hat{n}_m=\frac{\hat{\mu}_m}{\hat{o}}}. \tag{3.68}$$

We define a modified relative load:

$$\Lambda' \equiv \frac{(\lambda+\varepsilon)}{N_0}, \tag{3.69}$$

and substitute this into the optimum message number equation:

$$\Lambda' \ln 2 = -[\partial_{\hat{\mu}_m}\gamma(\hat{\mu}_m, \hat{n}_m)]_{\hat{n}_m=\frac{\hat{\mu}_m}{\hat{o}}}. \tag{3.70}$$

which is clearly closely related to Eq. 3.65.

We compare this modified expression to the original for optimum overabundance as a function of message number in Fig. 3.2C and demonstrate that the two make nearly identical predictions.

*Inclusion of the noise floor*

In bacterial cells, the noise is dominated by the noise floor for high expression genes. Including the noise floor, the coefficient of variation squared is ([2] and Sec. 3.10):

$$\mathrm{CV}_p^2 = \tilde{a}(\mu_m)^{-1} = \tfrac{\ln 2}{\mu_m} + C_0, \tag{3.71}$$

where $C_0 = 0.1$ for bacterial cells [2]. In spite of the addition of noise from the noise floor, the observed distribution of protein number is still well described by the gamma distribution [2]; however, we need to modify the statistical parameters to account for the noise floor. (See the definition of the *statistical noise model* in Sec. 3.7.1.) The modified gamma parameters are:

$$a = \tilde{a}, \tag{3.72}$$

$$\theta = \varepsilon\frac{\mu_m}{\tilde{a}}, \tag{3.73}$$

chosen such that the noise is determined by Eq. 3.71 but the protein number remains:

$$\mu_p = \varepsilon\mu_m, \tag{3.74}$$

the product of the message number and translation efficiency.

The qualitative effect of the noise floor is to increase the noise, especially for low-copy messages. Above $\mu_m = 7$ messages, the noise is dominated by the noise floor. Increases in transcription above this point have little effect on reducing the noise. As a consequence, the overabundance stays high, even for high copy messages. We compare this modified expression

to the original for optimum overabundance as a function of message number in Fig. 3.2C and demonstrate that bacterial cells are predicted to have much higher overabundance at high expression levels.

### *3.7.5 Discussion: Understanding the rationale for overabundance*

Essential protein overabundance is the signature prediction of the RLTO model. Its mathematical rationale is the highly-asymmetric fitness landscape. To understand why we expect this rationale to be generic, consider the form of the optimization condition for message number:

$$\Lambda \ln 2 = -\frac{\partial}{\partial \hat{\mu}_m}\gamma(\tfrac{\hat{\mu}_m}{\ln 2}, \tfrac{\hat{\mu}_m}{\delta \ln 2}). \tag{3.75}$$

The growth rate is maximized when the probability of slow-growth (*e.g.* arrest) is roughly equal to the relative load of adding one more message. Since the cell makes roughly $10^5$ messages per cell cycle, the relative load is extremely small and therefore the probability of slow growth must be as well. Making this probability very small requires vast overabundance for the inherently-noisy, low-expression proteins.

The reason we expect the RLTO model protein abundance predictions to be robust is that we generically expect the fitness cost of overabundance to be small due to the high multiplicity (*i.e.* total number of genes); whereas, the fitness cost of arrest of essential processes is very high.

### *3.7.6 Discussion: RLTO predicts larger overabundance in bacteria.*

There are two distinctive features of bacterial cells that could affect the model predictions: (i) the translation efficiency is constant [33] and bacterial gene expression is subject to a large-magnitude noise floor that increases the noise for high-expression genes [2]. The optimization of message number at fixed translation efficiency does result in a slightly modified optimization condition for the message number (Sec. 3.7.4); however, the predicted overabundance is only subtly perturbed (Fig. 3.2C). In contrast, the noise floor increases the predicted

overabundance, especially for high-expression proteins. As a result, the RLTO model predicts that the vast majority of bacterial proteins are expressed in significant overabundance. (See Fig. 3.2C.)

### *3.7.7 Discussion: RLTO predicts proteins are buffered to depletion.*

A principle motivation for our analysis is the observation that many protein levels appear to be buffered. To explore the prediction of the RLTO model for protein depletion, we first computed the optimal message numbers and translation efficiencies for a range of protein thresholds. To model the effect of protein depletion, we computed the change in growth rate as function of protein depletion (equivalent to a reduction of the translation efficiency relative to the optimum.) The growth rate is shown in Fig. 3.11 for the RLTO model with parameters representative of a bacterial cell. (See Sec. 3.7.4.)

In general, the RLTO model predicts that protein numbers have very significant robustness (*i.e.* buffering) to protein depletion. This is especially true for low expression proteins that are predicted to have the largest overabundance. For these genes, even a ten-fold depletion leads to very subtle reductions in the growth rate. For a three-fold reduction in the growth rate, only the very highest-expression genes (*e.g.* ribosomal genes) are expected to lead to qualitative phenotypes.

### *3.7.8 Results: Detailed development of load balancing*

*Prediction of the optimal load ratio.*

The two-stage amplification of the central dogma implies that the expression and noise levels can be controlled independently by the balance of transcription to translation. How does the cell achieve high and low gene expression optimally, and how does this strategy depend on the message cost?

To understand the optimization, we first define the load ratio $R$ for a gene as the metabolic cost of translation relative to transcription:

$$R \equiv \frac{\mu_p}{\lambda \mu_m} = \frac{\varepsilon}{\lambda}. \tag{3.76}$$

In Sec. 3.7.4, we show that the optimal load ratio is:

$$\hat{R} \equiv \frac{1}{\Lambda \ln 2}\ p_\Gamma(\hat{\mu}_m | \tfrac{\hat{\mu}_m}{\ln 2}, \tfrac{1}{\delta \ln 2}), \tag{3.77}$$

where $p_\Gamma$ is the PDF of the gamma distribution. The optimal load ratio is shown in Fig. 3.10C.

The dependence of the optimal load ratio $\hat{R}$ on $\Lambda$ is extremely weak, but it is strongly dependent on message number. As a result, for low transcription genes ($\mu_m \ll 10$), the metabolic load is predicted to be dominated by transcription; whereas, for highly transcribed genes ($\mu_m \gg 10$), the metabolic load is dominated by translation. These predictions are robust since they are independent of the relative load $\Lambda$.

*Measurements of the load ratio*

Unfortunately, there is somewhat limited data to which to compare the model. The best source we found was Kafri *et al.* [5] who analyzed the differences in fitness between transcription and transcriptional-and-translation of a fluorescent protein driven by the pTDH3 promoter in yeast. This promoter is one of the strongest in yeast. Based on the RLTO model, we would predict this promoter to have a very high translation efficiency and therefore a large load ratio; however, the translation efficiency is much lower than one would predict based on a global analysis and likewise its load ratio is roughly unity, which based on the smaller than expected translation efficiency is broadly consistent with our expectations. A satisfactory test of this prediction will require larger-scale measurements that probe more representative genes.

### *3.7.9 Results: Translation efficiency is predicted to increase with transcription.*

Now that we have defined the optimal load ratio (Eq. 3.77), the equation for optimal translation efficiency can be written concisely:

$$\hat{\varepsilon} = \lambda \hat{R} \qquad \text{or} \qquad \hat{E} = \Lambda \hat{R}, \tag{3.78}$$

where $\hat{R}$ depends weakly on the relative load $\Lambda$. The RLTO model predicts that optimal partitioning of amplification between transcription (gain $\mu_m$) and translation (gain $\varepsilon$) has two important qualitative features: (i) As the message cost ($\lambda$) rises, the optimal translation efficiency increases in proportion. (ii) The optimal translation efficiency is also approximately proportional to message number ($\hat{\varepsilon} \propto \mu_m$). (See Fig. 3.4A.) Therefore, the RLTO model predicts that low expression levels should be achieved with low levels of transcription and translation, whereas high expression genes are achieved with high levels of both. We call this relation between optimal transcription and translation *load balancing*.

### *3.7.10 Results: RLTO predicts that message number responds to message cost*

We will first focus on analyzing the implications of the message cost dependence in Eq. 3.78. At a fixed load ratio, Eq. 3.78 clearly implies that the translation efficiency increases as the message cost $\lambda$ increases; however, the message number (and load ratio) also respond to compensate to changes in $\lambda$. To probe the dependence on message cost in an experimentally relevant context, consider optimal message numbers in a reference condition (relative load $\Lambda_0$) relative to a second perturbed condition (relative load $\Lambda$). The predicted relation between the optimal messages numbers is shown in Fig. 3.12A. The resulting relation between the optimal message numbers is roughly linear on a log-log plot, predicting the approximate power-law relation:

$$\hat{\mu}_m(\Lambda) \propto \hat{\mu}_m(\Lambda_0)^{\alpha}, \tag{3.79}$$

describing a non-trivial global change in the regulatory program.

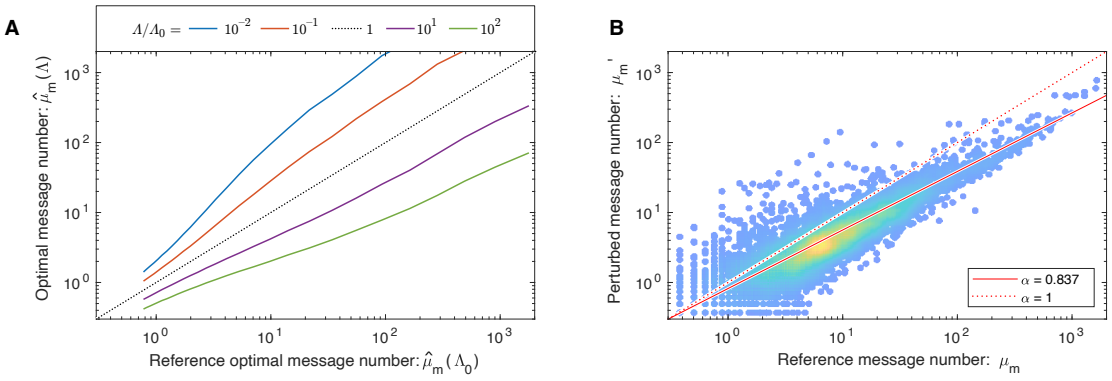



Figure 3.12: **The message cost affects transciption genome wide. Panel A: Message number decreases with increased relative load $\Lambda$.** The optimal message number responds to changes in the message cost. The RLTO model predicts an approximate power-law relation (linear on a log-log plot) between message numbers. **Panel B: A power-law relation is observed.** To test whether central dogma regulation would adapt dynamically as predicted, we analyzed the relation between the yeast transcriptome under reference conditions and phosphate depletion (perturbed), which increases the message cost [5]. (Data from Ref. [6].) As predicted by the RLTO model, a global change in regulation is observed, which generates a power-law relation with scaling exponent $\alpha = 0.837 \pm 0.01$. The observed exponent is smaller than one, as predicted by an increased relative load $\Lambda$.

### *3.7.11 Results: RLTO predicts the yeast global regulatory response*

To test the RLTO predictions, we compared the relative message numbers for yeast growing under phosphate depletion, which increases the message cost [5], to a reference condition [6]. As predicted, the relative transcriptome data was well described by a power law (Eq. 3.79) and the observed slope was smaller than one: $\hat{\alpha} = 0.837 \pm 0.001$, as predicted by the increased message cost. See Fig. 3.12B.

The observation of this large-scale regulatory change has an important implication: This response supports a nontrivial hypothesis that the RLTO model not only can predict how the cell is optimized in an evolutionary sense, but can predict global regulatory responses as well.

Note that Metzl-Raz *et al.* [6] also explored conditions that increased the cost of protein; however, here the predictions of the model are ambiguous. The complication arises due to the observed decrease in size of the cells in the experimental condition, which decreases the total metabolic load $N_0$. As discussed above, the relative load $\Lambda = \lambda/N_0$ is the key determinant in Eq. 3.79; however, even as the relative cost of transcription $\lambda$ decreases in the experimental condition, the total metabolic load $N_0$ also decreases, making no clear prediction about how the relative load $\Lambda$ changes.

### *3.7.12 Methods: Prediction of the proteome fraction*

*Parameter-free prediction of proteome fraction.*

We now turn our focus to an analysis of the implications of *load balancing*: the message number dependence of the optimal translation efficiency (Eq. 3.78). The most direct test of this prediction is measuring the relation between proteome fraction and message number. The RLTO model predicts proteome fraction (Eq. 3.46):

$$\hat{\Phi}_p = \hat{E}\hat{\mu}_m \propto \hat{\mu}_m^2, \tag{3.80}$$

where $\mu_m$ is the observed message number and the optimal relative translation efficiency is predicted by Eq. 3.78. The proportionality is only approximate but gives important intuition for how protein number depends on message number in the RLTO model, in contrast to a constant-translation-efficiency model: $\Phi_p \propto \mu_m$. To compare these predictions to protein abundance measurements, we will renormalize the protein fraction to be defined relative to total protein number rather than $N_0$. This renormalization eliminates the $\Lambda$ dependence to result in a parameter-free prediction of the proteome fraction.

*RLTO: proteome fraction*

Starting from Eq. 3.108, clearly:

$$\hat{\mu}_p \propto \hat{\mu}_m \; p_\Gamma(\hat{\mu}_m | \tfrac{\hat{\mu}_m}{\ln 2}, \hat{o} \ln 2), \tag{3.81}$$

which can be used to predict the proteome fraction (where we have restored the explicit gene $i$ subscript):

$$\hat{\Phi}_{p,i} \equiv \tfrac{\hat{\mu}_{p,i}}{\sum_j \hat{\mu}_{p,j}}, \tag{3.82}$$

where the second subscript is the gene index. To predict the proteome fraction, we computed the proportionality constant $C$:

$$C \equiv \textstyle\sum_i \left[ \mu_{m,i} p_\Gamma(\mu_m | \tfrac{\mu_{m,i}}{\ln 2}, o \ln 2)|_{o=\hat{o}(\mu_{m,i})} \right], \tag{3.83}$$

where the message numbers $\mu_{mi}$ for gene $i$ are the experimentally observed message numbers, the implicit $o_i$ values are predicted by the RLTO model (Eq. 3.65) for message number $\mu_{mi}$ and the sum index $i$ runs over all genes. The predicted optimal proteome fraction is:

$$\hat{\Phi}_{p,i} = C^{-1} \left[ \mu_{m,i} p_\Gamma(\mu_{m,i} | \frac{\mu_{m,i}}{\ln 2}, o \ln 2)|_{o=\hat{o}(\mu_{m,i})} \right], \tag{3.84}$$

which generates the predicted solid curves shown in Fig. 3.4BCD.

*Constant-translation-efficiency model: proteome fraction*

For the constant translation efficiency model, we define the normalization:

$$C' \equiv \sum_i \mu_{m,i}, \tag{3.85}$$

and the predicted proteome fraction is:

$$\Phi'_{p,i} = {C'}^{-1} \mu_{m,i}, \tag{3.86}$$

which generates the predicted dotted curves shown in Fig. 3.4BCD.

*Sources of experimental data for proteome fraction analysis*

For *E. coli* data, the protein abundance data was generated by mass spec measurements and the message abundance data was from [33]. For the yeast data, the protein abundance data is measured by mass spec and message abundances are determined by [4]. For the mammalian data, we used mouse data. The protein abundance data is measured by mass-spec and message abundances are determined by [59].

We estimated the message number $\mu_m$ as described in Sec. 3.9.2. For the mouse data, the study provided message lifetimes, the cell cycle duration and abundances in molecules per cell [59]. For the *E. coli* and yeast, the total number of proteins, messages *etc*, cell cycle duration and message lifetimes for each organism and their sources are described in Tab. 3.2.

### *3.7.13 Results: Load balancing is observed in eukaryotic cells*

A non-trivial prediction of the RLTO Model is that translation efficiency and message number should be roughly proportional. Qualitatively, this strategy allows expression levels to be increased while distributing the added metabolic load between transcription, which reduces noise, and translation, which does not affect the noise. We predict the optimal translation efficiency versus message number which matches the observations in eukaryotic

cells (Fig. 3.4BC). However, in *E. coli*, the translation efficiency and message number are *not* strongly correlated (Fig. 3.4D). Why does this organism appear not to load balance? We demonstrate that the observed translation efficiency is consistent with the RLTO model, augmented by a ribosome-per-message limit. (See Sec. 3.7.15.) Hausser *et al.* have proposed just such a limit, based on the ribosome footprint of mRNA molecules [15]. (See Sec. 3.7.17.) Although this augmented model is consistent with central dogma regulation in *E. coli*, it is not a complete rationale. This proposed translation-rate limit could be circumvented by increasing the lifetime of *E. coli* messages which would increase the translation efficiency. Why the message lifetime is as short as observed will require a more detailed *E. coli*-specific analysis.

### *3.7.14 Discussion: Relation between load balancing and previous results*

Hausser *et al.* have previously performed a more limited analysis of the trade-off between metabolic load and gene-expression noise [15]. In this section, we will provide some more context into the differences between the two approaches.

Hausser *et al.* assume a symmetric (not an asymmetric) fitness landscape and consider only the metabolic cost of transcription (but not translation). Their model depends on two (not one) gene-specific parameters: an optimal protein number and a sensitivity, which defines the curvature of the fitness [15]. The authors maximize fitness with respect to the transcription rate (but not the translation rate) and the condition they derive depends on the two (not one) unknown, gene-specific parameters. As a result, this condition is not predictive of global regulatory trends without non-trivial, gene-specific measurements or assumptions about the unknown sensitivity.

The Hausser model assumes that the growth rate has the form:

$$k = k_0 - \tfrac{1}{2}|k''|(N_p - \mu_p)^2 - k_0 \Lambda \mu_m, \tag{3.87}$$

where $k$ is the growth rate, and we have rewritten the form of the fitness to better match our own definitions. Here $k''$ is the second derivative of the growth rate at the optimal protein number $\mu_p$ and $N_p$ is the stochastic protein number. If we take the expectation with respect to the protein number, we get:

$$k = k_0 - \tfrac{1}{2}|k''|\sigma_p^2 - k_0\Lambda\mu_m, \tag{3.88}$$

and substituting the noise model for the variance of the protein number gives:

$$k = k_0 - \tfrac{1}{2}|k''|\mu_p^2(\tfrac{\ln 2}{\mu_m} + C_0) - k_0\Lambda\mu_m, \tag{3.89}$$

where $C_0$ is the noise floor, and we have assumed the mean protein number is optimal ($\mu_p$). If we maximize the growth rate with respect to $\mu_m$, we get the following condition on the optimal message number:

$$\hat{\mu}_m^2 = \tfrac{1}{2}\tfrac{|k''|}{k_0}\mu_p^2\tfrac{\ln 2}{\Lambda}, \tag{3.90}$$

which depends on the unknown curvature $k''$. To make global predictions about how transcription and translation are related, some added assumptions are necessary to describe how $k''$ scales with protein abundance.

To illustrate how this expression does not make explicit global predictions, let's consider a number of plausible possibilities. First, we will assume that $k''$ is independent of $\mu_p$ and on average all proteins are equally sensitive to changes in protein number. In this case, we find:

$$\mu_p \propto \hat{\mu}_m\sqrt{\Lambda}, \tag{3.91}$$

$$\hat{\varepsilon} \propto \sqrt{\Lambda}, \tag{3.92}$$

implying a constant translation efficiency which is inversely proportional to the square root of the relative load.

Alternatively, we can assume that $k'' \propto \mu_p^{-2}$ and, on average, the cell is equally sensitive to changes in the relative number of proteins (*i.e.* $\Delta p/\mu_p$), regardless of expression level. In this case,

$$\hat{\mu}_m \propto 1/\sqrt{\Lambda}, \tag{3.93}$$

$$\hat{\varepsilon} \propto \mu_p\sqrt{\Lambda}, \tag{3.94}$$

implying a constant message number, irrespective of expression level, and a translation efficiency that is proportional to expression level.

Finally, we will assume that $k'' \propto \mu_p^{-1}$, which is the intermediate case. Here:

$$\mu_p \propto \hat{\mu}_m^2\Lambda, \tag{3.95}$$

$$\hat{\varepsilon} \propto \hat{\mu}_m\Lambda, \tag{3.96}$$

implying that translation efficiency should increase with message number, analogous to our prediction. It appears that Hausser *et al.* implicitly also favor this model, since they define their sensitivity parameter to include a power of protein number $\mu_p$. They justify this assumption by arguing that since $\sigma_p^2 \propto \mu_p$, it makes sense to define the *sensitivity to noise* to include a factor of $\mu_p$ [15]. At best, this is somewhat fuzzy logic since, as we demonstrate in the paper, Eq. 3.96 implies that the protein variance does not scale $\sigma_p^2 \propto \mu_p$!

The authors also propose a lower limit on the translation-transcription ratio; however, their limit is dependent on the noise floor, which only affects genes with the highest transcription rates in eukaryotic cells. The implementation of a more appropriate estimate of the noise, relevant for the vast majority of genes, does not lead to the same limit.

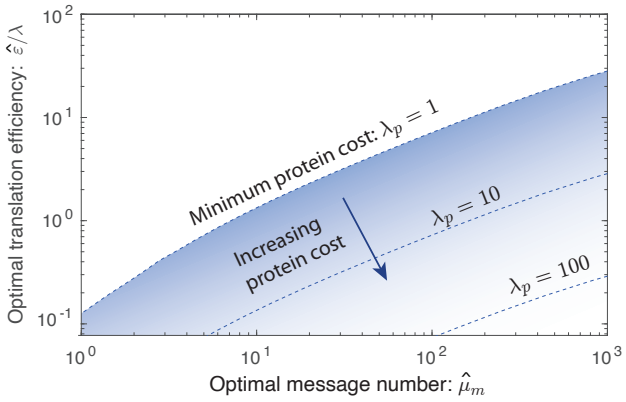



Figure 3.13: **Increased protein cost decreases optimal translation efficiency.** A protein cost of $\lambda_p = 1$ corresponds to the metabolic cost of protein synthesis only, and is the minimum protein cost. For larger protein costs, the optimal translation efficiency is lower. As a result, the $\lambda_p = 1$ curve represents an upper bound of the optimal translation efficiency.

### *3.7.15 Methods: Analysis of translational limits & gene-specific load analysis*

To explore the consequences of a protein-specific load, we can modify the metabolic load term in the growth rate equation (Eq. 3.58):

$$E\mu_m \to \lambda_p E\mu_m, \tag{3.97}$$

which includes an additional parameter: the protein cost $\lambda_p$, which is 1 if the fitness cost is equal to the metabolic load and greater than one if the cost is higher. We will also treat the metabolic load per message $\lambda$ as a gene-specific parameter in this section only. The optimization can be repeated for this augmented model.

To analyze the effect of increased protein load, we modify Eq. 3.58:

$$\ln \tfrac{k}{k_0} = -(\Lambda + \lambda_p E)\mu_m - \tfrac{1}{\ln 2}\gamma(\tfrac{\mu_m}{\ln 2}, \tfrac{n_p}{\varepsilon \ln 2}), \tag{3.98}$$

to contain the supplemental load factor $\lambda_p$ which is unity if the only protein load is metabolic and $\lambda_p > 1$ if there is additional load (*e.g.* toxicity). The optimization conditions (Eqs. 3.59 and 3.60) become:

$$0 = -\frac{\lambda+\lambda_p\hat{\varepsilon}}{N_0} - \frac{1}{(\ln 2)^2}\gamma_{,1}\left(\frac{\hat{\mu}_m}{\ln 2}, \frac{n_p}{\hat{\varepsilon}\ln 2}\right), \tag{3.99}$$

$$0 = -\frac{\lambda_p\hat{\mu}_m}{N_0} + \frac{n_p}{\hat{\varepsilon}^2(\ln 2)^2}\gamma_{,2}\left(\frac{\hat{\mu}_m}{\ln 2}, \frac{n_p}{\hat{\varepsilon}\ln 2}\right). \tag{3.100}$$

Using the same algebraic approach as before, we can derive the same optimal overabundance and load equations:

$$\Lambda \ln 2 = -\partial_{\hat{\mu}_m}\gamma\left(\frac{\hat{\mu}_m}{\ln 2}, \frac{\hat{\mu}_m}{\hat{o}\ln 2}\right), \tag{3.101}$$

$$\hat{R} = \frac{1}{\Lambda \ln 2} p_\Gamma\left(\hat{\mu}_m \middle| \frac{\hat{\mu}_m}{\ln 2}, \hat{o}\ln 2\right); \tag{3.102}$$

however, the relation between the load and the translation efficiency now has an extra factor: $\lambda_p$:

$$R = \frac{\lambda_p\varepsilon\mu_m}{\lambda\mu_m} = \frac{\lambda_p\varepsilon}{\lambda}, \tag{3.103}$$

representing the modified total load ratio.

### *3.7.16 Results: Increased protein-specific cost reduces the optimal translation efficiency.*

The relation between the overabundance and message number is unchanged (Eq. 3.75). This result can be rationalized in the following way: The optimal overabundance is determined by the noise which is determined by message number only. This relation is unaffected by the added parameter $\lambda_p$. However, the optimal translation efficiency is affected:

$$\hat{\varepsilon} = \frac{\lambda}{\lambda_p}\hat{R}, \tag{3.104}$$

where $\hat{R}$ is the optimal load ratio, defined by Eq. 3.78. The optimal curves are shown in Fig. 3.13.

How do these added considerations affect the RLTO predictions? First, we consider message and protein length. What are the optimal translation efficiencies for two proteins, one ten times the length of the other, at fixed protein number? In this case, we will assume that both the transcriptional cost ($\lambda$) as well as the translational cost ($\lambda_p$) increase tenfold. These increases cancel, resulting in the same optimal translation efficiency since it is only the relative cost of transcription to translation that is determinative of the translation efficiency.

