Supplementary Materials for Simultaneous cross-evaluation of heterogeneous *E. coli* datasets via mechanistic simulation

Derek N. Macklin[1†], Travis A. Ahn-Horst[2,3†], Heejo Choi[2,3†], Nicholas A. Ruggero[4†], Javier Carrera[5†], John C. Mason[6†], Gwanggyu Sun[2,3], Eran Agmon[2,3], Mialy M. DeFelice[2,3], Inbal Maayan[2,3], Keara Lane[7], Ryan K. Spangler[2,3], Taryn E. Gillies[2,3], Morgan L. Paull[8], Sajia Akhter[2], Samuel R. Bray[2], Daniel S. Weaver[4], Ingrid M. Keseler[9], Peter D. Karp[9], Jerry H. Morrison[3], Markus W. Covert[2,3*]

[1]Grand Rounds, Inc, San Francisco, CA 94107, USA

[2]Department of Bioengineering, Stanford University, Stanford, CA 94305, USA

[3]Allen Discovery Center at Stanford University, Stanford University, Stanford, CA 94305, USA

[4]X, Mountain View, CA 94305, USA

[5]Zymergen, Emeryville, CA 94608, USA

[6]Intrexon, South San Francisco CA 94080, USA

[7]Department of Molecular Biosciences, Northwestern University, Evanston, IL 60208, USA

[8]BridgeBio Pharma, Palo Alto, CA 94301, USA

[9]SRI International, Menlo Park, CA 94025, USA

[†]These authors contributed equally to this work.

[*]Correspondence to: mcovert@stanford.edu.

This PDF file includes:

Contents

1 Methods

Our methods have been organized into three groups to facilitate navigation of this document: (1) computational and modeling methods, which describe how our model was created, parameterized and used to run simulations; (2) simulation analysis methods, which describe in detail how all the work described in the main text was performed; and (3) experimental methods, which details our protein half-life and mRNA expression work.

1.1 Computational and Modeling Methods

1.1.1 Overview

We constructed a large-scale, integrated mathematical model of a cell that integrated as much organism-specific data as possible, across multiple environments and growth conditions, in which results and predictions could be experimentally verified. This effort required major innovation in data integration, modeling framework extensibility, and feedback regulation, all of which are described in more detail here. This text also serves as a companion to our source code (https://github.com/CovertLab/WholeCellEcoliRelease).

Running our large-scale cell model may be thought of as similar (but not strictly equivalent) to numerical integration of a system of ordinary differential equations, where the cellular states are analogous to the ODE state variables and the cellular processes are analogous to the differential equations (*1*). The most significant difference from ODE numerical integration is that shared resources stored in cellular states must be partitioned to each cellular process in order to ensure mass conservation (described in more detail below).

The model is primarily implemented in Python, with Cython used for computationally intensive inner loops. Workflows are defined, managed, and executed using FireWorks - a free open-source code for automating workflow execution which can be defined in Python (*2*). The model uses an application program interface (API) developed to facilitate model development, human readability, and consistent coding style.

1.1.2 Improvements over previous work

For readers familiar with our *M. genitalium* model (*3*), we summarize our improvements over that model in terms of modeling and computation briefly here. The most important modeling improvements are as follows:

- The optimization method used to simulate behavior of the metabolic model is much more robust and includes detailed quantitative (Michaelis-Menten) parameters for 380 reactions. The metabolic model now maintains concentrations of metabolite pools subject to resource availability rather than producing metabolites in a fixed ratio at every time step. This enables the metabolic model to adjust to time-dependent/cell cycle-dependent behavior from other simulated processes while maintaining homeostasis.

- Our model of translation uses translational efficiency data to inform ribosome binding to mRNA transcripts.

- We have a more detailed model of RNA decay that incorporates both the rates of degradation due to endonuclease-mediated cleavage and the rates of transcript digestion by exoRNases.