Now consider a tenfold protein-specific increase in protein cost at fixed message cost and fixed protein number. The message number and translation efficiency would change by compensatory factors of 10:

$$\hat{\mu}_m \rightarrow 10 \cdot \hat{\mu}_m, \tag{3.105}$$

$$\hat{\varepsilon}_m \rightarrow \tfrac{1}{10} \cdot \hat{\varepsilon}_m, \tag{3.106}$$

to maintain the protein number.

Returning to our original motivation, we can understand how genes with a higher protein-to-message cost migrate downwards and rightwards off the optimal $\lambda_p = 1$ curve, predicting a cloud versus a narrow strip in proteome fraction measurements shown in Fig. 3.4. If the relative load $\Lambda$ were directly measured, we would expect the predicted optimal translation efficiency curve for $\lambda_p = 1$ to lie at the top edge of the observed data cloud rather than the bisecting it. This bisection is the consequence of fitting an effective relative load parameter to the abundance data in the unaugmented RLTO model.

### *3.7.17 Discussion: Translation limits in E. coli*

A critical assumption in the RLTO model to this point has been that the optimal central dogma parameters are realizable in the cell; however, translation can be limited by a number of different mechanisms. The superior performance of the constant- over the optimal-translation-efficiency model in *E. coli* (Fig. 3.4D) suggests that this assumption may not be satisfied for bacteria. How do translation limits affect the model phenomenology?

When considering possible limits on translation, there are two natural mechanisms: (i) ribosome-number limit, where the number of ribosomes in the cell limits translation and (ii) a ribosome-per-message limit, where the number of ribosomes per message is limiting. Assuming the ribosome-number-limit mechanism, the original unconstrained optimization problem can be recast as a constrained optimization problem where the protein cost $\lambda_p$ is reinterpreted as a Lagrange multiplier to constrain the number of proteins translated (*e.g.* [69]). In spite of this reformulation, we would still predict the same functional form for the coupling between the optimal translation efficiency and message number. (*I.e.* it is still optimal to have a higher translation efficiency for highly-expressed genes even if the total number of proteins is fixed.) Therefore, the ribosome-number-limit mechanism cannot be the rationale for the constant translation efficiency observed in *E. coli*.

Assuming the ribosome-per-message-limit mechanism, we limit the translation efficiency to a restricted range of values. If the unconstrained optimum lies above this range, the optimum is at the maximum limiting value. If the unconstrained optima for all genes lie above the realizable range, the model predicts a translation efficiency uncoupled from the message number, as observed. These predictions are consistent with the observed central dogma regulatory program in *E. coli*. In added support of this hypothesis, Hausser *et al.* have argued that *E. coli* translates close to just such a ribosome-per-message limit as a consequence of the finite ribosome complex footprint on a message [15].

### *3.7.18 Methods: Estimate of the message cost and metabolic load*

We can estimate the message cost $\lambda$ from the known total protein number for yeast and mammalian cells. (For *E. coli* this estimate is not possible since the protein cost in not determinative of the translation efficiency.)

The optimal translation efficiency for gene $i$ is (Eq. 3.63):

$$\hat{\varepsilon} = \frac{\lambda}{\Lambda \ln 2} p_\Gamma(\hat{\mu}_m | \frac{\hat{\mu}_m}{\ln 2}, \hat{o} \ln 2), \tag{3.107}$$

and therefore the optimal protein number for gene $i$ is:

$$\hat{\mu}_p = \hat{\mu}_m \hat{\varepsilon} = \frac{\lambda}{\Lambda \ln 2} \hat{\mu}_m p_\Gamma(\hat{\mu}_m | \frac{\hat{\mu}_m}{\ln 2}, \hat{o} \ln 2). \tag{3.108}$$

We define the normalization constant $A$ (where we restore the explicit gene $i$ subscript):

$$A = \sum_i \hat{\mu}_{m,i} \cdot \frac{1}{\Lambda \ln 2} p_\Gamma(\hat{\mu}_{m,i} | \frac{\hat{\mu}_{m,i}}{\ln 2}, \hat{o}_i \ln 2), \tag{3.109}$$

where we have restored the explicit gene $i$ index running over all genes. Now, by summing Eq. 3.108, over all genes, we derive an expression for the total protein number $N_p^{\text{tot}}$ in terms of the message cost $\lambda$ and the normalization constant $A$:

$$N_p^{\text{tot}} = \lambda A. \tag{3.110}$$

Solving for the protein cost results in the estimate:

$$\hat{\lambda} = \frac{N_p^{\text{tot}}}{A}. \tag{3.111}$$

This message cost estimate $\hat{\lambda}$ can then be plugged into the metabolic load definition:

$$N_0 \equiv L_0 + \sum_i (\lambda + \varepsilon_i) \mu_{m,i}, \tag{3.112}$$

to estimate its size:

$$\hat{N}_0 \equiv L_0 + \hat{\lambda} N_m^{\text{tot}} + N_p^{\text{tot}}, \tag{3.113}$$

where we have ignored the non-protein and non-message contributions to the load ($L_0 = 0$).

*Detailed protocol*

We first estimate the message numbers, as described in Sec. 3.9.2, from data. For each gene $i$, we set the optimal message number equal to the observed message number and then

compute the optimal overabundance from the message number using Eq. 3.66. (Since the result is independent of the assumed $\Lambda$ value, we set an arbitrary initial value of $\Lambda = 10^{-5}$.) We then use these single gene optimal message number and overabundances to compute $A$ using Eq. 3.109. In Eqs. 3.111 and 3.113, we use the $N_p^{\text{tot}}$ from Tab. 3.2. $N_m^{\text{tot}}$ is computed by summing the estimated message numbers.

*Estimate the message cost and metabolic load in yeast*

In yeast, the estimates are:

$$A = 4.8 \times 10^5, \tag{3.114}$$
$$\hat{\lambda} = 1.0 \times 10^2, \tag{3.115}$$
$$\hat{N}_0 = 6.2 \times 10^7, \tag{3.116}$$
$$\hat{\Lambda} = 1.6 \times 10^{-6}, \tag{3.117}$$

where the data sources are described in detail in Sec. 3.9.1.

*Estimate the message cost and metabolic load in human cells*

In human cells, the estimates are:

$$A = 4.3 \times 10^6, \tag{3.118}$$
$$\hat{\lambda} = 7.1 \times 10^2, \tag{3.119}$$
$$\hat{N}_0 = 2.4 \times 10^9, \tag{3.120}$$
$$\hat{\Lambda} = 2.9 \times 10^{-7}, \tag{3.121}$$

where the data sources are described in detail in Sec. 3.9.1.

*Estimate of the modified relative load in bacterial cells*

In bacterial cells, we will assume a constant translation efficiency model. We therefore use the modified relative load formula (Eq. 3.69) to estimate $\Lambda'$. We will assume that the load is dominated by proteins and messages:

$$N_0 = \sum_i (\lambda + \varepsilon)\mu_{m,i} = (\lambda + \varepsilon)N_m, \tag{3.122}$$

where $N_m$ is the total number of messages. We can then solve this equation for $\Lambda'$:

$$\hat{\Lambda}' = \tfrac{\lambda+\varepsilon}{N_0} = \tfrac{1}{N_m} \approx 10^{-5}, \tag{3.123}$$

based on the total message number estimate for *E. coli*. (See Tab. 3.2.)

## 3.8 Supplemental Material: Model robustness & exploring alternatives to RLTO

In this section, we investigate the phenomenology of three different single-cell growth rate functions to determine what model features result in overabundance. We consider an *arrest model* (the RLTO model), a *slow-growth model*, and a *symmetric model*.

### *3.8.1 Methods: Defining alternative models*

In each case, we will assume that the protein number is described by a gamma distribution:

$$N_p \sim \Gamma(\tfrac{\mu_m}{\ln 2}, \varepsilon \ln 2). \tag{3.124}$$

We will assume the cell-cycle duration $T$ is determined by this stochastic protein number $N_p$ and then compute the population growth rate using Eq. 3.48 for a range of different message numbers $\mu_m$. In each case, $\tau_0 = 1/N_0$, $N_0 = 10^5$, $\varepsilon = 30$, $n_p = \varepsilon \ln 2$. The mean expression level is $\mu_p = \mu_m \varepsilon$.

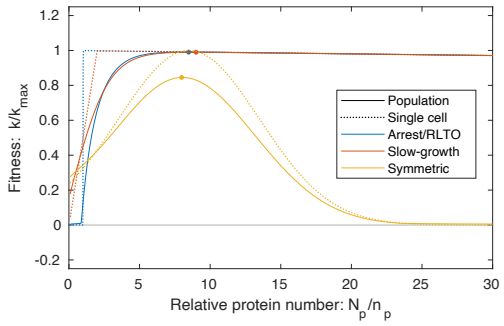



Figure 3.14: **Exploring the mathematical mechanism of overabundance.** Single-cell and population growth rate are compared for three different models: arrest (RLTO), slow-growth, and symmetric models. In the arrest model (RLTO), the growth rate goes to zero below threshold protein level $n_p$. In the slow-growth model, the growth rate transitions continuously to zero as the $N_p$ is depleted below $n_p$. In both the arrest and slow-growth models, there is a small negative slope above the threshold corresponding to the metabolic load. In the symmetric model, the fitness cost is symmetric about the optimum. Both the threshold-like and slow-growth models are optimized at mean expression levels $\mu_p$ far exceeding the threshold level $n_p$. This is a consequence of the highly-asymmetric dependence of the fitness on protein number $N_p$. This leads to the phenomenon of protein overabundance. In contrast, the symmetric model is optimized in close proximity to its single-cell optimum.

*Model 1: Arrest (RLTO) model*

The arrest (RLTO) model has cell cycle duration:

$$T = \tau_0 \begin{cases} \infty, & N_p < n_p \\ N_0 + N_p, & N_p > n_p \end{cases}, \tag{3.125}$$

where protein expression below threshold $n_p$ results in growth arrest.

*Model 2: Slow-Growth model*

In the slow-growth model, we imagine two processes: (i) checkpoint process X and (ii) other processes. The cell will divide after whichever process finishes last. Other processes will finish after time predicted by the metabolic load, identical to the threshold model defined above. However, we model checkpoint process X as the completion of a fixed amount of activity in an irreversible process. We will therefore assume it will take a time inversely proportional to the amount of enzyme X ($N_p$). The amount of activity is set by effective threshold $n_p$:

$$T = \tau_0 \max\{N_0 + N_p, \tfrac{2n_p N_0}{N_p}\} \tag{3.126}$$

such that $n_p$ defines the level of protein required to make the growth rate half the metabolic limit.

Unlike the arrest model, cell growth slows but does not stop for $N_p < n_p$. This model will test whether the results of the RLTO model are an artifact of the assumed arrest-based slow growth.

*Model 3: Symmetric model*

For the symmmetric model, we choose the model parameters such that the single-cell optimum was close to the other models: $n_0 = 8.5$, $\sigma_n = 5$. The cell-cycle duration is:

$$T = \tau_0 N_0 \exp(\tfrac{(N_p - n_0)^2}{2\sigma_n^2}), \tag{3.127}$$

such that the noise-free growth rate will be Gaussian is $N_p$.

### *3.8.2 Results: Overabundance is a robust prediction*

The growth rates as a function of the mean expression level $\mu_p$ are shown in Fig. 3.14. The symmetric model has a population optimum in close proximity to its single-cell optimum, as we intuitively expect. However, both the arrest (RLTO) model and the slow-growth model have optima far above the threshold number $n_p$. We therefore conclude that it is fitness asymmetry rather than growth arrest that is responsible for the overabundance phenomenon.

Why doesn't growth arrest of a sub-population lead to a stronger effect than the same sub-population growing slowly? In Ref. [65], we showed that the population doubling time $\overline{T}$ can be understood as the exponential mean of the stochastic cell-cycle duration:

$$\overline{T} \equiv f^{-1}[\mathbb{E}_T f(T)], \tag{3.128}$$

where $\mathbb{E}_T$ is the expectation over the stochastic duration $T$ and $f(t) \equiv \exp(-kt)$, where $k = \overline{T}^{-1} \ln 2$ is the population growth rate. Due to the functional form of $f(t)$, any long cell cycles are exponentially suppressed in their contribution to the exponential mean. Therefore, low-probability extremely-long-duration cell cycles only contribute to the growth rate by reducing the fraction of growing cells.

| Model organism | Growth condition | Doubling time: $T$ | Message lifetime: $\tau_m = \gamma_m^{-1}$ | Message recycling ratio: $m = T/\tau_m$ | Total number of messages /cell: $N_{m/c}^{\text{tot}}$ | Total number of messages /cell-cycle: $N_m^{\text{tot}}$ | proteins: $N_p^{\text{tot}}$ | Average translation efficiency: $\varepsilon$ | translation rate: $\beta_p$ ($\text{h}^{-1}$) |
|---|---|---|---|---|---|---|---|---|---|
| *Escherichia coli* (*E. coli*) | LB | 30 min [30] | 2.5 min [31] | 12 | $7.8 \times 10^3$ [70] | $9.4 \times 10^4$ | $3 \times 10^6$ [71] | 22 | 530 |
| | M9 | 90 min [30] | 2.5 min [31] | 36 | $2.4 \times 10^3$ [70] | $8.6 \times 10^4$ | $3 \times 10^6$ [71] | 24 | 580 |
| *Sacchromyces cerevisiae (Yeast–haploid)* | YEPD | 90 min [72] | 22 min [73] | 4 | $2.9 \times 10^4$ [74] | $1 \times 10^5$ | $5 \times 10^7$ [75] | $4 \times 10^2$ | 410 |
| *Mus musculus (Mammalian-mouse)* | Tissue | 27.5 h [59] | 15 h [59] | 1.8 | $1.7 \times 10^5$ [59] | $3 \times 10^5$ [59] | $3 \times 10^9$ [59] | $1 \times 10^4$ | 660 |
| *Homo sapiens (Human)* | Tissue | 24 h [44] | 14 h [76] | 1.7 | $3.6 \times 10^5$ [72] | $5 \times 10^5$ | $2 \times 10^9$ [71] | $4 \times 10^3$ | 120 |

Table 3.2: **Central dogma parameters for three model organisms with detailed references.** Columns three through seven hold representative values for measured central-dogma parameters for the model organisms described in the paper. Each value is followed by a reference for its source.

## *3.9 Supplemental Material: Quantitation of central dogma parameters for one-message-rule*

The RLTO model predicts the *one-message-rule* for the lower threshold on transcription for essential genes. In this section, we use transcriptome data from the literature to test this prediction. We first describe the sources of the data (Sec. 3.9.1), how the estimates are computed (Sec. 3.9.2), the results (Sec. 3.9.3) and discussion (Sec. 3.9.4).

### *3.9.1 Methods: Selection of central dogma parameter estimates*

The estimates for central dogma model parameters come from two types of data: (i) quantitative measurement of cellular-scale parameters for each organism (total number of messages in the cell, cell cycle duration, *etc*) and (ii) genome-wide studies quantitative of mRNA and protein abundance.

For the cellular-scale central dogma parameters, we relied heavily on an online compilation of biological numbers: BioNumbers [77]. This resource provides a collection of curated quantitative estimates for biological numbers, as well as their original source. In the interest

of conciseness, we have cited only the original source in the Tab. 3.2, although we are extremely grateful and supportive of the creators of the BioNumbers website for helping us very efficiently identify consensus estimates for the parameters of the central dogma parameters.

For the selection of genome-wide studies on abundance, we used many of the same resources cited in BioNumbers as well as studies selected by a previous study of a quantitative analysis of the central dogma: Hausser *et al.* [15].

*E. coli data*

1. Message lifetimes: The message lifetimes (and median lifetime) were taken from a recent transcriptome-wide study by Chen *et al.* [31]. These investigators measured the lifetime in both rapid (LB) and slow growth (M9).

2. Noise: Taniguchi *et al.* have performed a beautiful simultaneous study of the proteome and transcriptome with single-molecule sensitivity [2]. Although we use the noise analysis data from this study for our supplemental analysis of *E. coli* noise, it is not the source for our *E. coli* transcriptome data due to the extremely slow growth of the cells in this study (150 minute doubling time), which is not consistent with the growth conditions for the other sources of data.

3. mRNA abundance: Instead, we used data from the more recent Bartholomaus *et al.* study [70], which characterizes the transcriptome in both rapid (LB) and slow growth (M9).

4. Total cellular message number. This study was chosen since it was the source of the BioNumbers estimates of cellular message number in *E. coli* (BNID 112795 [77]).

5. Doubling time: The source of the doubling times for rapid (LB) and slow (M9) growth of *E. coli* comes from Bernstein [30].

6. Essential gene classification. The classification of essential genes in *E. coli* comes from the construction of the Keio knockout collection from Baba *et al.* [35].

7. Protein number. The total protein number in *E. coli* came from Milo's recent review of this subject [71].

*Yeast data*

1. Message lifetimes: The message lifetimes (and median lifetime) were taken from Chia *et al.* [73].

2. Noise: The noise data was taken from the Newman *et al.* study, which used flow cytometry of a library of fluorescent fusions to characterize protein abundance with single-cell resolution [1].

3. mRNA abundance: The transcriptome data comes from the very recent Blevins *et al.* study [78].

4. Total cellular message number. There are a wide-range of estimates for the total cellular message number in yeast: $1.5 \times 10^4$ [79] (BNID 104312 [77]), $1.2 \times 10^4$ [80] (BNID 102988 [77]), $6.0 \times 10^4$ [81] (BNID 103023 [77]), $2.6 \times 10^4$ [74] (BNID 106763 [77]) and $3.0 \times 10^4$ [82]. We used the compromise value of $2.9 \times 10^4$.

5. Doubling time: The doubling time was taken from [72].

6. Protein number. The total protein number in yeast comes from Futcher *et al.* [75].

7. Essential gene classification. The classification of essential genes in yeast comes from van Leeuwen *et al.* [57].

8. Proteome abundance data: The proteome abundance data came from two sources: flow cytometry of fluorescent fusions from Newman *et al.* [1] as well as mass-spec data from de Godoy *et al.* [3].

*Human data*

1. Message lifetimes: The message lifetimes (and median lifetime) were taken from Yang *et al.* [76] who reported a median half life of 10 h which corresponds to a lifetime of 14 h.

2. mRNA abundance: The transcriptome data comes from the data compiled by the Human Protein Atlas [83], which we averaged over tissue types.

3. Total cellular message number. The total cellular message number in human comes from Velculescu *et al.* [84] (BNID 104330 [77]).

4. Doubling time: The doubling time was taken from [44].

5. Protein number. The total protein number in human came from Milo's recent review of this subject [71].

6. Essential gene classification. The classification of essential genes in human comes from Wang *et al.* [58].

### *3.9.2 Methods: Quantitative estimates of central dogma parameters*

*Estimating the cellular message number:* $\mu_{m/c}$

For each model organism (and condition), we found a consensus estimate from the literature for the total number of mRNA messages per cell $N_{m/c}^{\text{tot}}$. This number and its source

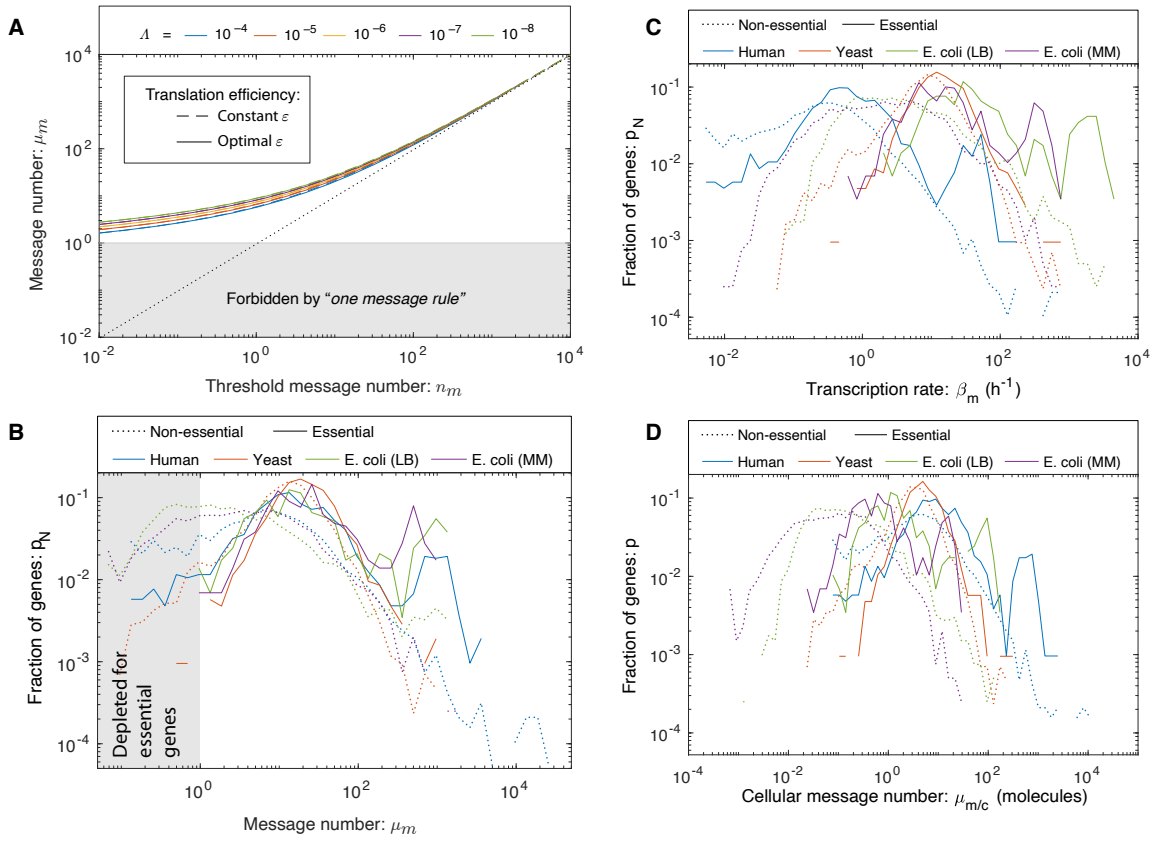



Figure 3.15: **The one message rule. Panel A: One-message-rule for essential genes.** For highly transcription genes (high $\mu_m$), little compensation for noise is required and the optimal message number tracks with the threshold message number $n_m$. However, as the threshold message number approaches one ($n_m \to 1$), the noise is comparable to the mean, and the optimal message number $\mu_m$ increases to compensate for the noise. As a result, a lower threshold of roughly one message per cell-cycle is required for essential genes. This threshold is predicted for both fixed (dashed) and optimized translation efficiency (solid). The threshold is weakly dependent on relative load $\Lambda$. **Panel B: A one message threshold is observed in three evolutionarily-divergent organisms.** As predicted by the RLTO model, essential, but not nonessential genes, are observed to be expressed above a one message per cell-cycle threshold. All organisms have roughly similar distributions of message number for essential genes, which are not observed for message numbers below a couple per cell cycle. **Panel C: The distribution of gene transcription rate.** The typical transcription rate varies by two orders-of-magnitude between organisms. **Panel D: The distribution of gene cellular message number.** There is also a two-order-of-magnitude variation between typical cellular message numbers. No consistent lower threshold is observed for either statistic.

are provided in Tab. 3.2. To estimate the number of messages corresponding to gene $i$, we re-scaled the un-normalized abundance level $r_i$:

$$N_{m/c,i} = N_{m/c}^{\text{tot}} \frac{r_i}{\sum_j r_j}, \tag{3.129}$$

where the sum over gene index $j$ runs over all genes.

*Estimating the transcription rate:* $\beta_m$

To estimate the transcription rate for gene $i$, we start from the estimated cellular message number $N_{m/c,i}$ and use the canonical steady-state noise model prediction for the cellular message number:

$$N_{m/c,i} = \beta_{m,i}/\gamma_{m,i}, \tag{3.130}$$

where $\gamma_{m,i}$ is the message decay rate. Since gene-to-gene variation in message number is dominated by the transcription rate (*e.g* [31]), we estimate the decay rate as the inverse gene-median message lifetime:

$$\gamma_{m,i} = \tau_m^{-1}, \tag{3.131}$$

for which a consensus value was found from the literature. This number and its source are provided in Tab. 3.2. We then estimate the gene-specific transcription rate:

$$\beta_{m,i} = N_{m/c,i}/\tau_m. \tag{3.132}$$

*Estimating the message number:* $\mu_m$

To estimate the message number of gene $i$, we use the predicted value from the canonical steady-state noise model:

$$N_{m,i} = T\beta_{m,i} = \frac{T}{\tau_m} N_{m/c,i}, \tag{3.133}$$

where $T$ is the doubling time and $N_{m/c,i}$ is the cellular message number (Eq. 3.129).

### *3.9.3 Results: Histograms of central dogma transcriptional statistics*

We generated histograms for each of the three transcriptional statistics: transcription rate $\beta_m$, cellular message number $\mu_{m/c}$, and message number $\mu_m$. The histograms for transcription rate and cellular message number do not show a consistent lower limit (as predicted) and are shown in Fig. 3.15; however, the histogram for message number does show a consistent lower bound for the three model organisms and is shown in Fig. 3.15B.