- Our simulated *E. coli* cells grow at different doubling times as a function of the environment. This improvement, supported by our model of DNA replication which can track multiple rounds of replication, is showcased in the main text and was not possible in the *M. genitalium* simulation.

- We have a quantitative model of transcriptional regulation that incorporates the function of 22 transcription factors regulating 355 genes. This includes one- and two-component signaling processes to modulate transcription factor activity, as well as the modulation of RNA polymerase recruitment via TF-DNA binding interactions.

Similarly, the major computational improvements are listed here:

- We have decreased simulation run-time by nearly two orders of magnitude. Whereas *M. genitalium* simulations took roughly 10 hours to run, *E. coli* simulations—which account for 50 times more molecules—take approximately 15 minutes to simulate the life cycle of an *E. coli* cell. We achieved this by improving file I/O, writing inner loops in Cython or C, and warm-starting the linear solver in our metabolic model.
- This improvement in run-time, together with recently published insights into cell growth and division (*4, 5*) enables us to reliably simulate multiple generations of cells (as showcased in the main text), which was not possible with the *M. genitalium* simulations.
- Additionally, we have improved the whole-cell application programming interface (API). Code is much more readable, and on-boarding new researchers takes roughly 2 weeks rather than 6 months.

Finally, while the model presented here is not gene-complete, it incorporates the biological processes for which the majority of high-throughput data is available. We spent considerable effort evaluating data sets and merging them into our framework. Ultimately, this enabled us to make the quantitative comparisons presented in the main text—comparisons that could not be made when modeling *M. genitalium*. However, this version of the *E. coli* model lacks several of the sub-models implemented in the *M. genitalium* model and additional *E. coli* specific gene functionality such as detailed DNA interactions, antibiotic resistance, protein maturation, toxin/antitoxin pairs, stress response and physical structures of the cell. We continue working toward a gene-complete model of *E. coli* and until all functionality is included, certain predictions will lack the desired accuracy such as growth in the presence of antibiotics or other environmental stresses and simulations of gene knockouts for genes without a functional implementation. A list of functionally implemented genes is provided in Table S6.

1.1.3 Model construction

To construct the model and identify parameters, we begin with data sets from the primary literature, our own experiments, and databases (e.g., EcoCyc) which we unify into a KnowledgeBase. The bulk of the experimental data we used to parameterize the model comes from three lab strains of *E. coli*: K-12 MG1655, B/r, and BW25113. The model can therefore be thought of as a composite strain which uses all of these data. One drawback of this approach is that in some cases these strains have different physiology, with the most notable example being a difference in growth rate between MG1655 (where most high-throughput data comes from) and B/r (where detailed composition data have been obtained) under similar environmental conditions. Our model was optimized using the B/r growth rates.

1.1.4 Simulation algorithm

As mentioned above, a whole-cell model may be thought of as similar to a system of ordinary differential equations (ODEs) where the cellular states are analogous to the ODE state variables and the cellular processes are analogous to the differential equations. Extending this analogy, the *E. coli* model is simulated using an algorithm that is comparable to those used to numerically integrate ODEs. The only significant difference from ODE numerical integration is that shared resources stored in cellular states must be partitioned to each cellular process in order to ensure mass conservation. Critically, we make the assumption that over a short time scale, each process acts independently and that the distribution of partitioned molecules maximally satisfies the relative needs of the processes, with higher priorities for some processes.

Algorithm 1 summarizes the simulation algorithm to execute a time step. The temporal evolution of the cell state is calculated on a short time scale (typically $<$1 second) by allocating cell state variables among processes (described in Algorithm 1 under “Allocate shared resources”), and executing the process code that updates counts in the state variables until the cell divides. Here, “demand” refers to the number of molecules (amino acids, ATP, ribosomal subunits, for example) requested by each process when assuming full access to all molecules in the cell at that time step, providing a quantifiable idea of the maximum number of molecules each process is capable of processing. These demands are then used to compare requests for each molecule-type across different processes, which is used to partition the limited amount of shared resources to each cellular process, and thus ensure mass conservation. If the total demand is greater than the availability for a given resource, then the resource is divided between processes proportionally to the demand (e.g., if one process is responsible for half of the overall demand for a particular molecule, that process will receive one half of the limited resource); otherwise, no restriction of supply occurs. Metabolism is assigned the lowest request priority and is left with whatever molecules remain after partitioning to the other processes, further guaranteeing other processes their requested molecules.