### *3.9.4 Discussion: E. coli essential genes below the one-message-rule threshold*

Since our own preferred model system is *E. coli*, we focus here. Our essential gene classification was based on the construction of the Keio knockout library [35]. By this classification, 10 essential genes were below threshold. (See Tab. 3.3.) Our first step was to determine what fraction of these genes were also classified as essential using transposon-based mutagenesis [85, 37]. Of the 10 initial candidates, only one gene, *ymfK*, was consistently classified as an essential gene in all three studies, and we estimate that its message number is just below the threshold ($\mu_m = 0.4$). *ymfK* is located in the lambdoid prophage element e14 and is annotated as a CI-like repressor which regulates lysis-lysogeny decision [38]. In $\lambda$ phase, the CI repressor represses lytic genes to maintain the lysogenic state. A conserved function for *ymfK* is consistent with it being classified as essential, since its regulation would prevent cell lysis. However, since *ymfK* is a prophage gene, not a host gene, it is not clear that its expression should optimize host fitness, potentially at the expense of phage fitness. In summary, closer inspection of below-threshold essential genes supports the threshold hypothesis.

## *3.10 Supplemental Material: Analysis of gene-expression noise*

This section provides a detailed development of gene expression noise. We continue the discussion of the model from Sec. 3.7.1 that provided a self-contained development of the noise models developed by others which are the input to the RLTO model. Secs. 3.10.1-3.10.6 describe the RLTO prediction of non-canonical noise scaling and the test of this model.

| Gene name | Message number: $\mu_m$ | Annotated function from Ecocyc | Essential (E)/ Nonessential (N) Ref. [35], [85], [37] |
|---|---|---|---|
| *alsK* | 0.3 | The alsK gene encodes a D-allose kinase. Its role in the degradation of D-allose is unclear; AlsK is not required for utilization of a D-allose carbon source; this effect may be due to the presence of other ambiguous sugar kinases within *E. coli* K-12. | E, N, N |
| *bcsB* | 0.4 | BcsB is encoded in a predicted operon together with *bcsA*, *bcsZ* and *bcsC*. In other organisms, these genes are involved in cellulose biosynthesis, a characteristic of the rdar (red, dry and rough) morphotype. However, the K-12 laboratory strain of *E. coli* does not show a rdar morphotype and does not produce cellulose. | E, N, N |
| *entD* | 0.4 | AcpS is the founding member of a 4'-phosphopantetheinyl (P-pant) transferase protein family that includes *E. coli* EntD, *E. coli* o195 protein, and *Bacillus subtilis* Sfp; family members share two conserved motifs but relatively low sequence identity overall. | E, N, N |
| *yafF* | 0.4 | No information about this protein was found by a literature search conducted on April 19, 2017. | E,-, N |
| *yagG* | 0.6 | *yagGH* is predicted to be a member of the XylR regulon; its products may mediate transport (YagG) and hydrolysis (YagH) of xylooligosaccharides; putative XylR and CRP binding sites are identified upstream of *yagGH*. | E,-, N |
| *yceQ* | 0.2 | No information about this protein was found by a literature search conducted on July 12, 2017. | E, E, N |
| *ydiL* | 0.2 | No information about this protein was found by a literature search conducted on April 7, 2017. | E, N, N |
| *yhhQ* | 0.4 | YhhQ is an inner membrane protein implicated in the uptake of queuosine (Q) precursors - 7-cyano-7-deazaguanine (preQ0) and 7-aminomethyl-7-deazaguanine (*preQ1*) - for Q salvage. Q-modified tRNA is absent in $\Delta$*queD* and $\Delta$*queD* $\Delta$*yhhQ* strains grown in minimal media with glycerol; Q-modified tRNA is detected when a $\Delta$*queD* strain is grown in minimal media plus 10 nM *preQ0* or *preQ1* but is absent when a $\Delta$*queD* $\Delta$*yhhQ* strain is grown under these conditions. *yhhQ* expressed from a plasmid restores the presence of Q-modified tRNA in a $\Delta$*queD* $\Delta$*yhhQ* strain. | E,-, N |
| *yibJ* | 0.3 | No information about this protein was found by a literature search conducted on July 9, 2018. | E, N, N |
| *ymfK* | 0.4 | YmfK is a component of the relic lambdoid prophage e14 and is likely the SOS-sensitive repressor. It is similar to the P34 gene of the *Shigella flexneri* bacteriophage SfV and belongs to the LexA group of SOS-response transcriptional repressors. | E, E, E |

Table 3.3: **Below-threshold essential genes identified in *E. coli*.** This table describes the message numbers and annotations for essential genes that we estimated to have expression below the threshold of one message per cell cycle. However, in the final column, we show classifications from three different studies. Only one of the identified genes, *ymfK*, was consistently defined as essential.

### *3.10.1 Results: RLTO model predicts non-canonical noise scaling*

The predicted scaling of the optimal translation efficiency with message number has many important implications, including on the global characteristics of noise. Based both on theoretical and experimental evidence, it is widely claimed that gene-expression noise should be inversely proportional to protein abundance [20, 2]:

$$\mathrm{CV}_p^2 \propto \mu_p^{-1}, \tag{3.134}$$

for low-expression proteins, as observed in *E. coli* [2]; however, the more fundamental prediction is that the noise is inversely proportional to the message number:

$$\mu_p = \mu_m \varepsilon, \tag{3.135}$$

$$\mathrm{CV}_p^2 = \tfrac{\ln 2}{\mu_m}. \tag{3.136}$$

In *E. coli*, the translation efficiency is roughly constant (*i.e.* $\hat{\mu}_p \propto \hat{\mu}_m$, Fig. 3.4D) and therefore Eq. 3.136 is consistent with the canonical noise model (Eq. 3.134). However, in eukaryotes, the translation efficiency grows with message number (*i.e.* $\hat{\mu}_p \propto \hat{\mu}_m^2$, Fig. 3.4BC). If we substitute this proportionality into Eq. 3.136, we predict the non-canonical noise scaling:

$$\mathrm{CV}_p^2 \propto \mu_p^{-1/2}, \tag{3.137}$$

for eukaryotic cells.

### *3.10.2 Methods: Analysis of gene expression noise*

The quantitative model for gene expression noise includes multiple contributions:

$$\mathrm{CV}_p^2 \approx \tfrac{1}{\mu_p} + \tfrac{\ln 2}{\mu_m} + c_0, \tag{3.138}$$

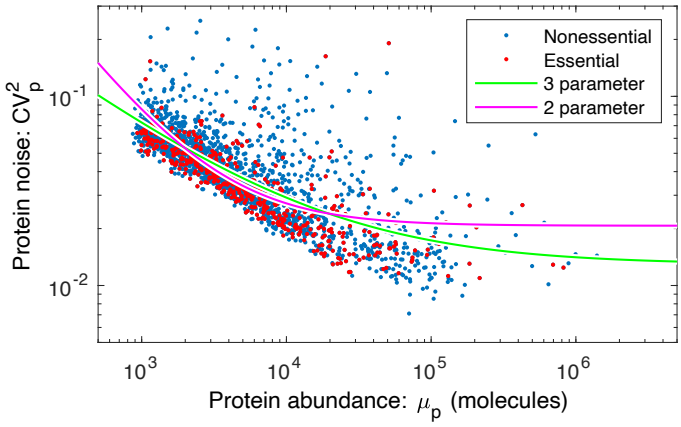



Figure 3.16: **Yeast noise fit against canonical noise model, with a noise floor.** Yeast noise data fit with the 2- (null hypothesis with $\mu_p^{-1}$ dependence) and 3- parameter ($\mu_p^a$) models. The two-parameter model corresponds to the canonical noise model (Eq. 3.134) and fails to quantitatively fit the data.

where the first term can be understood to represent the Poisson noise from translation, the second term the Poisson noise from transcription, and the last term, $c_0$, is called the *noise floor* and is believed to be caused by the cell-to-cell variation in metabolites, ribosomes, and polymerases *etc.* [22, 23].

In the main text of the paper, we have ignored the role of the noise floor in the analysis of noise in yeast. Unlike *E. coli*, where the noise floor is high ($\text{CV}_p^2 = 0.1$) and is determinative of the noise associated with almost all essential genes [2, 22, 23], in yeast the noise floor is much lower ($\text{CV}_p^2 = 0.01$) and therefore affects only genes with the highest expression.

In this section, we will consider models that include the noise floor, since its presence can make the noise scaling more difficult to interpret. To determine if the scaling of the noise is consistent with the canonical assumption that the noise is proportional to $\mu_p^{-1}$ for low expression, we will consider two competing empirical models for the noise (Fig. 3.16). In the null hypothesis, we will consider a model:

$$\eta_0(\mu_p; b, c) = \tfrac{b}{\mu_p} + c, \tag{3.139}$$

and an alternative hypothesis with an extra exponent parameter $a$:

$$\eta_1(\mu_p; a, b, c) = \frac{b}{\mu_p^a} + c. \tag{3.140}$$

We will assume that $\mathrm{CV}_p^2$ is normally distributed about $\eta$ with unknown variance $\sigma_\eta^2$.

In this context, a maximum likelihood analysis is equivalent to least-squares analysis. Let the sum of the squares be defined:

$$S_I(\boldsymbol{\theta}) \equiv \sum_i [\mathrm{CV}_{p,i}^2 - \eta_I(\mu_{p,i}; \boldsymbol{\theta})]^2, \tag{3.141}$$

for model $I$ where $\boldsymbol{\theta}$ represents the parameter vector. The maximum likelihood parameters are

$$\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} S_I(\boldsymbol{\theta}), \tag{3.142}$$

with residual norm:

$$\hat{S}_I = S_I(\hat{\boldsymbol{\theta}}). \tag{3.143}$$

To test the null hypothesis, we will use the canonical likelihood ratio test with the test statistic:

$$Z \equiv 2 \ln \frac{q_1}{q_0}, \tag{3.144}$$

where $q_0$ and $q_1$ are the likelihoods of the null and alternative hypotheses, respectively. Wilks' theorem states that $Z$ has a chi-squared distribution of dimension equal to the difference of the dimension of the alternative and null hypotheses ($3 - 2 = 1$).

*Hypothesis test I*

In our first analysis, we will estimate the variance directly. We computed the mean-squared difference for successive $\mathrm{CV}_p^2$ values, sorted by mean protein number $\mu_p$. The variance estimator is

$$\hat{\sigma}_\eta^2 = \tfrac{1}{2} \left\langle (\mathrm{CV}_{p,i}^2 - \mathrm{CV}_{p,i+1}^2)^2 \right\rangle_i = 6.3 \times 10^{-4}, \tag{3.145}$$

where the brackets represent a standard empirical average over gene $i$ for the $\mu_p$-ordered gene $\mathrm{CV}_p^2$ values. The test statistic can now be expressed in terms of the residual norms:

$$Z = (\hat{S}_1 - \hat{S}_2)/\hat{\sigma}_\eta^2, \tag{3.146}$$

$$= 3.3 \times 10^4, \tag{3.147}$$

which corresponds to a p-value far below machine precision. We can therefore reject the null hypothesis.

*Hypothesis test II*

In a more conservative approach, we can use maximum likelihood estimation to estimate the variance of each model independently as a model parameter. In this case, the test statistic can again be expressed in terms of the residual norms:

$$Z = N \ln \tfrac{\hat{S}_1}{\hat{S}_2}, \tag{3.148}$$

$$= 1.6 \times 10^2, \tag{3.149}$$

where $N$ is the number of data points. (Details of derivation are in Sec. 3.10.2.) In this case, the p-value can be computed assuming the Wilks' theorem (*i.e.* the chi-squared test):

$$p = 6 \times 10^{-36}, \tag{3.150}$$

again, strongly rejecting the null hypothesis.

*Maximum likelihood estimates of the parameters*

In the alternative hypothesis, the maximum likelihood estimate (MLE) of the empirical noise model (Eq. 3.140) parameters are (Fig. 3.16):

$$a = 0.57 \pm 0.02, \tag{3.151}$$

$$b = 3.0 \pm 0.5, \tag{3.152}$$

$$c = 0.013 \pm 0.001, \tag{3.153}$$

where the parameter uncertainty has been estimated using the Fisher Information in the usual way using the MLE estimate of the variance [86, 87].

*Details: Statistical details MLE estimate of the variance*

The minus-log-likelihood for the normal model $I$ is:

$$h_I(\hat{\boldsymbol{\theta}}, \sigma^2) = \tfrac{N}{2} \ln 2\pi\sigma^2 + \tfrac{1}{2\sigma^2} \hat{S}_I, \tag{3.154}$$

where $\hat{S}_I$ is the least-square residual. We then minimize $h_I$ with respect to the variance $\sigma^2$:

$$\partial_{\sigma^2} h|_{\hat{\sigma}^2} = 0, \tag{3.155}$$

to solve for the MLE $\hat{\sigma}^2$:

$$\hat{\sigma}^2 = \tfrac{1}{N} \hat{S}_I. \tag{3.156}$$

Next we evaluate $h$ at the variance estimator:

$$h_I(\hat{\boldsymbol{\theta}}, \hat{\sigma}^2) = \tfrac{N}{2} \left[ \ln 2\pi \tfrac{\hat{S}_I}{N} + 1 \right]. \tag{3.157}$$

The test statistics can be written in terms of the $h$'s:

$$Z = 2h_0(\hat{\boldsymbol{\theta}}, \hat{\sigma}^2) - 2h_1(\hat{\boldsymbol{\theta}}, \hat{\sigma}^2), \tag{3.158}$$

$$= N \ln \frac{\hat{S}_0}{\hat{S}_1}, \tag{3.159}$$

which can be evaluated directly in terms of the residual norms for the null and alternative hypotheses.

### *3.10.3 Results: Non-canonical noise scaling is observed in yeast.*

To test the RLTO model predictions for noise scaling, we reanalyze the dataset collected by Newman *et al.*, who performed a single-cell proteomic analysis of yeast by measuring the abundance of fluorescent fusions by flow cytometry [1]. Since the competing models (Eqs. 3.134 and 3.137) make different scaling predictions, we first apply a statistical test to determine whether the observed scaling is consistent with the canonical model (Eq. 3.134). We consider the null hypothesis of the canonical model (Eq. 3.139) and the alternative hypothesis with an unknown scaling exponent (Eq. 3.140). To test the models, we perform a null hypothesis test. (A detailed description of the statistical analysis, which includes the contribution of the noise floor, is given in the Sec. 3.10.) We reject the null hypothesis with a p-value of $p = 6 \times 10^{-36}$. The observed scaling exponent is $\hat{a} = -0.57 \pm 0.02$, which is close to our predicted estimated exponent from the RLTO model ($-\frac{1}{2}$).

### *3.10.4 Methods: Parameter-free prediction of noise from protein-message relation.*

By combining the noise model (Eq. 3.136) with a protein-message abundance relation, the relation between protein abundance and noise can be predicted without additional fitting parameters. (We call this prediction *parameter-free* since, although a parameter is fit when determining the protein-message abundance, once this relation has been established, no new parameters are fit in order to predict the noise.) To test this prediction, we will compare

three competing models: (i) the RLTO model, (ii) an empirical protein-message abundance model, and (iii) the constant-translation-efficiency model.

*Estimating protein number ($\mu_p$) for the noise analysis*

The protein abundance data for yeast grown in YEPD media and measured with flow cytometry fluorescence [1] were given in arbitrary units (AU). In order to convert from AU to protein number, the fluorescence values were rescaled by comparing with mass-spectrometry protein abundance data for yeast grown in YNB media [3]. Since the protein abundance from mass-spectrometry was given in terms of intensity, the intensity values were first rescaled by the total number of proteins in yeast, $5 \times 10^7$. (See Sec. 3.9.1.) The mass-spectrometry protein data was thresholded at 10 proteins, based on the assumption that the noise of the data for 10 and fewer proteins makes the data unreliable. Next, the log of the fluorescence protein abundance in AU as a function of the log of thresholded mass-spectrometry protein abundance was fit as a linear function with an assumed slope of 1 to find the offset, 3.9, (Fig. 3.17) which corresponds to a multiplicative scaling factor. We then used that offset value to rescale the fluorescence data from AU to protein number. We also compared to yeast grown in SD media [1] and found a similar offset result.

*Empirical models for yeast gene expression*

To generate the empirical model for protein number as a function of message number, we used protein abundance data from Newman *et al.* [1], re-scaled to estimate protein number (Sec. 3.10.4) and transcriptome data from Lahtvee *et al.* [88], re-scaled to estimate message number (Sec. 3.9.2).

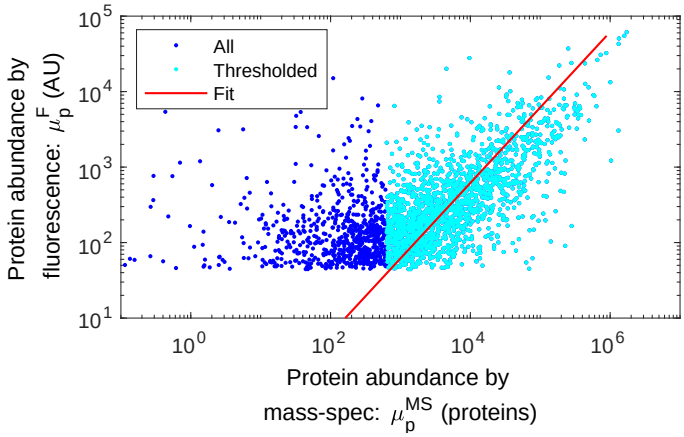



Figure 3.17: **Fit to rescale fluorescence intensity to protein number.** Protein abundance from flow cytometry fluorescence [1] as a function of mass-spectrometry scaled abundance [3]. The mass-spectrometry data was thresholded at 10 proteins, and then a linear fit was performed to find the multiplicative offset of 3.9, which was used to convert protein fluorescence AU to number.

*Empirical model for protein number*

We initially fit the empirical model for protein number,

$$\mu_p = C_0 \mu_m^{\alpha_0}, \tag{3.160}$$

to the data using a standard least-squares approach; however, the algorithm led to a very poor fit since it does not account for uncertainty in both independent and dependent variables. We therefore used an alternative approach [89], which assumes comparable error in both variables. The model parameters are:

$$\alpha_0 = 2.1 \pm 0.04, \tag{3.161}$$

$$C_0 = 8.0 \pm 1.0, \tag{3.162}$$

where the uncertainties are the estimated standard errors. The result of the empirical model fit is shown in Fig. 3.18A, along with the constant-translation-efficiency model, and the RLTO model.

*Empirical model for message number*

For the prediction of the coefficient of variation, it is useful to invert Eq. 3.160 to generate a model for message number as a function of protein number:

$$\mu_m = C_0^{-1/\alpha_0} \mu_p^{1/\alpha 0}, \tag{3.163}$$

$$= C_1 \mu_p^{\alpha_1}, \tag{3.164}$$

where the last line defines two new parameters: a coefficient $C_1$ and an exponent $\alpha_1$. The resulting parameters and uncertainties are:

$$\alpha_1 \equiv 1/\alpha_0, \tag{3.165}$$

$$= 0.48 \pm 0.01, \tag{3.166}$$

$$C_1 \equiv C_0^{-1/\alpha_0}, \tag{3.167}$$

$$= 0.37 \pm 0.02, \tag{3.168}$$

where the uncertainties are the estimated standard errors.

*Empirical model for translation efficiency*

To generate an empirical model for translation efficiency, we started from the empirical model for protein number (Eq. 3.160), and then use Eq. 3.135 to relate protein number,

message number, and translation efficiency:

$$\begin{aligned} \varepsilon &= \tfrac{\mu_p}{\mu_m}, && (3.169)\\ &= C_0 \mu_m^{\alpha_0 - 1}, && (3.170)\\ &= C_2 \mu_m^{\alpha_2}, && (3.171) \end{aligned}$$

where the last line defines two new parameters: a coefficient $C_2$ and an exponent $\alpha_2$. The resulting parameters and uncertainties are:

$$\begin{aligned} \alpha_2 &= \alpha_0 - 1, && (3.172)\\ &= 1.07 \pm 0.04, && (3.173)\\ C_2 &= C_0, && (3.174)\\ &= 8.0 \pm 1.0, && (3.175) \end{aligned}$$

where the uncertainties are the estimated standard errors.

*Empirical model for the coefficient of variation*

To generate an empirical model for the coefficient of variation, we started from the empirical model for message number (Eq. 3.164), and then substitute this into the statistical model prediction for $\mathrm{CV}_p^2$ (Eq. 3.136):

$$\begin{aligned} \mathrm{CV}_p^2 &= \tfrac{\ln 2}{\mu_m}, && (3.176)\\ &= C_0^{1/\alpha_0} \ln 2 \cdot \mu_p^{-1/\alpha_0}, && (3.177)\\ &= C_3 \mu_p^{\alpha_3}, && (3.178) \end{aligned}$$

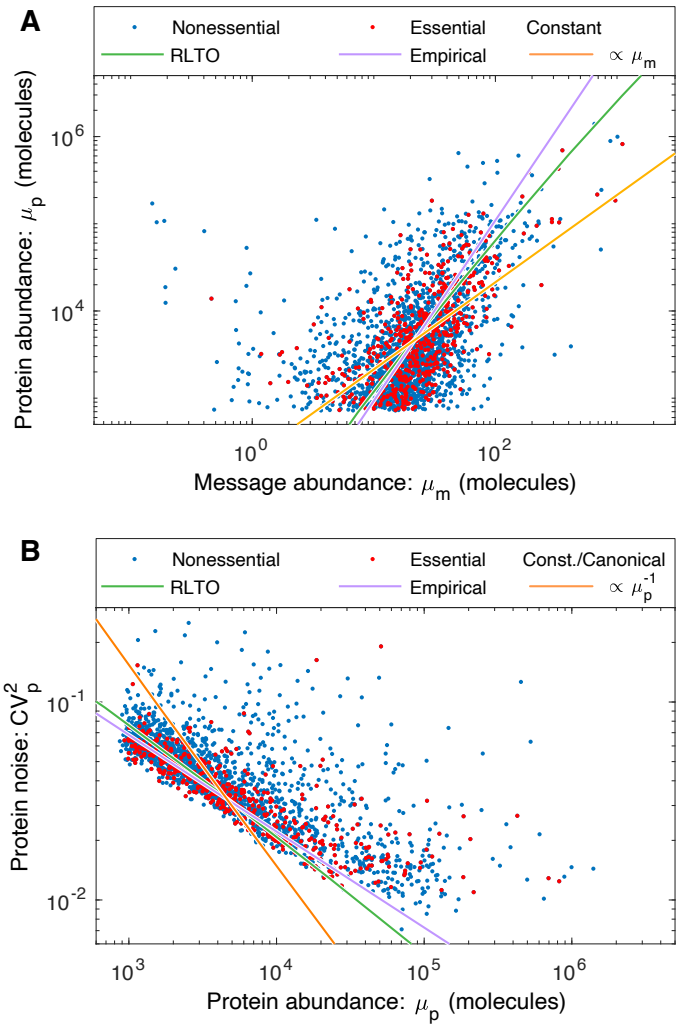



Figure 3.18: **Load balancing predicts the scaling of noise. Panel A: Three competing models for protein abundance in yeast.** The empirical model (purple) fits the slope and the y offset. The RLTO (green) and constant-translation-efficiency (orange) models fit a parameter corresponding to the y offset only. As discussed in the analysis of the proteome fraction, the RLTO model qualitatively captures the scaling of the protein abundance with message number better than the constant translation efficiency model; however, the predicted fit does not correspond to the optimal power law, which is represented by the empirical model. The protein abundance has a cutoff near $10^1$ due to autofluorescence [1]. **Panel B: Predictions of the noise-protein abundance relation.** Using each competing protein abundance model, the noise-protein abundance relation can be predicted using Eq. 3.136. The canonical noise model (Eq. 3.134) fails to capture even the scaling of the noise. In contrast, both the RLTO and empirical models quantitatively predict both the scaling and magnitude of the noise. The empirical model has the highest performance, presumably due to its two-parameter fit to the protein abundance in Panel A. A fit accounting for the noise floor is shown in Fig. 3.16.

where the last line defines two new parameters: a coefficient $C_3$ and an exponent $\alpha_3$. The resulting parameters and uncertainties are:

$$\alpha_3 \equiv -1/\alpha_0, \tag{3.179}$$

$$= -0.48 \pm 0.01, \tag{3.180}$$

$$C_3 \equiv C_0^{1/\alpha_0} \ln 2, \tag{3.181}$$

$$= 1.9 \pm 0.1, \tag{3.182}$$

where the uncertainties are the estimated standard errors.

### *3.10.5 Results: Parameter-free prediction of noise-abundance in yeast*

The fit of the competing protein-message abundance models are shown in Fig. 2.9A. Using each model, we can now predict the relation between protein abundance and noise without additional fitting parameters. The predictions of the three competing models are compared to the experimental data in Fig. 2.9B.

In both its ability to capture the protein abundance and predict the noise, the RLTO model vastly outperforms the constant-translation-efficiency model. The purely empirical model that best captures the protein abundance data, due to directly fitting both the y-offset and slope, also performs best in predicting the noise. It is important to emphasize that the prediction of the noise in all models is non-trivial since there are no free parameters fit, once the protein abundance relation is determined. We therefore conclude that the noise model (Eq. 3.136) quantitatively predicts the observed noise from the message number and that eukaryotic noise has non-canonical scaling due to load balancing.

### *3.10.6 Discussion: Implications of noise*

What are the biological implications of gene expression noise? Many important proposals have been made, including bet-hedging strategies, the necessity of feedback in gene regulatory

networks, *etc.* [11]. Our model suggests that robustness to noise fundamentally shapes the central dogma regulatory program. With respect to message number, the one-message-rule sets a lower bound on the transcription rate of essential genes. (See Fig. 3.6B.) With respect to protein expression, robustness to noise has two important implications: Protein overabundance significantly increases protein levels above what would be required in the absence of noise and therefore reshapes the metabolic budget. (See Fig. 3.6A.) Robustness to noise also gives rise to load balancing, the proportionality of the optimal transcription and translation rates. (See Fig. 3.6C.) Not only does robustness to noise affect central dogma regulation, but there is an important reciprocal effect: Load balancing changes the global scaling relation between noise and protein abundance. (See Fig. 3.18B.)

# Chapter 4

# OMNISEGGER: ENABLING ANALYSIS OF DIVERSE CELL MORPHOLOGIES

Timelapse microscopy is a powerful tool which allows for the observation of cell growth over time. Measurements of various quantities from the growing cells (cytometry) can be performed by extracting data from the micrographs. For example, a protein can be labeled a fluorescent protein (such as green fluorescent protein (GFP)) to track the amount of protein expressed over time by individual cells. In addition, because cell growth is exponential, the complexity of the data grows exponentially. There exist various pipeline packages to handle such measurements, and in this chapter I discuss the advancements and advantages of my pipeline package called OmniSegger.

OmniSegger is a pipeline analysis tool which combines several packages developed in the Wiggins Lab: SuperSegger (2016), Omnipose (2022), and Bactrack (2024). The analysis pipeline is as follows: image registration (alignment), cell segmentation, frame-to-frame linking/tracking, cytometry, and data output. OmniSegger greatly reworks the cell segmentation and linking/tracking steps to improve robustness and accuracy, and makes their outputs compatible with the original SuperSegger's pre-existing code.

The improvements to handling diverse morphologies and accuracy of subcellular measurements enables OmniSegger to demonstrate protein overabundance for essential-gene knockout experiments in *A. baylyi*[9].

## *4.1 Introduction: The original SuperSegger*

SuperSegger is a software originally created to analyze protein localization for thousands of cells [90]. In particular, the Wiggins Lab performed proteome-wide timelapse measurements

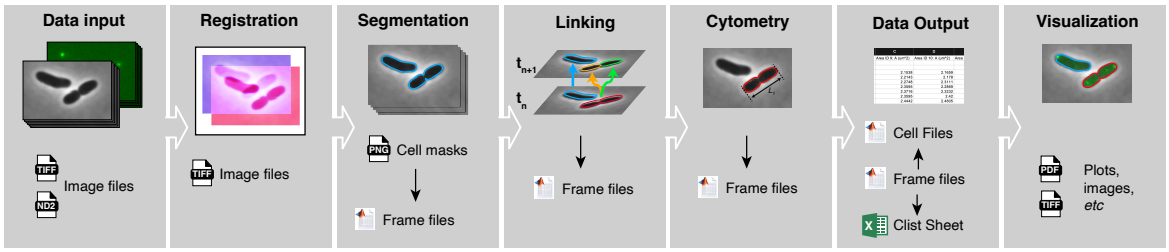



Figure 4.1: **OmniSegger pipeline schematic. Data input:** Multi-dimensional image data is loaded from image files. **Registration:** Images are aligned to remove stage drift. **Segmentation:** For each position and time-point, the first channel image is segmented to generate a the cell masks, which are saved to a png format file. These masks are then incorporated into a frame file, which is a composite data file containing all image information (all channels and cell masks). The cell masks png file is user editable. **Linking:** Cell masks from successive time points are then linked to form cell trajectories, including cell division. These links are corrected in time and saved into the frame files. **Cytometry:** Cell cytometry information for each cell is computed from the image information in the frame files. **Data output:** The output data is sliced into three different output formats: The *frame files* contain all information, including images, grouped by frame (*i.e.* all cells per time-point and x-y position). The *cell files* contain all information, including images, grouped by cell (*i.e.* all time-points per cell). The *clist file* contains all cytometry information (no image information) grouped per x-y position. **Visualization:** The package also contains numerous visualization tools which use the output data to generate figures, images, and plots.

of the ASKA K-12 ORF-clone collection. In these measurements, proteins were tracked at the single-molecule level; each protein was tagged with green fluorescent protein (GFP) and imaged in a timelapse with both phase-contrast and fluorescence microscopy [91, 92]. Features to analyze non-diffuse foci were also introduced and enabled precise analysis of the replication machinery in *E. coli* [93]. Without the automation provided by computer software, manual data analysis would take thousands of human hours. SuperSegger established the general analysis pipeline for OmniSegger, which I will describe in brief, and is summarized in the schematic Fig. 4.1.

### *4.1.1 Image registration*

Stage drift can occur during data acquisition, with possible causes such as thermal expansion or evaporation of the pad. The result of stage drift is that initial frames will have an x-y offset in comparison to final frames, which can lead to difficulty in tracking cells and microcolonies. To account for the effect of such drift, sub-pixel image alignment[1] is performed using a cross-correlation registration algorithm [94]. The resulting frame offsets are saved in `cropbox.mat`.

### *4.1.2 Cell segmentation*

Strictly speaking, the process of cell segmentation describes labeling an image to declare which pixels are cells, and which pixels are not cells–and therefore labeled as background. The original SuperSegger segmentation algorithm has a unique approach that combines thresholding and machine-learning.

The first step is the pre-processing of the phase contrast image by smoothing and normalization. Next, the algorithm thresholds the image and applies a filter to compensate for artifacts inherent to phase-contrast imaging. The image is then thresholded again to obtain the microcolony mask image. The algorithm applies a watershed operation to the microcolony mask, which oversegments the masks. The watershed regions provide possible cell boundaries, and the true and false segments are determined using a trained neural network. Next, a scoring function optimizes regions with another trained neural network in order to define the final cell boundaries. The region scoring function can override the decision of true or false boundaries as predicted from the previous step. The inclusion of the neural network steps allows the segmentation algorithm to reduce segmentation errors traditionally resulting from thresholding and watershedding. The SuperSegger paper describes the process in more detail [90]. Once the cell boundaries have been defined, for each frame, each cell region

[1]The technical term 'registration' is often used in imaging processing, but the SuperSegger software refers to this step as alignment.

is labeled with 1...$N$ from the top-left to the bottom-right of the image using MATLAB's `bwlabel` function.

Because `bwlabel` requires objects to *not* be connected in order to be labeled as distinct, the original segmentation algorithm has at least one-pixel boundaries between cells, which is labeled as background. This presents an issue because growing microcolonies of cells almost always are dense and have cells touching, which means the one-pixel boundary is false and results in inaccurate segmentation.

### *4.1.3 Linking/tracking*

For time-dependent analysis, cell regions must be linked in consecutive frames in order to determine cell lineages and division events. The linking algorithm creates a frame-to-frame cost for each region to be mapped to regions in the successive frames. The cost function takes into account region overlap, distance between regions fit with centroids, and change in area. The algorithm also searches for linking errors by comparing forward and reverse mappings (cell mappings in the preceding and following frames); if errors occur, SuperSegger will attempt to re-link cells or re-segment the local area. This combination of linking and segmentation error resolution allows the algorithm to target segmentation errors such as two-to-one merging.

Considering that unusual cell morphologies, such as filamentous cells, may not have overlap in sequential frames, are not well fit by centroids, and can have unpredictable change in area, results from the original SuperSegger linking cost algorithm are likely to be highly biased toward typical, rod-shaped cells.

### *4.1.4 Cytometry*

Once cells have been linked, unique IDs can be assigned to each cell. These cell IDs are distinct from cell region numbers. IDs persist throughout the entire timelapse, while cell region number is unique from frame to frame. Then, cell properties such as cell length, area,

birth and division (death) frame, and mother and daughter IDs, are calculated for each cell. These properties are crucial to experimental analysis, and their values depend heavily on the preceding steps in the SuperSegger pipeline.

If the timelapse includes fluorescence images, SuperSegger calculates statistics such as the average intensity of each cell, or the location of foci. First, the mean background intensity is subtracted from each image. Next, an *image curvature* filter is applied to eliminate noise and background from autofluorescence or non-specifically-bound fluorescent probes (C. Brennan, unpublished). The foci are then determined from the maxima of the curve-filtered image.

#### *4.1.5 Data output and visualization*

The data calculated in the previous steps are saved for each cell (cell files), for all cells in each frame (seg files), and for all cells in all frames. The latter was termed the 'clist', and the data retained in a `clist.mat` file.

The images and segmentation results can be visualized using the Viewer GUI, which includes various options to automate plotting of variables or to filter out ("gate") data.

### *4.2 From SuperSegger to OmniSegger*

OmniSegger skips the original cell segmentation algorithm, and runs the Omnipose package. The SuperSegger mask variables are replaced by the masks output by Omnipose. Omnipose segmentation is much more robust to diverse cell morphologies and, as a result, provides the foundation for timelapse analysis by OmniSegger. As previously mentioned in Sec. 4.1.3, there are various limitations to the original tracking algorithm. I have implemented a form of the Bactrack package into OmniSegger, which is offered as an alternative option for analysis of timelapses for non-rod shaped cells. In addition, the data visualization features have also been updated to accommodate improved segmentation results.

### *4.2.1 Omnipose segmentation*

Omnipose is a deep neural network (DNN)-based algorithm for cell segmentation [95]. It is originally based off of the Cellpose algorithm; however, it makes several changes which significantly improve performance. The Cellpose architecture outputs a vector field defined as the gradient of the heat distribution from the median pixel coordinate of the cell–this was referred to as the flow field. Next, Cellpose would apply Euler integration on the flow field to the cell center to determine the cell mask.

Depending on the shape of the cell (for example, consider a curve-shaped cell), the median pixel coordinate might fall outside the cell mask and in those cases, Cellpose would project the coordinate to the nearest pixel along the cell boundary. The cases where the cell center was ill-defined often occurred for longer cells, which typically have curvature, thus resulting in multiple possible projections of the median cell coordinate; the cells would often be segmented into multiple smaller masks (oversegmentation).

While Cellpose used the gradient of the heat distribution, Omnipose redefined the flow field as the gradient of the distance of each pixel from the cell boundary (gradient of the distance field). This method allows the flow field predictions to be independent of cell morphology, whereas Cellpose's method often failed for long cells. Omnipose also introduced suppressed Euler integration, where each time step has a suppression factor, leading to improved pixel clustering to generate the cell mask.

In addition, the bacterial phase-contrast Omnipose model (`bact_phase_omni`) was trained with a dataset of about 27,500 wild-type, mutated, and antibiotic-treated cells of various species, with numerous imaging conditions and augmentations, allowing the network to output a robust trained model. Furthermore, much of the Omnipose dataset was hand-annotated based on fluorescence signal of the cell membrane, allowing for more precise calling of the division time as compared to simply relying upon ambiguous phase-contrast signal. The fluorescence signal serves as an external marker to validate when cell division has occurred.

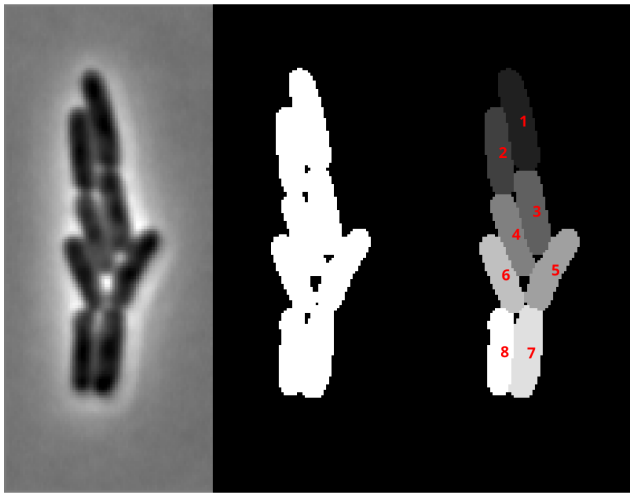



Figure 4.2: **Semantic vs Instance segmentation.** Left: Phase contrast image of cells. Middle: Semantic segmentation of image with all cells segmented as one label, often leading algorithms to have a one-pixel boundary between cells in contact. Right: Instance segmentation; each cell is assigned a unique label.

Even with the improvements to cell segmentation via deep learning algorithms, many algorithms currently perform semantic segmentation which produces binary cell masks: 0 indicating background and 1 indicating cells. As a consequence, the resulting masks require a minimum 1-pixel separation gap between cells. A unique feature of Omnipose is that its mask predictions are instance segmentations–each cell mask is assigned a unique label, and can therefore be distinguished even if touching, as is often the case when imaging dense microcolonies. The difference between semantic and instance segmentation is shown in Fig. 4.2. In addition, if cells are thin, the receded boundaries from semantic segmentation result in inconsistent cell widths and exclude a significant fraction of the cell area.

The use of the distance field, the suppressed Euler integration, instance segmentation, and training a phase-contrast model on a diverse dataset allows Omnipose to be a leading segmentation tool. Omnipose is much more robust to various imaging conditions and cell morphologies than other cell segmentation algorithms.

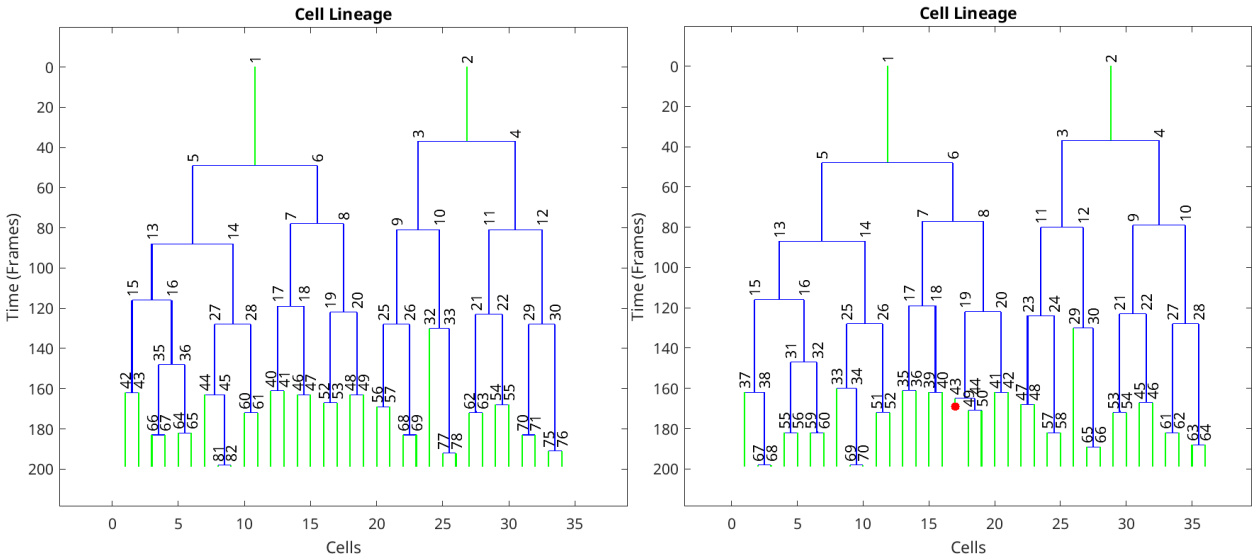
### *4.2.2 Bactrack linking*

Motivated by observations where the original linking algorithm greatly failed for filamentous cells, I introduced the problem to Sherry Yang, who developed the Bactrack package [96] in Python, which provided much improved tracking performance. Bactrack is a cell tracking tool which uses hierarchical segmentation and mixed-integer programming optimization. The hierarchical segmentation approach of Bactrack was inspired by ultrack [97]; Omnipose flow field inputs are used to create hierarchical segmentations from low to high resolution.

The hierarchical segmentations contain multiple possible segmentations at various levels of detail, and each segment can be represented as a node, while frame-to-frame links can be represented by edges. The resulting nodes and edges form a directed graph, where the direction follows increasing time. The edges are assigned weights based on overlap or distance–the OmniSegger implementation uses overlap, as it had more reliable results. The node and edge branches in the directed graph are then removed and optimized using the branch-and-cut technique by an MIP solver algorithm. This process is represented schematically in Fig. 4.4.

The optimization of the graph determines the frame-to-frame linking of cells. In particular, the constraints for optimization are one-to-two division (persistent in time because the graph is directed) for binary cell fission. We also briefly considered allowing one-to-multiple divisions or two-to-one merging events based on our observations of a filamentous dataset; however, permitting these constraints slowed solver performance and was decided to be niche for a typical application. In summary, given a graph of possible segmentations, a matrix of linking costs, and the constraint that cells only divide, Bactrack optimizes the final graph of cell segmentations. Once the graph has been optimized, Bactrack outputs *both* masks in Omnipose labeled-mask format, and linking results as a Pandas dataframe.

Bactrack allows the option to use one of three different MIP solvers: HiGHS [98] using the `scipy.optimize.milp` Python wrapper, and CBC [99] and Gurobi [100] through Python-MIP [101], though Gurobi and HiGHS are the fastest [102]. OmniSegger only implements HiGHS and Gurobi.




(a) Lineage generated by OmniSegger from SuperSegger linking & Omnipose segmentation.

(b) Lineage generated by OmniSegger from Bactrack linking & segmentation.

Figure 4.3: **Bactrack linking results in different calling of divisions.** Lineage trees generated for a growing microcolony differ slightly in calling of division times, for example cells from progenitor 3 are determined to divide later based on Bactrack segmentation and linking. The division results from Omnipose and Bactrack should be compared for biological accuracy.

While in theory, the masks that result from the optimized hierarchical segmentation should be more accurate than Omnipose segmentation, in practice, I believe that the resulting Bactrack masks were observed to be less biologically accurate; the Bactrack masks should be checked if they call division too early or too late compared to Omnipose masks when tested on the same dataset (see Fig. 4.3). Therefore, OmniSegger inputs Omnipose masks into Bactrack to generate only linking results. Furthermore, the error resolution code in OmniSegger relies on checking linking results. If the linking results are now determined by Bactrack, error resolution by OmniSegger is not possible and thus the version of OmniSegger with Bactrack implementation does not correct the underlying masks. This can be most dramatically demonstrated by a high time-resolution timelapse, which can suffer from inconsistent determination of division from frame-to-frame.

The linking results contain the mappings between a 'source' ($t$) frame and a 'target' ($t+1$) frame. The format of results is as follows: the first column lists the source frame starting at frame 0. The second column lists the label IDs in the source frame that have mappings to the target frame. The third column contains the label IDs in the target frame that are mapped from the source frame. In addition, the fourth and fifth columns are source frame cell areas and target frame cell areas, which are used for tracking error calculations. Note that if a cell in a source frame does not have a mapping in the target frame (*i.e.*, a stray cell which appears in the source frame but disappears in the target frame), there will be no entry for that cell ID in the Bactrack linking results; SuperSegger fills these links in `fillBactrackLinks.m`.

The linking results are then converted from Pandas dataframe into comma-separated values (csv). The csv file is read into MATLAB during the linking stage in OmniSegger and then converted into SuperSegger linking format with forward and reverse mappings.

Bactrack is implemented with OmniSegger through the following GitHub branches: Bactrack branch superSeggerDev, OmniSegger branch bactrackdev. I recommend trying this version when the main branch of OmniSegger fails to track unusual morphologies for time-lapses at medium to low frame rates.

### *4.2.3 Improvements in data visualization*

We introduce new data visualization ideas and improvements to generalize for diverse cell morphologies: 1) cell outlines, 2) medoid for cell ID display, and 3) figure making tools.

1. **Cell outlines.** The displayed cell outlines used to be calculated by dilating the mask, but are now replaced using a more robust cell perimeter function directly based on the mask (`getperim.m`).

2. **Medoid of cell skeleton for cell ID display.** Many cell analysis software packages, including the original SuperSegger, determine the centroids of cell masks as an approximation for the cell 'center' and uses the centroid in order to perform measurements. A

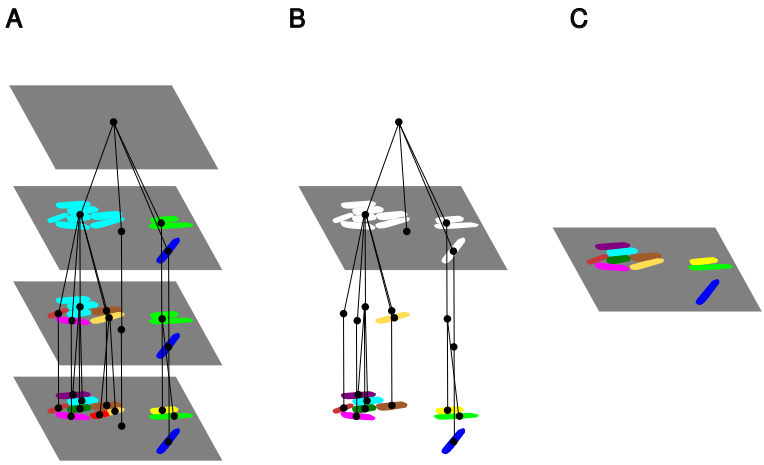



Figure 4.4: **Schematic of hierarchical segmentation: Panel A:** initial hierarchical segmentation, which is generated from Omnipose flow fields as implemented by Bactrack, **Panel B:** pruning of the hierarchical segmentation, and **Panel C:** final resultant segmentation mask image. Figure adapted from [7].

centroid is a point calculated from the mean position of all data points, *i.e.*, the mean pixel position of the cell mask. For a given number of $N$ pixels with the coordinate of pixel $i$ given by $x_i, y_i$, the centroid $(\bar{x}, \bar{y})$ is simply calculated by:

$$\bar{x} = \frac{\sum_i x_i}{N}, \tag{4.1}$$

and

$$\bar{y} = \frac{\sum_i y_i}{N} \tag{4.2}$$

In MATLAB, this is calculated using the `regionprops` function.

The centroid approximation for the cell 'center' fails for more diverse morphologies, for example with filamented cells that contain curvature, where the centroid would fall outside the mask. Rather than calculating the centroid, OmniSegger determines the *medoid*. The medoid is the position of a pixel contained in the cell mask which has the minimum sum of distances from every other pixel in the mask:

$$\min d_i = \min \sum_{j \neq i} \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2} \tag{4.3}$$

Furthermore, I find that restricting the medoid to the skeleton results in an even more intuitive result for the cell 'center' (see `find_medoid.m`). Using the medoid of the skeleton as the cell 'center' displays the cell IDs much more intuitively than the centroid, applicable to both rod-shaped and other morphologies (see Fig. 4.5).

3. **Figure making tools.** Generating publication-quality figures is important for disseminating results of scientific research. OmniSegger includes new functions for generating figures: `getFamily`, which determines a cell lineage from a progenitor cell, `drawCellSpline`, which draws vectorized outlines of cell boundaries on a cell-by-cell

basis, and `makeMosaic`, which generates a mosaic image of selected frames from a timelapse, with the ability to display fluorescence and cell outline overlays.

### *4.2.4 Modified cell length measurement*

In SuperSegger, the cell length is calculated using MATLAB's `regionprops` function. `regionprops` fits a cell as an ellipse, and we expect this approximation to be poor for morphologies with curvature. Therefore, we replaced the cell length measurement with a rod length in `rodGeom`.

The area $A$ of a cross-section of a 3D rod is approximated by a rectangle of length $L$ and width $2R$ with two half-circle end caps of radius $R$:

$$A = \pi R^2 + 2LR \tag{4.4}$$

We can define the length of this rod as the length of the rectangle plus with the two end caps:

$$l \equiv L + 2R \tag{4.5}$$

We can also find the integral of the distance field, $B = B_{\text{caps}} + B_{\text{rect}}$:

$$B = \int_0^R (R - r)2\pi r dr + \int_0^R (R - r)2L dr \tag{4.6}$$

and evaluate to find:

$$B = \frac{\pi}{3}R^3 + LR^2 \tag{4.7}$$

As we want to solve for the rod radius $R$ and length $l$, we substitute for $L$ in Eqs. 4.4 and 4.7 using Eq. 4.5 to find:

$$A = (\pi - 4)R^2 + 2lR \tag{4.8}$$

$$B = \frac{\pi}{3}R^3 + (l - 2R)R^2 \tag{4.9}$$

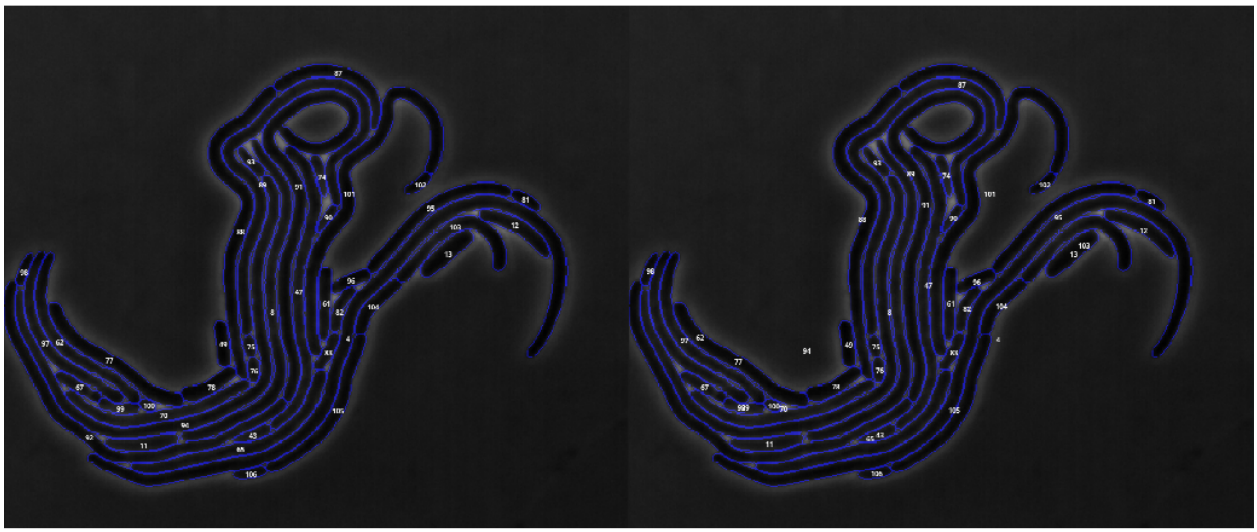


Figure 4.5: **Visualization of Cell IDs via skeleton medoid.** Left: Cell IDs labeled at medoid of skeleton. Right: Cell IDs labeled at centroid. Notice that due to cell curvature, some cell IDs (for example, cell 101) are displayed outside the corresponding cell area.