Defining four critical parts of the simulation code will help to better understand our approach:

States. The simulation States are defined as the counts, locations, and attributes of every species in the model at a given time step, which are then operated on by $\mathrm{Processes}$. There are two classes of States within the model - $\mathrm{BulkMolecules}$ and $\mathrm{UniqueMolecules}$. The $\mathrm{BulkMolecules}$ state tracks species in the simulation where individuals are not further distinguished from each other. For example, two ATP molecules in the cytoplasm are considered identical and tracked in $\mathrm{BulkMolecules}$. The $\mathrm{UniqueMolecules}$ state tracks species in the simulation where individuals are distinguishable from each other by an attribute and cannot be interchanged without effect. For example, two ribosomes on different mRNA transcripts are uniquely identified by the transcript they are translating and their location on the transcript.

Processes. The simulation Processes update the simulation States from one time step to the next. Each Process represents an aspect of physiology of an *E. coli* cell. We discuss the implementation of each process in detail in Section 1.1.6.

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

Views. Views is a class that provides a programmatic interface that allow States and Processes to interact cleanly during simulation.

Algorithm 1: Whole-cell dynamic simulation

```
Initialize simulation states (described in Algorithm 2, 3, 4)
repeat
    /* Allocate shared resources */
    for each molecule i do
        for each process j do
            1. Calculate demand $d_{i,j}$ of process $j$ for molecule $i$.
            2. Divide total count $c_i$ of molecule $i$ into partition $p_{i,j}$, for each process
               proportional to the demand such that $p_{i,j} = c_i \frac{d_{i,j}}{\sum\limits_{j} d_{i,j}}$.
    /* Calculate temporal evolution */
    for each process j do
        1. Retrieve partitioned molecules $p_{i,j}$.
        2. Compute the contribution of process $j$ to the temporal evolution of the
           partitioned molecules $\Delta p_{i,j}$.
        3. Update partitioned molecule counts $p_{i,j} = p_{i,j} + \Delta p_{i,j}$.
    /* Merge partitioned molecules */
    for each molecule i do
        Update counts $c_i$ based on updated partitions computed in each process,
          $c_i = \sum\limits_{j} p_{i,j}$
    Increment simulation step by 1
until cell division;
Result: Whole-cell model is executed for one cell cycle
```

1.1.5 Initializing simulations

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

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

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

where M_{total} is the total mass of RNA or protein, c_i and MW_i are the counts and molecular weight of RNA or protein i, and N_a is Avogadro's number. The counts of the i-th RNA or protein can be described as a fraction of the total counts (c_{total}) of RNA or protein: $c_i = c_{total} \cdot f_i$, where f_i is the mass fraction of RNA or protein i. Substituting c_i in Equation 1 and rearranging gives Equation 2:

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

where $\vec{f}$ is the distribution of RNA or protein per cell. The total masses of RNA and protein per cell (M_{total}) are known from the $\mathrm{KnowledgeBase}$. The RNA distribution $\vec{f}^{RNA}$ is computed from a reconciliation of the RNA-sequencing dataset measured in this study with the expectation that the expression of RNA polymerases and ribosomes must be sufficient to meet cellular demands (please see the iterative parameter estimation approach described in Section 1.2.2). The protein distribution $\vec{f}^{protein}$ is computed from $\vec{f}^{RNA}$ according to the following equation, which is an abbreviated form of the protein equation in Fig. 1 for the ith protein:

$$\frac{d\text{protein}_i}{dt} = a \cdot \text{mRNA}_i \cdot \psi_i - \left(\frac{\ln 2}{t_{1/2,i}} + \frac{\ln 2}{\tau}\right)\text{protein}_i \tag{3}$$