We first solve for $R$. We combine the equations for $A$ and $B$ to cancel out the terms with $l$ resulting in a cubic equation, which we rearrange to the form of a depressed cubic equation:

$$R^3 - \frac{3A}{\pi}R + \frac{6B}{\pi} = 0 \tag{4.10}$$

We define two constants $p$ and $q$ to solve the depressed cubic equation:

$$p \equiv -\frac{3A}{\pi} \tag{4.11}$$

$$q \equiv \frac{6B}{\pi} \tag{4.12}$$

We then calculate a quantity related to the discriminant:

$$D_1 = \frac{q^2}{4} + \frac{p^3}{27}, \tag{4.13}$$

and using Cardano's formula, calculate the three possible roots for $R$:

$$C_1 = \sqrt[3]{-\frac{q}{2} + \sqrt{D_1}} \tag{4.14}$$

$$C_2, C_3 = C_1 * \frac{-1 \pm \sqrt{-3}}{2} \tag{4.15}$$

The resulting values are then filtered for only real, positive roots, and the smallest root is chosen as the *rod radius*, $r_{\text{rod}}$. The *rod length*, $l_{\text{rod}}$, is then calculated by plugging in the rod radius into Eq. 4.4:

$$l_{\text{rod}} = \frac{A - \pi r_{\text{rod}}^2}{2r_{\text{rod}}} + 2r_{\text{rod}} \tag{4.16}$$

In MATLAB, we calculate $B$ using the discrete sum (rather than the integral) of the distance field of the mask. To account for this discrepancy, we subtract an offset term found

from simulating pre-defined masks and calculating their rod radius and length:

$$B' = B - 0.5A_{\text{mask}} \tag{4.17}$$

$B'$ is used when calculating $q$ in Eq. 4.12.

## *4.3 Room for improvement: Drawbacks and limitations*

While OmniSegger, Omnipose, and Bactrack offer significant improvements to the time-lapse analysis pipeline, in practice, various issues remain. The most pressing issues for OmniSegger at the moment are i) inconsistent calling of divisions and ii) an open question about how omnipotent a cell segmentation-analysis software can be.

### *4.3.1 Pants: Segmentation and tracking remain a paired challenge*

The majority of cell tracking algorithms take masks as inputs, then define frame-to-frame linking costs for each cell mask based on properties such as mask overlap, and minimize the costs to determine links. Determining links and segmentation simultaneously is often computationally intensive, as the cost matrix presents an exponentially growing combinatorics problem; if not determined simultaneously, cell tracking accuracy is totally dependent on cell segmentation results. Thus, if the preceding cell segmentation contains errors, the tracking and lineage determination will be disrupted.

Precisely determining the exact time of division for a cell in an experiment is impossible– timelapses observe the instantaneous event in discrete time steps. Furthermore, the time of division is often ambiguous when only observed by phase-contrast imaging; this was the primary reason for training the Omnipose bacteria phase-contrast model based on underlying membrane or cytosol fluorescence signal when possible. Due to the ambiguous nature of the phase-contrast image, the algorithm will have a 'flickering' effect; for example, the cell may be determined to be divided into two cells in frame $t_0$, but one cell in the subsequent frame $t_0 + 1$, and back to two cells in frame $t_0 + 2$. This flickering effect presents an issue

for 2D segmentation algorithms, especially when the timelapses are taken at high frame rate. Current existing 2D segmentation models only perform segmentation by considering individual frames, while the model determination about cell division must persist across multiple frames.

Omnipose introduces the idea of a 'spacetime' model, where the segmentation algorithm takes temporal information into account, effectively becoming a 3D model (2D+T). Let's consider the kymograph for a cell dividing into two daughter cells: the mother cell grows until the cell wall septates roughly near the center. Conceptually, the kymograph appears like pants; as time increases, the mother is the waist of the pants, which splits into two daughter pants legs. The split is persistent in time. Assuming a frame rate high enough that cells overlap from frame to frame, the 3D spacetime segmentation also presents a solution for cell linking, as each 3D lineage volume contains the time of division and the mother-daughter information. While the 3D spacetime Omnipose model (`bact_phase_spacetime`) is promising, it is lacking the large ground-truth, annotated training dataset as used for the 2D segmentation model and as a result is much less robust.

While persistent in time because of the constraint posed on the Bactrack algorithm, I noticed that the division time determined by Bactrack was not biologically accurate, and therefore chose only to implement its tracking function. In theory, the division time determination might be able to be fixed by adjusting the hierarchical segmentation, as the information should be contained in the flow field.

Though highly robust, Omnipose can have segmentation errors, and its 2D model does not fix the persistent cell division issue. Therefore, running Bactrack on Omnipose masks which flicker between one to two cells leads to lineage errors.

SuperSegger contains some error resolution functions which can edit the underlying masks to solve the division issue, though the methods are not fully robust, as they typically only compare the frames immediately before and after a division error, rather than for all time. In practice, OmniSegger tends to work better than SuperSegger-Bactrack on unusual morphologies.

### *4.3.2 Segmentation and tracking steps are now modular*

Though Omnipose and Bactrack are suggested to be used with OmniSegger, the full implementation of the two packages is not hard-coded into OmniSegger. Instead, their output–masks and a linking matrix, respectively–is the input to SuperSegger. In fact, any other segmentation or linking algorithm can be used, as long as the masks are in `png` format and the linking matrix in `csv` format follows the same style as Bactrack. The modularity of OmniSegger allows it to be compatible with future advances in cell segmentation and tracking.

### *4.3.3 Generalization of cell cytometry software*

While the introduction of modular segmentation and linking steps allows OmniSegger to be much more robust to analyzing timelapses of diverse cell morphologies, the analysis software is still biased towards rod-shaped cells. For example, quantities such as long axis length or cell pole age are calculated. However, consider cocci, which are spherically shaped bacteria. The significance of a long axis or a cell pole measurement becomes unclear for such a morphology. Cytometry calculations must therefore be more generalized, or carefully checked, if analyzing morphologies which are not rod-shaped. In theory, SuperSegger can be used to analyze both prokaryotic and eukaryotic timelapses.

## ***4.4 Results: Improved analysis capabilities of OmniSegger***

The analysis capabilities of OmniSegger are demonstrated using a curated set of test data representing various morpholgies and imaging experiments. None of these datasets have been used as ground-truth training data for the Omnipose `bact_phase_omni` model. I further discuss competing packages that have segmentation and timelapse analysis features.

### *4.4.1 Test datasets and morphologies*

*Rod-shaped cells: Control dataset*

A cropped timelapse of exponentially growing *E. coli* taken with a Nikon Eclipse Ti-E microscope has been tested as a control dataset for rod-shaped cells. A sample image and lineage tree are shown in Fig. 4.9.

*Filamentous cells dataset*

A cropped timelapse of exponentially growing MG1655 *E. coli* taken with a Nikon Eclipse Ti-E microscope has been tested as a dataset representing filamentous cells. The strain has been treated with sub-Minimum Inhibitory Concentration of 10$\mu$M hydroxyurea [103, 104], leading to filamentation. A sample image is shown in Fig. 4.10.

*Fluorescence dataset*

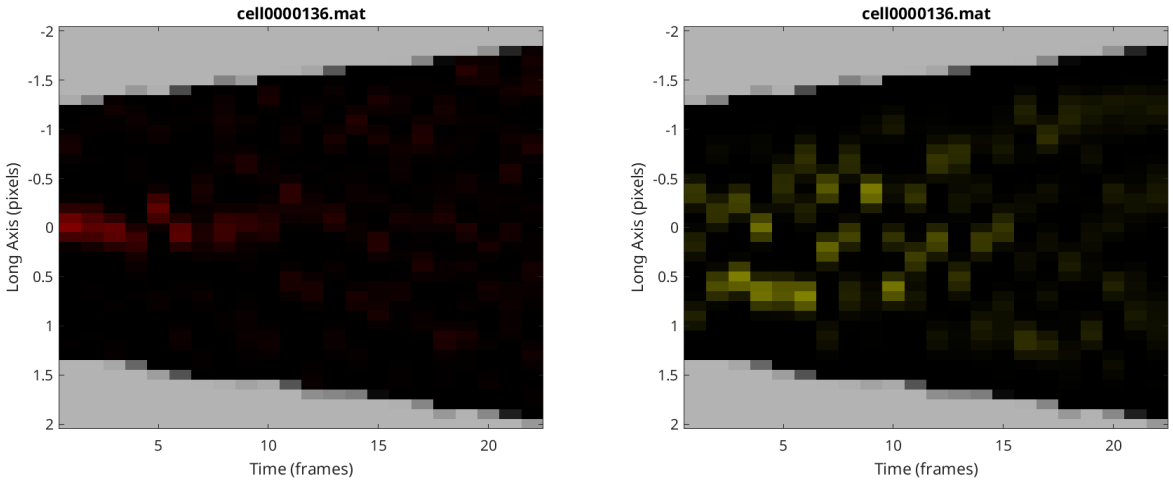
A cropped timelapse of Pal-mCherry *E. coli* taken with a Nikon Eclipse Ti-E microscope has been tested as a sample dataset for phase-contrast and fluorescence analysis. A kymograph mosaic of the fluorescence signal for single cells is shown in Fig. 4.13.

*Punctate foci dataset*

A cropped timelapse of YPet-DnaN *A. baylyi* taken with a Nikon Eclipse Ti-E microscope has been tested as a sample dataset for punctate foci fluorescence analysis. A sample image and fluorescence kymograph are shown in Fig. 4.11.

*Multichannel fluorescence dataset*

A cropped timelapse of mCherry-DnaN YPet-SSB AB1157 *E. coli* taken with a Nikon Eclipse Ti-E microscope in the Wiggins lab has been tested as a sample dataset for multichannel fluorescence analysis. Cell kymographs for each channel are shown in Fig. 4.6.




(a) mCherry-DnaN fluorescence.

(b) YPet-SSB fluorescence.

Figure 4.6: **OmniSegger can analyze multichannel fluorescence timelapses.** Two kymographs from a single cell (ID: 136) in a timelapse of *E. coli* strain AB1157 with YPet-SSB and mCherry-DnaN fluorescent fusions, demonstrating the analysis capabilities for multichannel timelapse movies.

*Large frame-of-view dataset*

A snapshot of *FtsZ-GFP E. coli* taken with a Nikon Eclipse Ti-E microscope has been tested as a sample dataset for a large frame of view analysis. A sample image is shown in Fig. 4.7.

*Brightfield dataset*

A cropped timelapse of *Pal-mCherry E. coli* taken with a Nikon Eclipse Ti-E microscope has been tested as a sample dataset for brightfield timelapse analysis. A sample image is shown in Fig. 4.16.

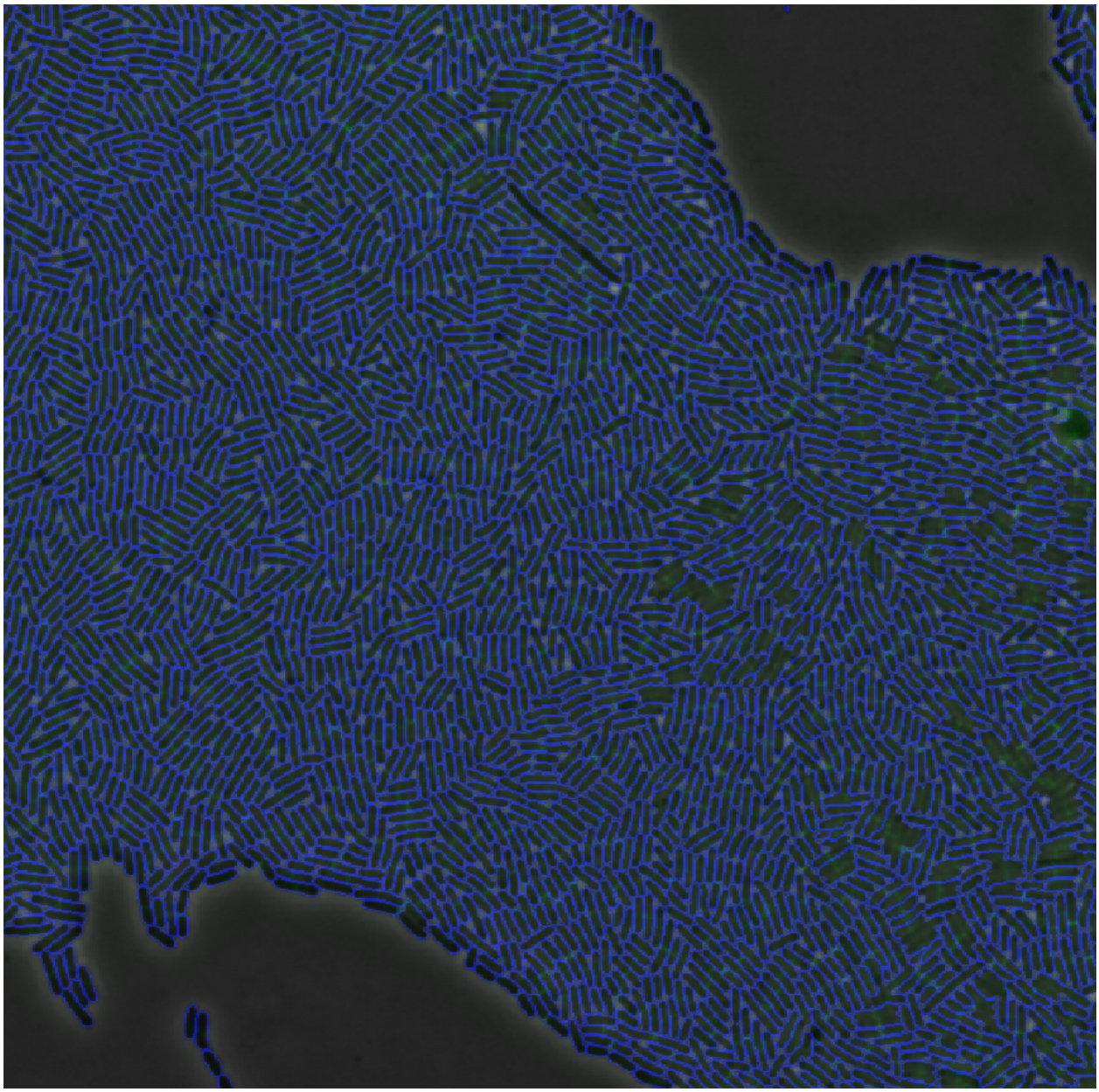
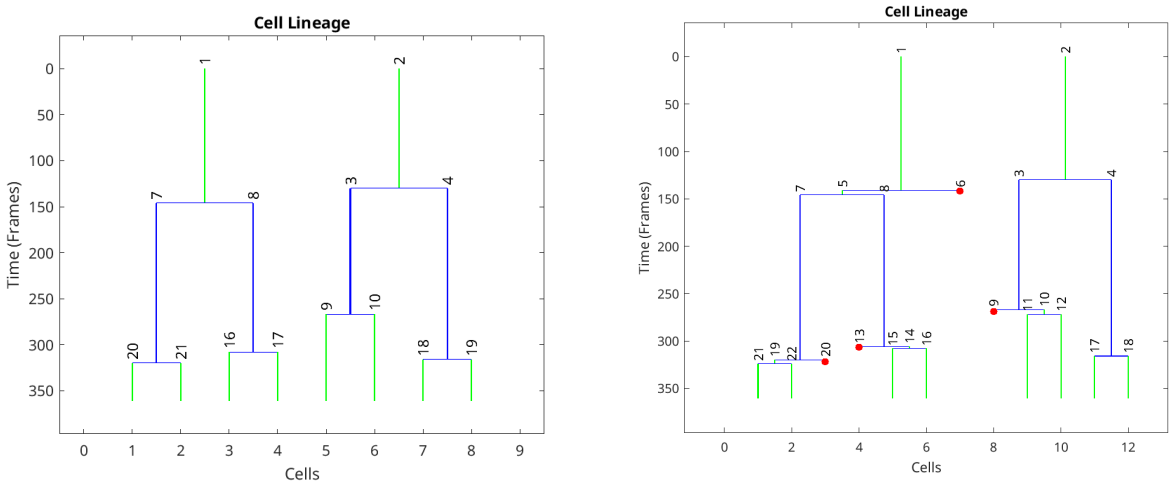


Figure 4.7: **Segmentation on high density, large frame-of-view snapshot of FtsZ-GFP *E. coli*.** The image has partially been zoomed in to show segmentation performance.




(a) Lineage from Omnipose.

(b) Lineage from Bactrack.

Figure 4.8: **OmniSegger correction determines consistent cell division in time.** Two lineage trees generated by OmniSegger from the same timelapse of *E. coli* strain MG1655 with a frame rate of $10s^{-}1$. Left: Omnipose segmentation struggles due to inconsistent calling of division during segmentation, however, OmniSegger segmentation corrects the masks to resolve linking errors. Right: Bactrack improperly segments the cells and the error correction code in OmniSegger is turned off, resulting in several linking errors.

*High time-resolution dataset*

A cropped timelapse of MG1655 *E. coli* taken with a Nikon Eclipse Ti-E microscope has been tested as a sample dataset for brightfield timelapse analysis. The frame rate for the dataset is $10^{-1}$s and results in ambiguous determination of cell division from phase-contrast imaging. Two resulting lineage trees are shown, one as determined by OmniSegger with Omnipose segmentation (subject to inconsistent division calling in time, but able to be fixed by OmniSegger) and another as determined by OmniSegger with Bactrack segmentation (error correction by OmniSegger turned off). See Figure 4.8.

### *4.4.2 Competing segmentation and analysis packages*

The adoption of machine learning algorithms have greatly improved the accuracy of cell segmentation. It is important to note that many packages focus solely on segmentation rather than cytometry, and thus the competing packages mentioned only include segmentation and analysis suites. Furthermore, some packages allow segmentation plugins, but I only discuss the out-of-the-box features.

Many built-in segmentation algorithms of analysis packages are threshold-based and therefore have segmentation results sensitive to imaging conditions or morphology [105, 106, 107]. Other packages which measure cell statistics, including lineages and mean fluorescence levels, have worse segmentation performance compared to Omnipose as demonstrated by comparing Intersection over Union metrics [108, 109, 110]. Furthermore, other competing packages emphasize segmentation less than single-cell cytometry [111, 112, 113]. A comparison of the segmentation results from competing pipeline analysis packages are shown in Figures 4.10 and 4.9. Even worse, a couple of software programs require user-input training or parameters on a dataset-by-dataset basis: in practice, these software programs require hours spent unnecessarily calibrating segmentation results that may be less accurate compared to algorithms such as Omnipose. In comparison, OmniSegger allows for compelling analysis and visualization capabilities with minimal user input or coding experience required. I provide a comparison of features and functions for the most updated timelapse analysis packages in Table 4.1, based on my own user experience and available documentation.

## *4.5 Results: Analysis of unusual cell morphology in A. baylyi*

OmniSegger was used to analyze essential-gene knockouts in *A. baylyi*[9]. Many essential-gene knockouts result in extreme morphologies [117], and datasets were previously not able to be quantified for several years at the single-cell level due to poor segmentation performance of existing packages [95]. The development of Omnipose and its integration with SuperSegger enables widespread quantitative analysis of transformed knockout mutants.

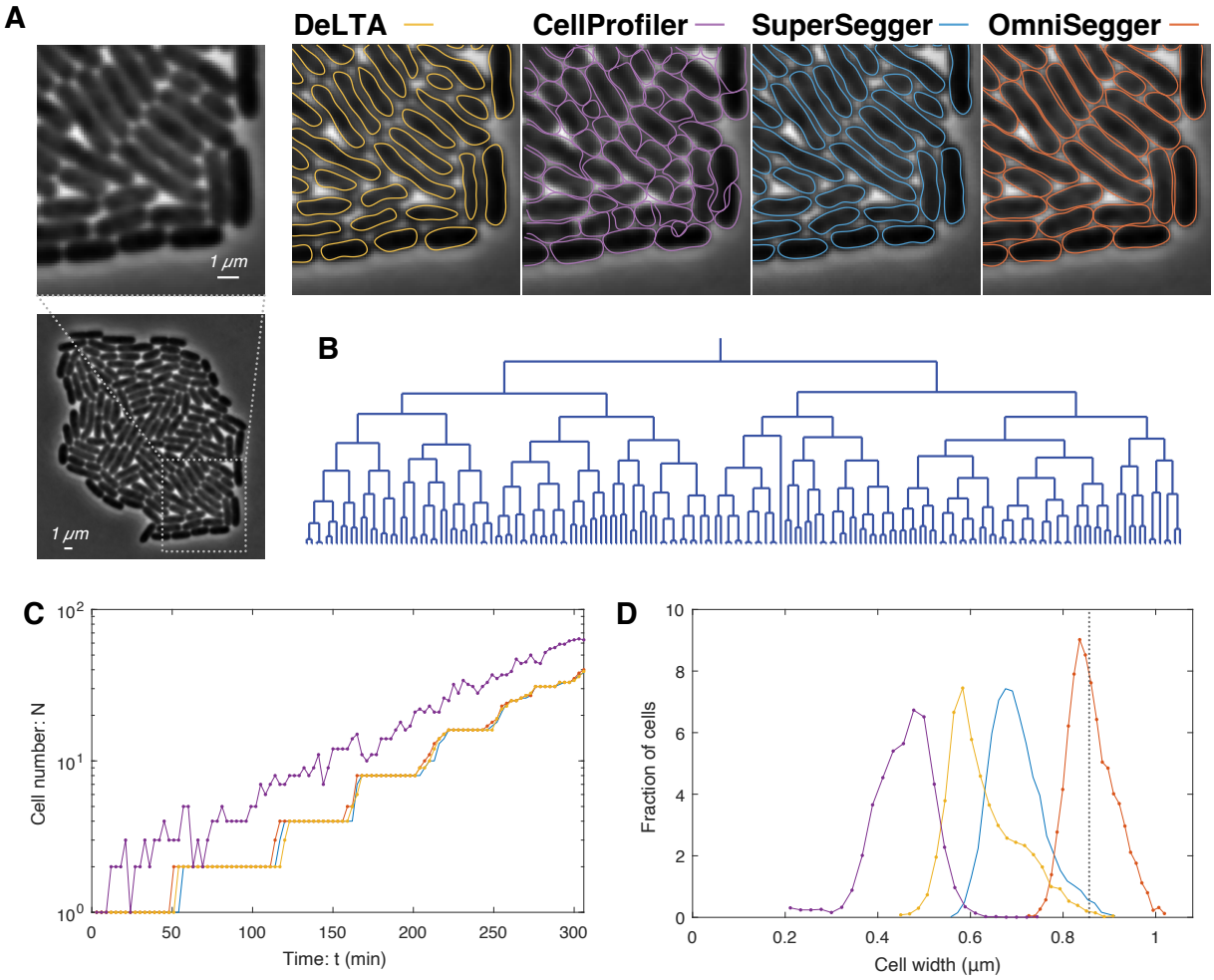



Figure 4.9: **Comparing OmniSegger performance for typical/rod-shaped morphologies: *Wild-type E. coli.*** Updated cell boundaries allow improved timelapse analysis of cell cytometry. **Panel A: Segmentation comparison.** Phase-contrast image of a wild-type *E. coli* colony represents one frame in a timelapse. Cell boundaries are determined by segmentation methods of DeLTA (yellow), Ilastik-CellProfiler (purple), SuperSegger (blue), and OmniSegger (orange). **Panel B: Lineage tree. Panel C: Cell number comparison.** OmniSegger, SuperSegger, and DeLTA have comparable performance in terms of cell number count over time. **Panel D: Cell width comparison.** Histograms for cell widths from all frames of the timelapse are shown. Black dotted line indicates manually calculated estimate of average width. OmniSegger measures the most consistent and accurate cell width in comparison to alternative packages.

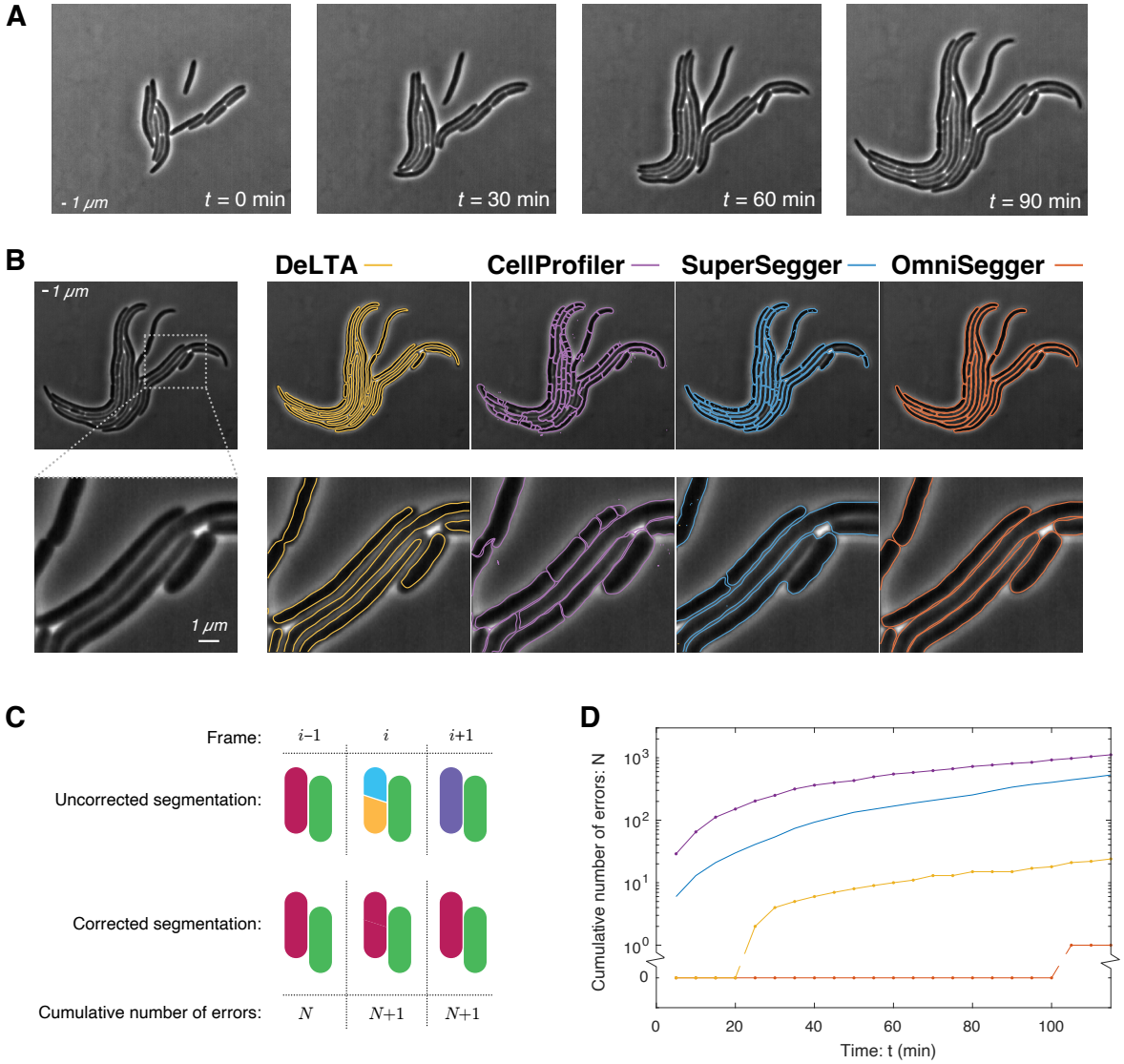



Figure 4.10: **Comparing OmniSegger performance for diverse morphologies: *Filamented E. coli.* Panel A: timelapse mosaic.** Sample frames taken from a timelapse of a growing wild-type *E. coli* colony treated with sub-Minimum Inhibitory Concentration (MIC) of 10$\mu$M hydroxyurea. Hydroxyurea inhibits DNA synthesis and results in a phenotype of cell filamentation when below the MIC. **Panel B: Segmentation comparison.** Cell boundaries are determined by segmentation methods of DeLTA (yellow), Ilastik-CellProfiler (purple), SuperSegger (blue), and OmniSegger (orange); segmentation of competing packages result in oversegmentation, undersegmentation, or less accurate pixel classification. **Panel C: Error counting schematic.** Example of segmentation error: the left-hand cell is oversegmented in frame $i$, resulting in erroneous assignment of cell ID and improper linking in subsequent frame $i+1$. **Panel D: Performance comparison.** OmniSegger has the fewest segmentation errors per frame of the timelapse, and therefore requires the least manual corrections for data analysis.

| **Software** | No User-defined Training/Inputs | Segmentation Method | Tracking Method | Visualization GUI | Output Files | Language | OS Support | Quantitation of Non-diffuse Foci | Modality | Automated Plot Generation | Year Updated |
|---|---|---|---|---|---|---|---|---|---|---|---|
| OmniSegger | ✓ | Deep learning (Omnipose) | Traditional | ✓ | mat, xls | MATLAB & Python | Linux, Windows, MacOS | ✓ | Phase Pad, Brightfield, Cytoplasmic, Membrane | ✓ | 2024 |
| SuperSegger [90] | ✓ | ML-informed Threshold | Traditional | ✓ | mat | MATLAB | Linux, Windows, MacOS | ✓ | Phase Pad | ✓ | 2018 |
| DeLTA [109] | ✓ | Deep learning | Deep learning | X | nc | Python | Linux, Windows | X | Phase Pad, Phase MM | X | 2024 |
| Ilastik-CellProfiler [106, 113] | X | ML-informed Threshold or Watershed | Traditional | X | xls | Standalone | Linux, Windows, MacOS | X | All: with training | ✓ | 2024 |
| FAST [110] | X | Threshold | Unsupervised learning | ✓ | mat | MATLAB, Standalone | Linux, Windows, MacOS | X | Phase Pad, Brightfield | ✓ | 2023 |
| CellShape [114] | X | Threshold | N/A | ✓ | N/A | Python | Linux, Windows, MacOS | ✓ | Phase Pad | ✓ | 2017 |
| Oufti [115] | X | Threshold | Traditional | ✓ | mat, out, csv | MATLAB | Linux, Windows | ✓ | Phase Pad | ✓ | 2016 |
| MicrobeJ [116] | ✓ | Threshold | Traditional | ✓ | res, csv | Java (ImageJ) | Linux, Windows, MacOS | ✓ | Phase Pad | ✓ | 2024 |

Table 4.1: A comparison of features and functions for cellular imaging analysis software packages.

Omnipose provided significant improvement to the accuracy and performance of cell segmentation and its `bact_phase_omni` model was trained on Δ*dnaA*, Δ*ftsN*, and Δ*murA A. baylyi* mutants, but was *not* trained on Δ*dnaN* mutants. The phenotype of the Δ*dnaN* mutant is cell filamentation, similar but distinct from the filamentation of the Δ*ftsN* mutant. As a result, the segmentation results for Δ*dnaN* in particular required low-level manual correction of the masks, which was done in Napari [118].

The Omnipose masks were input to OmniSegger, which performed the single-cell measurements, tracking, plotting and data visualization. A representative frame mosaic, fluorescence over time, and lineage are shown in Fig. 4.11. In the DnaN knockout experiment, we use a strain with a fluorescent fusion of essential replisome gene *dnaN* at the endogenous locus to indicate the level of DnaN being expressed and also to observe the localization of the replisome. In Panel A, the blue-outlined cell at $t = 0$ min is transformed by having *dnaN* knocked out, while gaining kanamycin antibiotic resistance. The wild-type cells are outlined in green and the untransformed cells outlined in red; these do not have kanamycin resistance and therefore are not expected to proliferate. As the timelapse runs, we notice that selective

media ($Km^+$), wild-type and un-transformed cells are arrested while the transformed cells continue to proliferate, as seen in the lineage tree (Panel D). Furthermore, since the transformed cells are no longer able to produce DnaN protein, all daughter cells inherit protein from the original progenitor. In fact, we do see the depletion of DnaN from cell-proliferation-induced dilution as the fluorescence intensity decreases exponentially over time (Panel E). The observation that the progenitor cell carries more protein than strictly needed supports the hypothesis of overabundance: in fact, the cell is able to proliferate for several generations after having the essential gene knocked out.

### *4.6 Results: New cell bias using FtsZ-GFP and Pal-mCherry E. coli*

At any given time, an exponentially growing population in steady-state should have more young cells than old cells, and this effect will be referred to as *new cell bias*, where young cells are defined as cells with age less than half their doubling time T and old cells are defined as cells with age greater than doubling time T. To provide intuition for the new cell bias, every old cell enriches the population with two new cells at division. We expect this ratio to be $1/\sqrt{2}$ due to the enrichment of newborn cells in the population [65].

In order to test for this effect experimentally, snapshot images and timelapses sampling an exponentially growing population of fluorescently-labeled cells were analyzed. FtsZ is a protein which assembles into a ring (the Z-ring) during cell division. Using a GFP fusion to FtsZ, a feature of the timing of cell division decided by Omnipose is that the Z-ring appears as a bright focus at the cell pole during the beginning of the cell cycle (see Fig. 4.12). This feature is used as a marker to identify young cells in the image. Pal is a highly expressed outer membrane protein in *E. coli*. We notice that the Pal signal increases greatly at mid-cell during cell division (see Fig. 4.13), and use the feature as a marker to identify old cells in the image.

After processing with SuperSegger, the cell files containing the fluorescence image of each cell is analyzed using custom code to identify the ratio of young cells to total cells in the FtsZ-GFP images, and the ratio of old cells to total cells in the Pal-mCherry images.

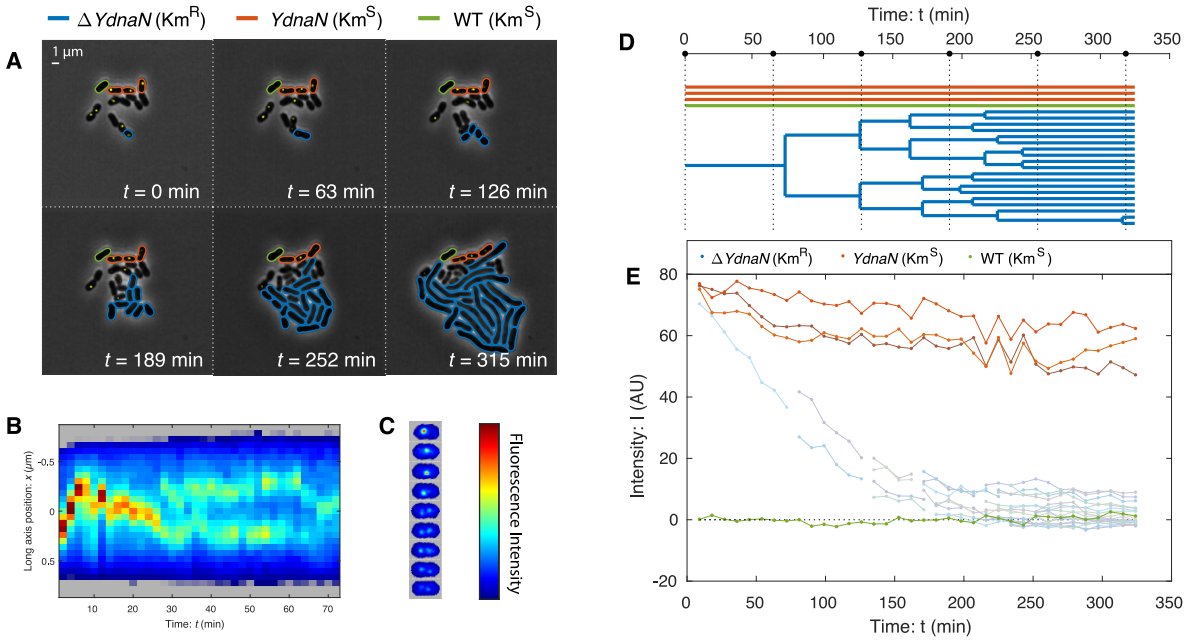



Figure 4.11: **OmniSegger pipeline visualization gallery.** The pipeline includes numerous visualization tools. **Panel A: Composite image frame mosaic.** The package can create multi-channel timelapse mosaics, including vectorized cell outlines. **Panel B & C: Fluorescence kymographs and cell towers.** Visualization of fluorescence localization over time for a single cell, generated by OmniSegger. **Panel D: Lineage trees** Temporal representation for mother-daughter relations of *A. baylyi* mutants Δ *YdnaN* (blue), *YdnaN* (red), and wild-type (green) cells. **Panel E: Cell cytometry plots.** OmniSegger can generate plots to show the dynamics of various cell characteristics, for example, total fluorescence intensity over time.

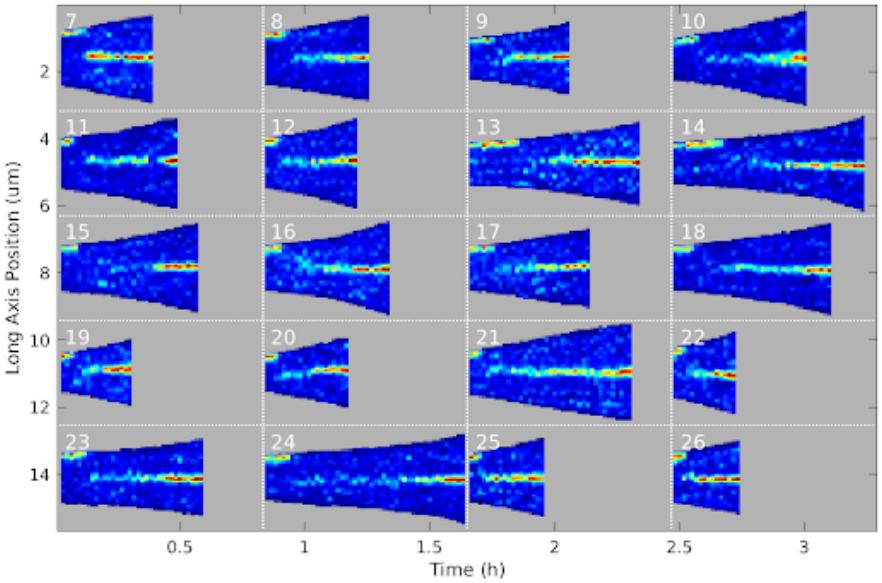



Figure 4.12: **Omnipose finds FtsZ to be localized to the pole at birth.** Cell birth and division times are decided by Omnipose image segmentation. A new cell is identified as having a bright focus at the pole, possibly due to FtsZ remaining from the Z-ring, or as an artifact of Omnipose determining cell division too early. In the analysis, a new cell is defined when the peak intensity at the pole is greater than three times the average intensity from the rest of the cell.

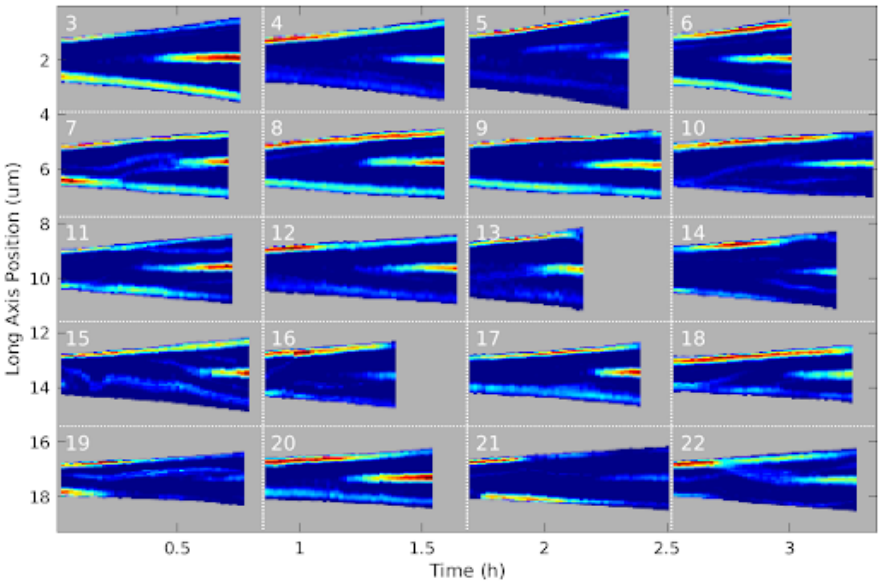



Figure 4.13: **Omnipose finds mid-cell Pal intensity to be greatest at division.** Cell birth and division times are decided by Omnipose image segmentation. In the analysis, an old cell is defined when the peak intensity at mid-cell is greater than twice the average intensity from the rest of the cell.

Next, we determine four ratios: from the FtsZ-GFP datasets, the ratio of young cells in the snapshot image ($N_{\mathrm{y,SS}}$) and in the timelapse ($N_{\mathrm{y,TL}}$), and from the Pal-mCherry datasets, the ratio of old cells in the snapshot image ($N_{\mathrm{o,SS}}$), and in the timelapse ($N_{\mathrm{o,TL}}$). In total, we use over 250,000 cells in this analysis.

These ratios make sense because the timelapse samples a synchronous culture, while the snapshot samples an asynchronous culture. We account for the total number of cells of each sample by dividing the asynchronous value by the synchronous value, which represents the number of cells that have undergone a full cell cycle.

We divide the four ratios to find the ratio of young cells to old cells:

$$\frac{\frac{N_{\mathrm{y,SS}}}{N_{\mathrm{y,TL}}}}{\frac{N_{\mathrm{o,SS}}}{N_{\mathrm{o,TL}}}} = \frac{\frac{98404}{57998}}{\frac{48526}{49552}} = 0.69 \pm 0.01, \tag{4.18}$$

which approximately returns our theoretically expected ratio of $1/\sqrt{2} \approx 0.071$.

## *4.7 Results: Brightfield cell segmentation*

Brightfield microscopy is an imaging modality where white light is transmitted through the sample, and sample features are distinguished through change in contrast due to the absorbance of the transmitted light. A brightfield microscope only requires a condenser lens, objective lens, light source, and camera and thus offers the most simple setup for microscopy. In particular, the setup of a brightfield microscope offers an advantage over a phase-contrast microscope when also imaging with fluorescence microscopy; the phase-contrast microscope requires the addition of a phase shift ring and a neutral density ring, which partially blocks the path of the transmitted fluorescence light and thus attenuates the fluorescent signal [119].

Furthermore, brightfield microscopy is widely used in cell biology labs [120, 121], and thus introducing a brightfield segmentation algorithm facilitates new experiments by removing the barrier of requiring a lab to change their microscope setup or purchase new equipment.

In the spirit of generalizing the applicability of SuperSegger for cell imaging analysis, I trained a model for Omnipose which can segment brightfield images of bacterial cells, and can be used with OmniSegger.

### *4.7.1 The challenge posed by brightfield segmentation*

Though the setup of brightfield microscopy is simple, the specific imaging details are complicated. For bacterial samples, images of bacteria taken in the focal plane yield very low sample contrast [122] due to the low light absorbance of the samples. The contrast is so low that it becomes difficult for the human eye to distinguish cells from background (see Fig. 4.14). However, images taken slightly above or below the focal plane increase contrast at the expense of resolution. In fact, it is common to see images taken slightly out-of-focus in the literature [123, 8], and out-of-focus focal planes have been used to improve segmentation results as well [121, 122, 124].

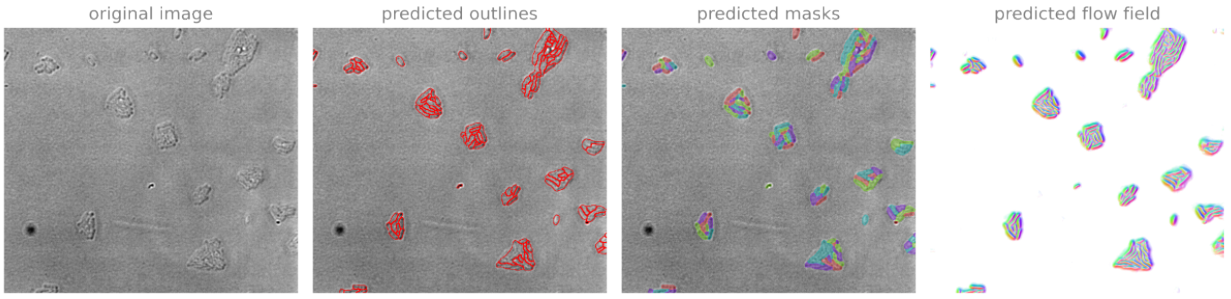



Figure 4.14: **Brightfield model performance on in-focus image.** Brightfield image of *E. coli* in the focal plane. The low contrast of in-focus brightfield makes the cells difficult to distinguish, even with the human eye.

The fact that there is no agreed upon metric defining a slightly over- or under-focused focal plane means that the images taken will not capture cell boundaries consistently [125], with differences ranging from some researchers taking over-focused images differently, to other researchers taking under-focused images instead. We could imagine defining some offset values from the focal plane for more consistency, but this may also vary depending on experimental setup and sample thickness. Because cell boundaries are not well-defined, the resulting segmentation is less accurate and reliable than that based on phase-contrast.

Can a segmentation model for in-focus images be trained, and if not, can we train a model which can segment over- and under-focused images? To answer this question, I took z-stacks to capture both over-, under-, and in-focus fields of view for the creation of a ground-truth and testing dataset for a brightfield segmentation model.

### *4.7.2 Impossibility of training a in-focus model*

To train a model for in-focus cells, I identified the in-focus planes of the z-stacks, then labeled and trained only using the in-focus images as ground-truth. The performance of the model was poor upon evaluation on test data of in-focus images. In order to test if the training dataset was too small due to only including one focal plane of each frame-of-view, I increased the size of the training dataset by including planes which where fractions of a

micron above and below the exact in-focus plane; the performance of the second trained model on in-focus planes was also poor.

The failure to train a model for in-focus segmentation confirmed the intuition that a machine-learning model may not be able to outperform the segmentation done by the human eye in its ability to distinguish cells. However, the in-focus plane may contain more image "information" than out-of-focus planes, as the captured image is least blurred. In theory, this information would be able to be quantified and utilized for segmentation purposes by introducing a new predicted variable to Omnipose's network architecture.

### *4.7.3 Training the brightfield Omnipose model*

Upon concluding that a in-focus plane was not feasible, I next changed my focus to training a model to segment under- and over-focused images. I trained two models: one using all focal planes and another using only over- and under-focused planes.

As cells in over-focused images appear similar to phase-contrast, I was able to generate rough masks using the Omnipose phase model, and then I hand-annotated the masks in Napari to refine and correct errors. The over- and under-focused images had a slight offset which I accounted for by performing image registration on the ground-truth masks.

The trained model which excludes the in-focus planes has better performance. It is able to reliably segment both over- and under-focused brightfield images.

Surprisingly, there is limited public availability of brightfield image datasets for bacterial cells. I used existing published datasets [8] and took more data to improve the robustness of the model.

Number of images: 471.

### *4.7.4 Performance of the brightfield model*

The brightfield segmentation model for Omnipose performs reliably on under- and over-focused images. The result on an test image from the DeepBacs dataset [8] that the model

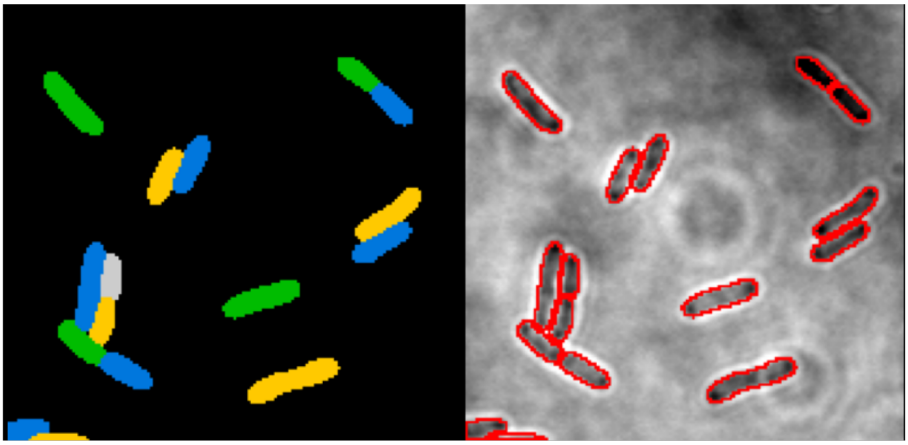


Figure 4.15: **Brightfield model performance on test image.** Performance of the Omnipose brightfield model on an over-focused *E. coli* image from [8]. Left: masks, right: boundaries as determined by masks overlaid on the original image. The segmentation accurately captures cell boundaries.

has not seen before is shown in Fig. 4.15, which I compared to the current brightfield segmentation competitors, MiSiC [123] and DeepBacs [8]. Importantly, ground-truth masks used to train the competing models appear lackluster in their respective publications. Ideally, the model would also be trained on more diverse data–for example, it likely is not able to segment filamented cells.

## *4.8 Conclusion: Analysis of new morphologies and modalities enables new experiments*

The development of OmniSegger enables the quantification of a variety of novel experiments involving mutants, antibiotic-treated cells, and multi-species interactions, at the single cell level. Many of these experiments involve observing extreme cell morphologies. Due to the challenge of accurate and robust cell segmentation, for example with essential gene knockout mutants, these analyses were previously not possible. Furthermore, the close integration of Omnipose and introduction of new trained models allows OmniSegger to handle multiple image modalities (see Fig. 4.16). The development of Omnipose enabled the segmentation step

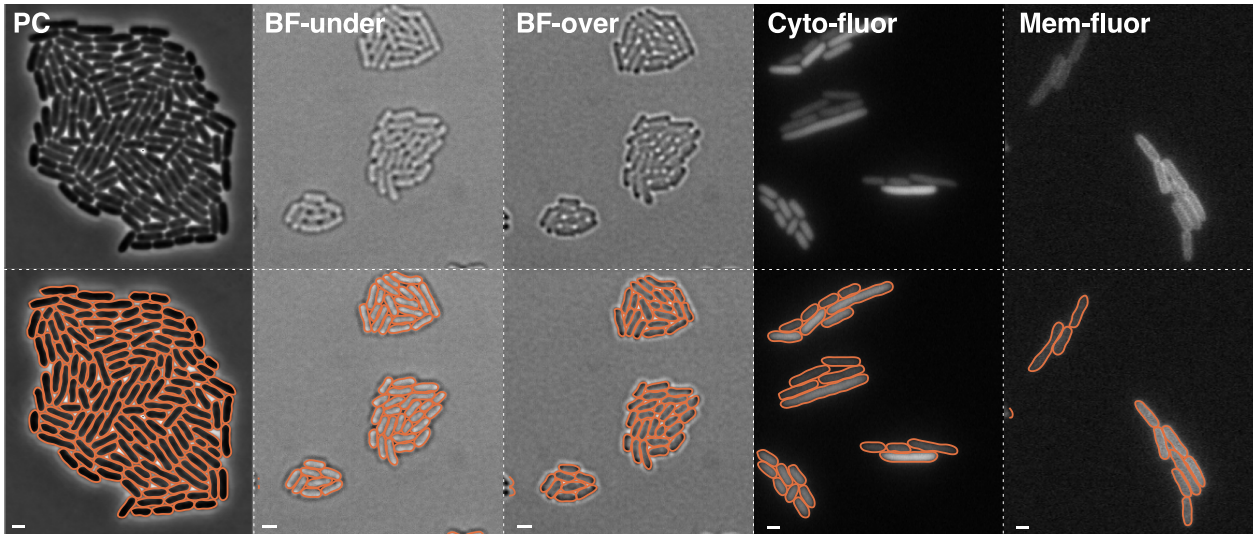



Figure 4.16: **Performance in multiple imaging modalities.** From left to right: phase-contrast image of wild-type MG1655 *E. coli* captured on a Nikon Eclipse Ti-E microscope; underfocused brightfield image of *Pal-mCherry ZipA-sfGFP E. coli* captured on a custom lab microsope; overfocused brightfield image of *Pal-mCherry ZipA-sfGFP E. coli* captured on a custom lab microsope; cytoplasmic fluorescence image of *E. coli* strain JW3984 from the ASKA collection with fusion lysC-GFP captured on a Nikon Eclipse Ti-E microscope; membrane fluorescence image of *E. coli* strain JW1466 from the ASKA collection with fusion cycA-GFP captured on a Nikon Eclipse Ti-E microscope. Segmentation outlines generated by OmniSegger. Scale bar: 1$\mu$m.

of the image analysis pipeline, and its implementation in OmniSegger introduces a powerful, highly automated tool for the biological imaging community, with minimal coding experience needed.

# Chapter 5

# FUTURE OUTLOOKS

## *5.1 Noise & Overabundance*

With the new understanding of the impact of gene expression noise on central dogma processes, we should continue to test the predictions made by the RLTO model. In particular, some interesting experiments would be to look for more experimental evidence of overabundance. We have genome-wide evidence from transformation transposon insertion mutant sequencing (TFNseq) [49] in *A. baylyi*, however, single-cell imaging-based experiments have only been performed for four essential genes (*dnaN* - replication, *ftsN* - cell division/septation, *murA* - cell wall synthesis, *dnaA* - replication) [9]. Notably, the measured overabundance from the genomic-based approach differs from what is measured from imaging experiments. With analysis enabled by OmniSegger, we can now consider knocking out essential genes involved in other processes, in addition to examining the knockout phenotype.

The overabundance hypothesis also predicts different rates of translation and transcription based on the abundance of essential proteins; in particular, the translation efficiency should increase with increased transcription in eukaryotes. A possible experiment would be to examine the strengths of promoters and ribosome binding sites for genes which are predicted to be sufficient as opposed to overabundant.

Speaking of sufficiency, we observe several (31%) essential genes in *A. baylyi* that are sufficient rather than overabundant (69%) (see Fig. 5.1). For example, among these genes are *nrdA* and *nrdB*, which makeup subunits of ribonucleotide reductase, involved in DNA synthesis. The sufficiency strategy suggests that these genes may be subject to tight regulation,

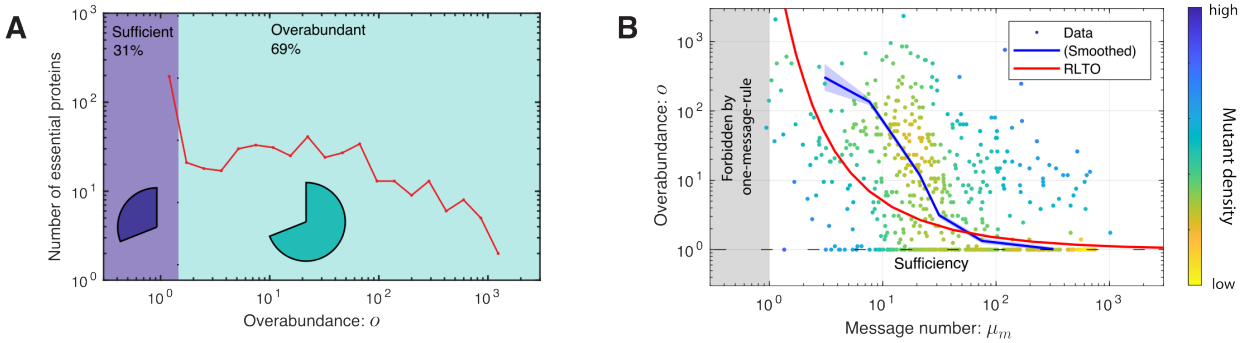



Figure 5.1: **Panel A: Overabundance varies by orders of magnitude between essential proteins.** The protein overabundance is inferred from the arrest time in TFNseq. Sufficient expression genes have overabundance $o = 1$, while overabundant genes vary from $o > 1$ to very large overabundance ($o > 100$). **Panel B: Overabundance is large for low-expression essential proteins.** The measured message-number-overabundance pairs are shown for essential genes (including estimated gene density.) The smoothed experimental data is shown in blue (with experimental uncertainty.) The RLTO model (red) predicts that overabundance grows rapidly as the transcription level is reduced. The RLTO model qualitatively captures the trend of the data (blue); however, it appears to underestimate the measured overabundance for intermediate expression genes. Figure adapted from [9].

such as auto-regulation. Therefore, looking for evidence of a regulatory process for these sufficient proteins would add a missing puzzle piece to the story.

In terms of application, the strategy of overabundance predicts that many essential proteins need to be significantly depleted before having any effect on growth (i.e., overabundant proteins are buffered against depletion). As a consequence, the essential proteins which follow the sufficiency strategy may be easily targeted by inhibitors (*e.g.* antibiotics). Searching for a correlation between widely used antibiotics and the overabundance of their protein targets may give rise to insights for the rationale of why certain antibiotics are effective in combating bacterial infections. Such an analysis may inform the development of more targeted and efficient antibiotics.

## *5.2 OmniSegger*

As with any software package, there are an infinite number of small fixes and tweaks possible to improve the software, user experience, and documentation. However, in this section, I discuss more ambitious pursuits which will allow OmniSegger to be much more powerful.

### *5.2.1 Synthetic cell images*

The Omnipose bacterial phase-contrast model, spacetime model, and brightfield model all present significant innovations for cell segmentation. However, the spacetime and brightfield models are currently limited in robustness due to a lack of training data.

In addition to collecting a large amount of micrographs of diverse cells with various imaging conditions, a significant bottleneck to training the models is the time and labor required for manual annotation of ground-truth masks. The performance of image segmentation models is highly dependent on the quality and quantity of training data; inaccuracies in the ground-truth training data can lead to artifacts when predicting masks from images. Therefore, thousands of hours have gone into carefully inspecting and fixing each ground-truth mask by hand. However, human annotations can also introduce bias to the labeling of the data [126, 127].

Rather than generating masks from images, a clever application of machine-learning is to generate synthetic images of cells from masks, which are realistic enough to improve model performance without human annotation [128]. Furthermore, a style transfer could be applied to the synthetic image in order to generate training data for other imaging modalities (*i.e.*, phase-contrast to brightfield). Alternatively, the physical properties of cells and microscope optics can be used to simulate cell imaging in phase-contrast as done with SyMBac [129]. The development of methods to create synthetic images for training data will: i) drastically reduce the time and labor needed to annotate ground-truth data, ii) generate annotations

without human bias, and iii) enable models to be robust to different biological conditions and imaging modalities.

### *5.2.2 Implementation in MATLAB vs Python*

The code for the original SuperSegger was written in 2016 in MATLAB; many of its competitors at the time were based in MATLAB or ImageJ [130]. However, in 2024, Python has become the most popular coding language, with a wide library of open-source packages. Furthermore, Python is free, powerful enough to process matrices quickly, and used widely in the machine learning community [131], while MATLAB and its toolboxes require a license that largely limits its usage to the academic sphere. The majority of current cell analysis softwares are now written in Python [108], with a growing number of libraries as well.

Porting SuperSegger 2 to Python is the next logical step for future use and availability to a wider audience. In addition, because Omnipose and Bactrack are both written in Python, the implementation and installation process would be greatly simplified. There are also many options for data visualization and graphical interface, while MATLAB mainly provides its limited App Designer interface.

As a counterargument, I note that the compatibility issues for Python packages can become very difficult to solve and manage, and there are no obligations for developers to maintain their packages; MATLAB is continuously updated and maintained as a software product. These compatibility issues will present a hindrance both during the rewrite process and once completed, again in the maintanance process. However, I believe the translation of SuperSegger 2 to Python is very necessary to be widely-used and competitive.

# BIBLIOGRAPHY

[1] John R S Newman, Sina Ghaemmaghami, Jan Ihmels, David K Breslow, Matthew Noble, Joseph L DeRisi, and Jonathan S Weissman. Single-cell proteomic analysis of *S. cerevisiae* reveals the architecture of biological noise. *Nature*, 441(7095):840–6, Jun 2006.

[2] Yuichi Taniguchi, Paul J Choi, Gene-Wei Li, Huiyi Chen, Mohan Babu, Jeremy Hearn, Andrew Emili, and X Sunney Xie. Quantifying *E. coli* proteome and transcriptome with single-molecule sensitivity in single cells. *Science*, 329(5991):533–8, Jul 2010.

[3] Lyris M F de Godoy, Jesper V Olsen, Jürgen Cox, Michael L Nielsen, Nina C Hubner, Florian Fröhlich, Tobias C Walther, and Matthias Mann. Comprehensive mass-spectrometry-based proteome quantification of haploid versus diploid yeast. *Nature*, 455(7217):1251–4, Oct 2008.

[4] Sina Ghaemmaghami, Won-Ki Huh, Kiowa Bower, Russell W Howson, Archana Belle, Noah Dephoure, Erin K O'Shea, and Jonathan S Weissman. Global analysis of protein expression in yeast. *Nature*, 425(6959):737–41, Oct 2003.

[5] Moshe Kafri, Eyal Metzl-Raz, Ghil Jona, and Naama Barkai. The cost of protein production. *Cell Rep*, 14(1):22–31, Jan 2016.

[6] Eyal Metzl-Raz, Moshe Kafri, Gilad Yaakov, and Naama Barkai. Gene transcription as a limiting factor in protein production and cell growth. *G3 (Bethesda)*, 10(9):3229–3242, Sep 2020.

[7] Jinha Jung, Edoardo Pasolli, Saurabh Prasad, James C. Tilton, and Melba M. Crawford. A framework for land cover classification using discrete return LiDAR data: Adopting pseudo-waveform and hierarchical segmentation. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 7(2):491–502, February 2014.

[8] Christoph Spahn, Estibaliz Gómez-de Mariscal, Romain F. Laine, Pedro M. Pereira, Lucas von Chamier, Mia Conduit, Mariana G. Pinho, Guillaume Jacquemet, Séamus Holden, Mike Heilemann, and Ricardo Henriques. DeepBacs for multi-task bacterial image analysis using open-source deep learning approaches. *Communications Biology*, 5(1), July 2022.

[9] Han Kyou Choi, Teresa W. Lo, Kevin J. Cutler, Dean Huang, William Ryan Will, and Paul A Wiggins. Protein overabundance is driven by growth robustness, August 2024.

[10] Teresa W. Lo, Han Kyou James Choi, Dean Huang, and Paul A. Wiggins. The one-message-per-cell-cycle rule: A conserved minimum transcription level for essential genes. 2023.

[11] Jonathan M Raser and Erin K O'Shea. Noise in gene expression: origins, consequences, and control. *Science*, 309(5743):2010–3, Sep 2005.

[12] R. Phillips, J. Kondev, J. Theriot, and N. Orme. *Physical Biology of the Cell*. Garland Science, 2013.

[13] J Paulsson and M Ehrenberg. Random signal fluctuations can reduce random fluctuations in regulated components of chemical regulatory networks. *Phys Rev Lett*, 84(23):5447–50, Jun 2000.

[14] Johan Paulsson. Summing up the noise in gene networks. *Nature*, 427(6973):415–8, Jan 2004.

[15] Jean Hausser, Avi Mayo, Leeat Keren, and Uri Alon. Central dogma rates and the trade-off between precision and economy in gene expression. *Nat Commun*, 10(1):68, Jan 2019.

[16] Arjun Raj, Charles S Peskin, Daniel Tranchina, Diana Y Vargas, and Sanjay Tyagi. Stochastic mRNA synthesis in mammalian cells. *PLoS Biol*, 4(10):e309, Oct 2006.

[17] Srividya Iyer-Biswas, F Hayot, and C Jayaprakash. Stochasticity of gene products from transcriptional pulsing. *Phys Rev E Stat Nonlin Soft Matter Phys*, 79(3 Pt 1):031911, Mar 2009.

[18] Peccoud J and Ycart B. Markovian modeling of gene-product synthesis. *Theor. Popul. Biol.*, 48:222–234, 1995.

[19] Nir Friedman, Long Cai, and X Sunney Xie. Linking stochastic dynamics to population distribution: an analytical framework of gene expression. *Phys Rev Lett*, 97(16):168302, Oct 2006.

[20] Arren Bar-Even, Johan Paulsson, Narendra Maheshri, Miri Carmi, Erin O'Shea, Yitzhak Pilpel, and Naama Barkai. Noise in protein expression scales with natural protein abundance. *Nat Genet*, 38(6):636–43, Jun 2006.

[21] David E Weinberg, Premal Shah, Stephen W Eichhorn, Jeffrey A Hussmann, Joshua B Plotkin, and David P Bartel. Improved ribosome-footprint and mRNA measurements provide insights into dynamics and regulation of yeast translation. *Cell Rep*, 14(7):1787–1799, Feb 2016.

[22] Michael B Elowitz, Arnold J Levine, Eric D Siggia, and Peter S Swain. Stochastic gene expression in a single cell. *Science*, 297(5584):1183–6, Aug 2002.

[23] Peter S Swain, Michael B Elowitz, and Eric D Siggia. Intrinsic and extrinsic contributions to stochasticity in gene expression. *Proc Natl Acad Sci U S A*, 99(20):12795–800, Oct 2002.

[24] F Crick. Central dogma of molecular biology. *Nature*, 227(5258):561–3, Aug 1970.

[25] J L Hargrove and F H Schmidt. The role of mRNA and protein stability in gene expression. *FASEB J*, 3(12):2360–70, Oct 1989.

[26] A L Koch and H R Levy. Protein turnover in growing cultures of *Escherichia coli*. *J Biol Chem*, 217(2):947–57, Dec 1955.

[27] Miguel Martin-Perez and Judit Villén. Determinants and regulation of protein turnover in yeast. *Cell Syst*, 5(3):283–294.e5, Sep 2017.

[28] DT. Gillespie. A rigorous derivation of the chemical master equation. *Physica A*, 188(1):404–425, 1992.

[29] Daniel T. Gillespie. Exact stochastic simulation of coupled chemical reactions. *The Journal of Physical Chemistry*, 81(25):2340–2361, 1977.

[30] Jonathan A Bernstein, Arkady B Khodursky, Pei-Hsun Lin, Sue Lin-Chao, and Stanley N Cohen. Global analysis of mRNA decay and abundance in *Escherichia coli* at single-gene resolution using two-color fluorescent DNA microarrays. *Proc Natl Acad Sci U S A*, 99(15):9697–702, Jul 2002.

[31] Huiyi Chen, Katsuyuki Shiroguchi, Hao Ge, and Xiaoliang Sunney Xie. Genome-wide study of mRNA degradation and transcript elongation in *Escherichia coli*. *Mol Syst Biol*, 11(5):808, May 2015.

[32] Premal Shah, Yang Ding, Malwina Niemczyk, Grzegorz Kudla, and Joshua B Plotkin. Rate-limiting steps in yeast protein translation. *Cell*, 153(7):1589–601, Jun 2013.

[33] Rohan Balakrishnan, Matteo Mori, Igor Segota, Zhongge Zhang, Ruedi Aebersold, Christina Ludwig, and Terence Hwa. Principles of gene regulation quantitatively connect DNA to RNA and proteins in bacteria. *Science*, 378(6624):eabk2066, Dec 2022.

[34] Matteo Mori, Zhongge Zhang, Amir Banaei-Esfahani, Jean-Benoît Lalanne, Hiroyuki Okano, Ben C Collins, Alexander Schmidt, Olga T Schubert, Deok-Sun Lee, Gene-Wei Li, Ruedi Aebersold, Terence Hwa, and Christina Ludwig. From coarse to fine: the absolute *Escherichia coli* proteome under diverse growth conditions. *Mol Syst Biol*, 17(5):e9536, May 2021.

[35] Tomoya Baba, Hsuan-Cheng Huan, Kirill Datsenko, Barry L Wanner, and Hirotada Mori. The applications of systematic in-frame, single-gene knockout mutant collection of *Escherichia coli* K-12. *Methods Mol Biol*, 416:183–94, 2008.

[36] S Y Gerdes, M D Scholle, J W Campbell, G Balázsi, E Ravasz, M D Daugherty, A L Somera, N C Kyrpides, I Anderson, M S Gelfand, A Bhattacharya, V Kapatral, M D'Souza, M V Baev, Y Grechkin, F Mseeh, M Y Fonstein, R Overbeek, A-L Barabási, Z N Oltvai, and A L Osterman. Experimental determination and system level analysis of essential genes in *Escherichia coli* MG1655. *J Bacteriol*, 185(19):5673–84, Oct 2003.

[37] Emily C A Goodall, Ashley Robinson, Iain G Johnston, Sara Jabbari, Keith A Turner, Adam F Cunningham, Peter A Lund, Jeffrey A Cole, and Ian R Henderson. The essential genome of *Escherichia coli* K-12. *mBio*, 9(1), Feb 2018.

[38] Preeti Mehta, Sherwood Casjens, and Sankaran Krishnaswamy. Analysis of the lambdoid prophage element e14 in the *E. coli* K-12 genome. *BMC Microbiol*, 4:4, Jan 2004.

[39] A Deloupy, V Sauveplane, J Robert, S Aymerich, M Jules, and L Robert. Extrinsic noise prevents the independent tuning of gene expression noise and protein mean abundance in bacteria. *Sci Adv*, 6(41), Oct 2020.

[40] Brian L Chin, Owen Ryan, Fran Lewitter, Charles Boone, and Gerald R Fink. Genetic variation in *Saccharomyces cerevisiae*: circuit diversification in a signal transduction network. *Genetics*, 192(4):1523–32, Dec 2012.

[41] A Ciechanover. Intracellular protein degradation: from a vague idea thru the lysosome and the ubiquitin-proteasome system and onto human diseases and drug targeting. *Cell Death Differ*, 12(9):1178–90, Sep 2005.

[42] Eran Eden, Naama Geva-Zatorsky, Irina Issaeva, Ariel Cohen, Erez Dekel, Tamar Danon, Lydia Cohen, Avi Mayo, and Uri Alon. Proteome half-life dynamics in living human cells. *Science*, 331(6018):764–8, Feb 2011.

[43] Ido Golding, Johan Paulsson, Scott M Zawilski, and Edward C Cox. Real-time kinetics of gene activity in individual bacteria. *Cell*, 123(6):1025–36, Dec 2005.

[44] Geoffrey M. Cooper. *The Cell: A Molecular Approach. 2nd edition*. Sinauer Associates 2000, 2000.

[45] Teresa W. Lo, H. James Choi, Dean Huang, and Paul A. Wiggins. Noise robustness and metabolic load determine the principles of central dogma regulation. *Science Advances*, 10(34), August 2024.

[46] Erez Dekel and Uri Alon. Optimality and evolutionary tuning of the expression level of a protein. *Nature*, 436(7050):588–92, Jul 2005.

[47] Leeat Keren, Jean Hausser, Maya Lotan-Pompan, Ilya Vainberg Slutskin, Hadas Alisar, Sivan Kaminski, Adina Weinberger, Uri Alon, Ron Milo, and Eran Segal. Massively parallel interrogation of the effects of gene expression levels on fitness. *Cell*, 166(5):1282–1294.e18, Aug 2016.

[48] Jason M Peters, Alexandre Colavin, Handuo Shi, Tomasz L Czarny, Matthew H Larson, Spencer Wong, John S Hawkins, Candy H S Lu, Byoung-Mo Koo, Elizabeth Marta, Anthony L Shiver, Evan H Whitehead, Jonathan S Weissman, Eric D Brown, Lei S Qi, Kerwyn Casey Huang, and Carol A Gross. A comprehensive, CRISPR-based functional analysis of essential genes in bacteria. *Cell*, 165(6):1493–1506, Jun 2016.

[49] Larry A Gallagher, Jeannie Bailey, and Colin Manoil. Ranking essential bacterial processes by speed of mutant death. *Proc Natl Acad Sci U S A*, 117(30):18010–18017, 07 2020.

[50] Hans G. Schlegel Joseph W. Lengeler, Gerhart Drews, editor. *Biology of the Prokaryotes*. Georg Thieme Verlag, Rüdigerstrasse 14, D-70469 Stuttgart, Germany, 1998.

[51] Nathan M Belliveau, Griffin Chure, Christina L Hueschen, Hernan G Garcia, Jane Kondev, Daniel S Fisher, Julie A Theriot, and Rob Phillips. Fundamental limits on the rate of bacterial growth and their influence on proteomic composition. *Cell Syst*, 12(9):924–944.e2, 09 2021.

[52] Matthew Scott, Carl W Gunderson, Eduard M Mateescu, Zhongge Zhang, and Terence Hwa. Interdependence of cell growth and gene expression: origins and consequences. *Science*, 330(6007):1099–102, Nov 2010.

[53] J. I. Steinfeld, J. S. Francisco, and W. L. Hase. *Chemical Kinetics and Dynamics*. Prentice-Hall, 2nd edition, 1999.

[54] Daniel A Charlebois, Nezar Abdennur, and Mads Kaern. Gene expression noise facilitates adaptation and drug resistance independently of mutation. *Phys. Rev. Lett.*, 107(21):218101, November 2011.

[55] Melanie R Silvis, Manohary Rajendram, Handuo Shi, Hendrik Osadnik, Andrew N Gray, Spencer Cesar, Jason M Peters, Cameron C Hearne, Parth Kumar, Horia Todor, Kerwyn Casey Huang, and Carol A Gross. Morphological and transcriptional responses to CRISPRi knockdown of essential genes in *Escherichia coli*. *mBio*, 12(5):e0256121, Oct 2021.

[56] Stefano Donati, Michelle Kuntz, Vanessa Pahl, Niklas Farke, Dominik Beuter, Timo Glatter, José Vicente Gomes-Filho, Lennart Randau, Chun-Ying Wang, and Hannes Link. Multi-omics analysis of CRISPRi-knockdowns identifies mechanisms that buffer decreases of enzymes in *E. coli* metabolism. *Cell Syst*, 12(1):56–67.e6, Jan 2021.

[57] Jolanda van Leeuwen, Carles Pons, Guihong Tan, Jason Zi Wang, Jing Hou, Jochen Weile, Marinella Gebbia, Wendy Liang, Ermira Shuteriqi, Zhijian Li, Maykel Lopes, Matej Ušaj, Andreia Dos Santos Lopes, Natascha van Lieshout, Chad L Myers, Frederick P Roth, Patrick Aloy, Brenda J Andrews, and Charles Boone. Systematic analysis of bypass suppression of essential genes. *Mol Syst Biol*, 16(9):e9828, Sep 2020.

[58] Tim Wang, Kıvanç Birsoy, Nicholas W Hughes, Kevin M Krupczak, Yorick Post, Jenny J Wei, Eric S Lander, and David M Sabatini. Identification and characterization of essential genes in the human genome. *Science*, 350(6264):1096–101, Nov 2015.

[59] Björn Schwanhäusser, Dorothea Busse, Na Li, Gunnar Dittmar, Johannes Schuchhardt, Jana Wolf, Wei Chen, and Matthias Selbach. Global quantification of mammalian gene expression control. *Nature*, 473(7347):337–42, May 2011.

[60] Hugo Dourado, Matteo Mori, Terence Hwa, and Martin J Lercher. On the optimality of the enzyme-substrate relationship in bacteria. *PLoS Biol.*, 19(10):e3001416, October 2021.

[61] Matteo Mori, Severin Schink, David W Erickson, Ulrich Gerland, and Terence Hwa. Quantifying the benefit of a proteome reserve in fluctuating environments. *Nat Commun*, 8(1):1225, 10 2017.

[62] Guillaume Lambert and Edo Kussell. Memory and fitness optimization of bacteria under fluctuating environments. *PLoS Genet*, 10(9):e1004556, Sep 2014.

[63] J Monod, A M Pappenheimer, Jr, and G Cohen-Bazire. The kinetics of the biosynthesis of beta-galactosidase in *Escherichia coli* as a function of growth. *Biochim Biophys Acta*, 9(6):648–60, Dec 1952.

[64] Barbara Bosch, Michael A DeJesus, Nicholas C Poulton, Wenzhu Zhang, Curtis A Engelhart, Anisha Zaveri, Sophie Lavalette, Nadine Ruecker, Carolina Trujillo, Joshua B Wallach, Shuqi Li, Sabine Ehrt, Brian T Chait, Dirk Schnappinger, and Jeremy M Rock. Genome-wide gene expression tuning reveals diverse vulnerabilities of *M. tuberculosis*. *Cell*, 184(17):4579–4592.e24, Aug 2021.

[65] Dean Huang, Teresa Lo, Houra Merrikh, and Paul A Wiggins. Characterizing stochastic cell-cycle dynamics in exponential growth. *Phys Rev E*, 105(1-1):014420, Jan 2022.

[66] Moshe Kafri, Eyal Metzl-Raz, Felix Jonas, and Naama Barkai. Rethinking cell growth models. *FEMS Yeast Res*, 16(7), Nov 2016.

[67] Ethan Levien, Jiseon Min, Jane Kondev, and Ariel Amir. Non-genetic variability in microbial populations: survival strategy or nuisance? *Reports on Progress in Physics*, 84(11):116601, nov 2021.

[68] E. O. Powell. Growth rate and generation time of bacteria, with special reference to continuous culture. *Microbiology*, 15(3):492–511, 1956.

[69] George Arfken. *Mathematical Methods for Physicists*. Academic Press, Inc., San Diego, third edition, 1985.

[70] Alexander Bartholomäus, Ivan Fedyunin, Peter Feist, Celine Sin, Gong Zhang, Angelo Valleriani, and Zoya Ignatova. Bacteria differently regulate mRNA abundance to specifically respond to various stresses. *Philos Trans A Math Phys Eng Sci*, 374(2063), Mar 2016.

[71] Ron Milo. What is the total number of protein molecules per cell volume? A call to rethink some published values. *Bioessays*, 35(12):1050–5, Dec 2013.

[72] B. Alberts, D. Bray, J. Lewis, M. Raff, K. Roberts, and J.D. Watson. *Molecular Biology of the Cell*. Garland, 4th edition, 2002.

[73] L L Chia and C McLaughlin. The half-life of mRNA in *Saccharomyces cerevisiae*. *Mol Gen Genet*, 170(2):137–44, Feb 1979.

[74] Vicent Pelechano, Sebastián Chávez, and José E Pérez-Ortín. A complete set of nascent transcription rates for yeast genes. *PLoS One*, 5(11):e15442, Nov 2010.

[75] B Futcher, G I Latter, P Monardo, C S McLaughlin, and J I Garrels. A sampling of the yeast proteome. *Mol Cell Biol*, 19(11):7357–68, Nov 1999.

[76] Edward Yang, Erik van Nimwegen, Mihaela Zavolan, Nikolaus Rajewsky, Mark Schroeder, Marcelo Magnasco, and James E Darnell, Jr. Decay rates of human mRNAs: correlation with functional characteristics and sequence attributes. *Genome Res*, 13(8):1863–72, Aug 2003.

[77] Ron Milo, Paul Jorgensen, Uri Moran, Griffin Weber, and Michael Springer. Bionumbers–the database of key numbers in molecular and cell biology. *Nucleic Acids Res*, 38(Database issue):D750–3, Jan 2010.

[78] William R Blevins, Teresa Tavella, Simone G Moro, Bernat Blasco-Moreno, Adrià Closa-Mosquera, Juana Díez, Lucas B Carey, and M Mar Albà. Extensive post-transcriptional buffering of gene expression in the response to severe oxidative stress in baker's yeast. *Sci Rep*, 9(1):11005, Jul 2019.

[79] L M Hereford and M Rosbash. Number and distribution of polyadenylated RNA sequences in yeast. *Cell*, 10(3):453–62, Mar 1977.

[80] Tobias von der Haar. A quantitative estimation of the global translational activity in logarithmically growing yeast cells. *BMC Syst Biol*, 2:87, Oct 2008.

[81] Daniel Zenklusen, Daniel R Larson, and Robert H Singer. Single-RNA counting reveals alternative modes of gene expression in yeast. *Nat Struct Mol Biol*, 15(12):1263–71, Dec 2008.

[82] Fumihito Miura, Noriko Kawaguchi, Mikio Yoshida, Chihiro Uematsu, Keiji Kito, Yoshiyuki Sakaki, and Takashi Ito. Absolute quantification of the budding yeast transcriptome by means of competitive PCR between genomic and complementary DNAs. *BMC Genomics*, 9:574, Nov 2008.

[83] Mathias Uhlén, Linn Fagerberg, Björn M Hallström, Cecilia Lindskog, Per Oksvold, Adil Mardinoglu, Åsa Sivertsson, Caroline Kampf, Evelina Sjöstedt, Anna Asplund, IngMarie Olsson, Karolina Edlund, Emma Lundberg, Sanjay Navani, Cristina Al-Khalili Szigyarto, Jacob Odeberg, Dijana Djureinovic, Jenny Ottosson Takanen, Sophia Hober, Tove Alm, Per-Henrik Edqvist, Holger Berling, Hanna Tegel, Jan Mulder, Johan Rockberg, Peter Nilsson, Jochen M Schwenk, Marica Hamsten, Kalle von Feilitzen, Mattias Forsberg, Lukas Persson, Fredric Johansson, Martin Zwahlen, Gunnar von Heijne, Jens Nielsen, and Fredrik Pontén. Proteomics. Tissue-based map of the human proteome. *Science*, 347(6220):1260419, Jan 2015.

[84] V E Velculescu, S L Madden, L Zhang, A E Lash, J Yu, C Rago, A Lal, C J Wang, G A Beaudry, K M Ciriello, B P Cook, M R Dufault, A T Ferguson, Y Gao, T C He, H Hermeking, S K Hiraldo, P M Hwang, M A Lopez, H F Luderer, B Mathews, J M Petroziello, K Polyak, L Zawel, and K W Kinzler. Analysis of human transcriptomes. *Nat Genet*, 23(4):387–8, Dec 1999.

[85] S Y Gerdes, M D Scholle, J W Campbell, G Balázsi, E Ravasz, M D Daugherty, A L Somera, N C Kyrpides, I Anderson, M S Gelfand, A Bhattacharya, V Kapatral, M D'Souza, M V Baev, Y Grechkin, F Mseeh, M Y Fonstein, R Overbeek, A-L Barabási, Z N Oltvai, and A L Osterman. Experimental determination and system level analysis of essential genes in *Escherichia coli* MG1655. *J Bacteriol*, 185(19):5673–84, Oct 2003.

[86] C R Rao. Information and the accuracy attainable in the estimation of statistical parameters. *Bulletin of Calcutta Mathematical Society*, 37:81–91., 1945.

[87] H Cramer. *Mathematical methods of statistics.* Princeton University Press., 1946.

[88] Petri-Jaan Lahtvee, Benjamín J Sánchez, Agata Smialowska, Sergo Kasvandik, Ibrahim E Elsemman, Francesco Gatto, and Jens Nielsen. Absolute quantification of protein and mRNA abundances demonstrate variability in gene-specific translation efficiency in yeast. *Cell Syst*, 4(5):495–504.e5, May 2017.

[89] Kristoffer Herland Hellton and Magne Thoresen. The impact of measurement error on principal component analysis. *Scandinavian Journal of Statistics*, 41(4):1051–1063, April 2014.

[90] Stella Stylianidou, Connor Brennan, Silas B Nissen, Nathan J Kuwada, and Paul A Wiggins. SuperSegger: robust image segmentation, analysis and lineage tracking of bacterial cells. *Mol Microbiol*, 102(4):690–700, 11 2016.

[91] Nathan J. Kuwada, Beth Traxler, and Paul A. Wiggins. Genome-scale quantitative characterization of bacterial protein localization dynamics throughout the cell cycle. *Molecular Microbiology*, 95(1):64–79, November 2014.

[92] Nathan J Kuwada, Beth Traxler, and Paul A Wiggins. High-throughput cell-cycle imaging opens new doors for discovery. *Curr Genet*, 61(4):513–6, Nov 2015.

[93] Sarah M Mangiameli, Christopher N Merrikh, Paul A Wiggins, and Houra Merrikh. Transcription leads to pervasive replisome instability in bacteria. *eLife*, 6, January 2017.

[94] Manuel Guizar-Sicairos, Samuel T. Thurman, and James R. Fienup. Efficient subpixel image registration algorithms. *Optics Letters*, 33(2):156, January 2008.

[95] Kevin J. Cutler, Carsen Stringer, Teresa W. Lo, Luca Rappez, Nicholas Stroustrup, S. Brook Peterson, Paul A. Wiggins, and Joseph D. Mougous. Omnipose: a high-precision morphology-independent solution for bacterial cell segmentation. *Nature Methods*, 19(11):1438–1448, October 2022.

[96] Sherry Yang. Bactrack GitHub, 2024.

[97] Jordão Bragantini, Merlin Lange, and Loïc Royer. Large-scale multi-hypotheses cell tracking using ultrametric contours maps, 2023.

[98] Q. Huangfu and J. A. J. Hall. Parallelizing the dual revised simplex method. *Mathematical Programming Computation*, 10(1):119–142, December 2017.

[99] R. Lougee-Heimer. The common optimization interface for operations research: Promoting open-source software in the operations research community. *IBM Journal of Research and Development*, 47(1):57–66, January 2003.

[100] Gurobi Optimization, LLC. Gurobi Optimizer Reference Manual, 2024.

[101] Haroldo G. Santos and Túlio A.M. Toffolo. Mixed Integer Linear Programming with Python, 2024.

[102] Hans Mittelmann. The MIPLIB2017 Benchmark Instances and Robustness Spotcheck, 2024.

[103] Bryan W. Davies, Michael A. Kohanski, Lyle A. Simmons, Jonathan A. Winkler, James J. Collins, and Graham C. Walker. Hydroxyurea induces hydroxyl radical-mediated cell death in escherichia coli. *Molecular Cell*, 36(5):845–860, December 2009.

[104] Samvel A. Nazaretyan, Neda Savic, Michael Sadek, Brandy J. Hackert, Justin Courcelle, and Charmain T. Courcelle. Replication rapidly recovers and continues in the presence of hydroxyurea in escherichia coli. *Journal of Bacteriology*, 200(6), March 2018.

[105] Zhenzhou Wang. Cell segmentation for image cytometry: Advances, insufficiencies, and challenges. *Cytometry Part A*, 95(7):708–711, December 2018.

[106] David R. Stirling, Madison J. Swain-Bowden, Alice M. Lucas, Anne E. Carpenter, Beth A. Cimini, and Allen Goodman. CellProfiler 4: improvements in speed, utility and usability. *BMC Bioinformatics*, 22(1), September 2021.

[107] Fabrice de Chaumont, Stéphane Dallongeville, Nicolas Chenouard, Nicolas Hervé, Sorin Pop, Thomas Provoost, Vannary Meas-Yedid, Praveen Pankajakshan, Timothée Lecomte, Yoann Le Montagner, Thibault Lagache, Alexandre Dufour, and Jean-Christophe Olivo-Marin. Icy: an open bioimage informatics platform for extended reproducible research. *Nature Methods*, 9(7):690–696, June 2012.

[108] Atiyeh Ahmadi, Matthew Courtney, Carolyn Ren, and Brian Ingalls. A benchmarked comparison of software packages for time-lapse image processing of monolayer bacterial population dynamics. *Microbiology Spectrum*, 12(8), August 2024.

[109] Owen M. O'Connor, Razan N. Alnahhas, Jean-Baptiste Lugagne, and Mary J. Dunlop. DeLTA 2.0: A deep learning pipeline for quantifying single-cell spatial and temporal dynamics. *PLOS Computational Biology*, 18(1):e1009797, January 2022.

[110] Oliver J. Meacock and William M. Durham. Tracking bacteria at high density with FAST, the Feature-Assisted Segmenter/Tracker. *PLOS Computational Biology*, 19(10):e1011524, October 2023.

[111] Hannah Jeckel and Knut Drescher. Advances and opportunities in image analysis of bacterial cells and communities. *FEMS Microbiology Reviews*, 45(4), November 2020.

[112] David A. Van Valen, Takamasa Kudo, Keara M. Lane, Derek N. Macklin, Nicolas T. Quach, Mialy M. DeFelice, Inbal Maayan, Yu Tanouchi, Euan A. Ashley, and Markus W. Covert. Deep learning automates the quantitative analysis of individual cells in live-cell imaging experiments. *PLOS Computational Biology*, 12(11):e1005177, November 2016.

[113] Stuart Berg, Dominik Kutra, Thorben Kroeger, Christoph N. Straehle, Bernhard X. Kausler, Carsten Haubold, Martin Schiegg, Janez Ales, Thorsten Beier, Markus Rudy, Kemal Eren, Jaime I Cervantes, Buote Xu, Fynn Beuttenmueller, Adrian Wolny,

Chong Zhang, Ullrich Koethe, Fred A. Hamprecht, and Anna Kreshuk. ilastik: interactive machine learning for (bio)image analysis. *Nature Methods*, 16(12):1226–1232, September 2019.

[114] Ángel Goñi-Moreno, Juhyun Kim, and Víctor de Lorenzo. CellShape: A user-friendly image analysis tool for quantitative visualization of bacterial cell factories inside. *Biotechnology Journal*, 12(2), September 2016.

[115] Ahmad Paintdakhi, Bradley Parry, Manuel Campos, Irnov Irnov, Johan Elf, Ivan Surovtsev, and Christine Jacobs-Wagner. Oufti: an integrated software package for high-accuracy, high-throughput quantitative microscopy analysis. *Mol Microbiol*, 99(4):767–77, Feb 2016.

[116] Adrien Ducret, Ellen M Quardokus, and Yves V Brun. MicrobeJ, a tool for high throughput bacterial cell detection and quantitative analysis. *Nat Microbiol*, 1(7):16077, 06 2016.

[117] Jeannie Bailey, Julie Cass, Joe Gasper, Ngoc-Diep Ngo, Paul Wiggins, and Colin Manoil. Essential gene deletions producing gigantic bacteria. *PLOS Genetics*, 15(6):e1008195, June 2019.

[118] Nicholas Sofroniew, Talley Lambert, Grzegorz Bokota, Juan Nunez-Iglesias, Peter Sobolewski, Andrew Sweet, Lorenzo Gaifas, Kira Evans, Alister Burt, Draga Doncila Pop, Kevin Yamauchi, Melissa Weber Mendonça, Genevieve Buckley, Wouter-Michiel Vierdag, Loic Royer, Ahmet Can Solak, Kyle I. S. Harrington, Jannis Ahlers, Daniel Althviz Moré, Oren Amsalem, Ashley Anderson, Andrew Annex, Peter Boone, Jordão Bragantini, Matthias Bussonnier, Clément Caporal, Jan Eglinger, Andreas Eisenbarth, Jeremy Freeman, Christoph Gohlke, Kabilar Gunalan, Hagai Har-Gil, Mark Harfouche, Volker Hilsenstein, Katherine Hutchings, Jessy Lauer, Gregor Lichtner, Ziyang Liu, Lucy Liu, Alan Lowe, Luca Marconato, Sean Martin, Abigail McGovern, Lukasz Migas, Nadalyn Miller, Hector Muñoz, Jan-Hendrik Müller, Christopher Nauroth-Kreß, David Palecek, Constantin Pape, Eric Perlman, Kim Pevey, Gonzalo Peña-Castellanos, Andrea Pierré, David Pinto, Jaime Rodríguez-Guerra, David Ross, Craig T. Russell, James Ryan, Gabriel Selzer, MB Smith, Paul Smith, Konstantin Sofiiuk, Johannes Soltwedel, David Stansby, Jules Vanaret, Pam Wadhwa, Martin Weigert, Jonas Windhager, and Philip Winston. napari: a multi-dimensional image viewer for Python, 2024.

[119] M. Tscherepanow, F. Zöllner, M. Hillebrand, and F. Kummert. *Automatic Segmentation of Unstained Living Cells in Bright-Field Microscope Images*, page 158–172. Springer Berlin Heidelberg, 2008.

[120] Nina Parker, Mark Schneegurt, and Anh-Hue Thi Tu. *Microbiology by OpenStax*. Open Stax Textbooks, February 2023.

[121] Rehan Ali, Mark Gooding, Tünde Szilágyi, Borivoj Vojnovic, Martin Christlieb, and Michael Brady. Automatic segmentation of adherent biological cell boundaries and nuclei from brightfield microscopy images. *Machine Vision and Applications*, 23(4):607–621, May 2011.

[122] Felix Buggenthin, Carsten Marr, Michael Schwarzfischer, Philipp S Hoppe, Oliver Hilsenbeck, Timm Schroeder, and Fabian J Theis. An automatic method for robust and fast cell detection in bright field images from high-throughput microscopy. *BMC Bioinformatics*, 14(1), October 2013.

[123] Swapnesh Panigrahi, Dorothée Murat, Antoine Le Gall, Eugénie Martineau, Kelly Goldlust, Jean-Bernard Fiche, Sara Rombouts, Marcelo Nöllmann, Leon Espinosa, and Tâm Mignot. Misic, a general deep learning-based method for the high-throughput cell segmentation of complex bacterial communities. *eLife*, 10, September 2021.

[124] Claire L. Curl, Catherine J. Bellair, Trudi Harris, Brendan E. Allman, Peter J. Harris, Alastair G. Stewart, Ann Roberts, Keith A. Nugent, and Lea M. D. Delbridge. Refractive index measurement in viable cells using quantitative phase-amplitude microscopy and confocal microscopy. *Cytometry Part A*, 65A(1):88–92, March 2005.

[125] Xi Long, W. Louis Cleveland, and Y. Lawrence Yao. Effective automatic recognition of cultured cells in bright field images using fisher's linear discriminant preprocessing. *Image and Vision Computing*, 23(13):1203–1213, November 2005.

[126] Ryan Thiermann, Michael Sandler, Gursharan Ahir, John T Sauls, Jeremy Schroeder, Steven Brown, Guillaume Le Treut, Fangwei Si, Dongyang Li, Jue D Wang, and Suckjoon Jun. Tools and methods for high-throughput single-cell imaging with the mother machine. *eLife*, 12, April 2024.

[127] R. Stuart Geiger, Dominique Cope, Jamie Ip, Marsha Lotosh, Aayush Shah, Jenny Weng, and Rebekah Tang. "garbage in, garbage out" revisited: What do machine learning application papers report about human-labeled training data? *Quantitative Science Studies*, 2(3):795–827, 2021.

[128] Vincent Hickl, Abid Khan, René M. Rossi, Bruno F. B. Silva, and Katharina Maniura-Weber. Segmentation of dense and multi-species bacterial colonies using models trained on synthetic microscopy images, 2024.

[129] Georgeos Hardo, Maximilian Noka, and Somenath Bakshi. Synthetic Micrographs of Bacteria (SyMBac) allows accurate segmentation of bacterial cells using deep neural networks. *BMC Biology*, 20(1), November 2022.

[130] Renske van Raaphorst, Morten Kjos, and Jan-Willem Veening. BactMAP: An R package for integrating, analyzing and visualizing bacterial microscopy data. *Molecular Microbiology*, 113(1):297–308, November 2019.

[131] TIOBE Index, 2024.

# Appendix A

# RUNNING CONDA ENVIRONMENTS IN MATLAB

I include this appendix chapter for those of the niche interest in running Conda environments in MATLAB, which was surprisingly nontrivial. The code described here was used to automatically run the Omnipose and Bactrack Python packages directly in MATLAB, without manual input from the user.

The general idea is to add the relevant directories of the conda environment to the MATLAB Path, and the key code was adapted from Julian Hapke's answer from the MATLAB Answers forum.

## *A.1 Windows*

You can simply add the paths to the conda environment and corresponding scripts to the MATLAB Path, then cleanly remove them from the path once the conda environment is no longer needed.

## *A.2 Linux and MacOS*

First, the path to the conda installation must be added to the MATLAB Path. Next, the conda installation should be initialized in the MATLAB Command Window. Finally, the conda environment must be activated each time it is needed through `source activate environment_name` before scripts are run.

# Appendix B

# THE SUPER FLAT PAD TECHNIQUE

A quick protocol of how to make a super flat pad for imaging. The lab used to use VALP rubber gaskets and filled them with agarose and growth media solution; however, Kevin and I developed this method resulting in a more uniform pad: the large surface area of the large glass slides allows for a flat focal plane so all cells in the frame-of-view can be in focus. In fact, it became somewhat of a religious result for me as having a flat pad is key to getting a good dataset, especially for timelapses that run overnight and are subject to going out of focus.

Prepare materials:

- 2 large glass slides (75 × 50 × 1 mm)
- glass slide for imaging (3" × 1" × 1.2mm - love the units)
- #1.5 glass coverslip
- agarose powder
- LB/M9/growth media of choice
- 5mL Erlenmeyer flask
- razor blade
- tape
- hot glue gun

- binder clips

Create agarose gel:

1. Wrap two pieces of tape around one large glass slide, two times (total of four pieces of tape, with each side of the glass having 2 layers of tape).

2. Mix agarose powder to approximately 2-4% in growth media. For example, use 5mL growth media : 0.15g agarose powder. Microwave about 30s several times until evenly melted, swirling to mix in between each microwave session. It may help to lightly stuff a moist paper towel at the opening of the flask to avoid evaporation of the water in the gel.

3. Pipette 1mL melted agarose solution onto coverglass with tape.

4. Place second large glass slide on top of tape and squish together with binder clips to ensure even gel surface.

5. Wait a few minutes for gel to solidify.

6. Remove top large glass slide by sliding off gently, to not damage the surface of the gel.

7. Cut piece of gel off with razor blade. Can seal gel with both glass slides in a zipped bag in the refrigerator for a few days.

Prepare imaging slide:

1. Place piece of gel on imaging glass slide using razor blade, gently laying to ensure flatness.

2. Pipette 1µL cells onto gel - they should not be too dense in solution.

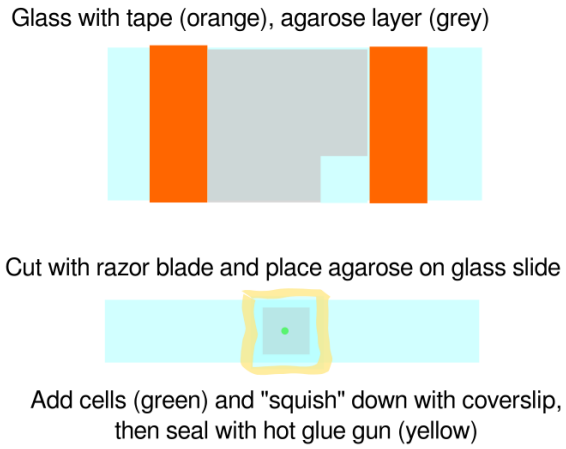



Figure B.1: Super flat pad technique.

3. Wait about 1 minute to dry slightly, then place cover slip on top and use razor blade to squish residual liquid of cells into pad or over the edge of the pad.

4. Seal coverslip with hot glue gun.

# Appendix C

# BRIGHTFIELD TRAINING PIPELINE

The following is a rough protocol for how I trained the Omnipose brightfield model. I highly recommend using a Jupyter notebook.

1. Crop training image stacks and fluorescence images to roughly 512 × 512 px. For example, a 2048 × 2048 image will be cropped into 16 subparts.

2. Run Omnipose phase-contrast model on overfocused (cells dark) images to generate rough masks. Convert to ncolor.

3. Open image stack and mask in Napari. Manually correct errors while checking against fluorescence signal.

4. Convert ncolor mask back to regular uint mask (`ncolor.format_labels(masks,clean=True)`). Save corrected mask as 'mask_edited' to distinguish from uncorrected mask.

5. Separate out tif stack into planes and channels, and save all tifs and masks in a ground truth folder. Keep in subfolders for each z: z0, z1, ...; each subfolder should contain image and masks file pairs.

6. Account for possible shift between z-planes. Do a DIY style transfer by renormalizing mask image to have dark cells with a grey background: `maskSim = label2binary(mask, 0.43)`. Invert underfocused (cells bright) image. Use

`SymmetricDiffeomorphicRegistration` from `dipy.align.imwarp` to determine mapping for registration between mask and inverted image. Save out as '_warped' to distinguish from unregistered image.

7. Train Omnipose on ground truth folders. Training command is run in CLI:

```
nohup omnipose --train --use_gpu
--dir /home/tlo/Documents/trainingdirectory --mask_filter labelmasks
--img_filter bf_warped --n_epochs 4000 --pretrained_model None
--nchan 1 --learning_rate 0.1 --diameter 0 --batch_size 16
--RAdam --all_channels --look_one_level_down
--dataloader --num_workers 8 &
```

8. Test performance of different models trained with different ground truth.

9. Can use intermediate trained model to generate rough masks for unannotated ground truth.

# Appendix D

## RECOMMENDED CLASSES TO TA

Classes that are alright to teach: PHYS 543 with Anna Goussiou, PHYS 334 with Jens Gundlach/Blayne Heckel, PHYS 114 tutorial, PHYS 231 with Suzanne White Brahmia, Paul's microscopy class.

Classes I did not enjoy teaching/are time consuming: PHYS 117/121 tutorial and lab. PHYS 323/325 depending on instructor.

# Appendix E

## RESTAURANTS & BOBA

My top restaurants on the Ave, in no particular order: Taste of Xian, Sizzle & Crunch, Chipotle, Nuodle, Little Thai, The BoB, Aladdin Gyro, Cedars of Lebanon, Shawarma King (lower Ave one), Time Bistro, Six Pack Foods Company, A-Pizza Mart, Araya's Place.

Special mention to Little Duck.

Best boba shops along the Ave, in order: TP Tea, Seattle Best Tea, Boba Gem, Cafe Happy, Yifang Taiwan Fruit Tea, Dont Yell At Me.

Special mention to Timeless Tea.