where mRNA$_i$ is the concentration of the ith mRNA from which protein i is translated, ψ_i is the translation efficiency of mRNA$_i$, $t_{1/2,i}$ is the half-life of protein$_i$, and τ is the doubling time of the cell. The constant a represents the remaining parameters in the protein equation in Fig. 1 and is assumed to be the same across all protein types. At steady-state, $\frac{d\text{protein}_i}{dt} = 0$ and protein$_i$

can be isolated:

$$\text{protein}_i = \frac{a \cdot \text{mRNA}_i \cdot \psi_i}{\frac{\ln 2}{t_{1/2,i}} + \frac{\ln 2}{\tau}} \tag{4}$$

The protein distribution $\vec{f}^{protein}$ is obtained from the per-protein expression:

$$\vec{f}^{protein} = \frac{1}{\sum_{i=1}^{n} \text{protein}_i} \cdot \begin{bmatrix} \text{protein}_1 \\ \text{protein}_2 \\ \vdots \\ \text{protein}_n \end{bmatrix} \tag{5}$$

where n is the number of protein monomers (4353). Since a is a constant, the protein distribution expression simplifies to:

$$\vec{f}^{protein} = \text{normalize}\left(\frac{\vec{f}^{RNA} \cdot \vec{\psi}}{\vec{k_d} + \frac{\ln 2}{\tau}}\right) \tag{6}$$

where $\vec{\psi}$ is the translation efficiency distribution, and $\vec{k_d}$ is a vector of the protein degradation rates. Total counts and the expected distribution are then used to sample a $\mathrm{multinomial}$ distribution to statistically compute the counts of each individual RNA or protein species using Equation 7.

$$\vec{c} = \mathrm{multinomial}(\vec{f}, c_{total}) \tag{7}$$

Details can be found in Algorithm 2.

```
Algorithm 2: Initializing counts of RNA and protein
  Input : $M_{total}^{RNA}$, $M_{total}^{Protein}$ Total mass per cell of RNA and protein
  Input : $MW_i, MW_j$ Molar molecular weights of RNA $i = 1$ to $n_{RNA}$ and protein $j = 1$ to $n_{protein}$
  Input : $f_i^{RNA}$ mass fraction based on RNA expression of RNA $i = 1$ to $n_{RNA}$
  Input : $k_{d,j}$ degradation rate of protein $j = 1$ to $n_{protein}$
  Input : $N_a$ Avogadro's number
  Input : $\psi_j$ translational efficiencies of each mRNA $j = 1$ to $n_{protein}$
  1. Calculate total counts of RNAs ($C_{total}^{RNA}$) based on total RNA mass ($M_{total}^{RNA}$) and distribution of expression ($\vec{f}^{RNA}$) from KnowledgeBase
      $c_{total}^{RNA} = \frac{M_{total}^{RNA}}{\vec{f}^{RNA} \cdot \vec{MW}/N_a}$
  2. Calculate counts of each RNA ($\vec{c}_{RNA}$) by sampling a multinomial distribution $c_{total}^{RNA}$ times weighted by the expected distribution of expression ($\vec{f}^{RNA}$).
      $\vec{c}_{RNA}$ = $\mathrm{multinomial}\ (\vec{f}^{RNA}, c_{total}^{RNA})$
  3. Calculate expected distribution of protein counts ($\vec{f}^{protein}$) based on expected distribution of RNA counts ($\vec{f}^{RNA}$), translational efficiencies ($\vec{\psi}$), protein degradation rates ($\vec{k_d}$), and dilution using a steady state assumption.
      $\vec{f}^{protein} = \frac{\vec{f}^{RNA} \cdot \vec{\psi}}{\vec{k_d} + \frac{ln(2)}{\tau}}$
  4. Calculate total counts of proteins ($c_{total}^{protein}$) based on total protein mass ($M_{total}^{protein}$) and distribution of counts ($\vec{f}^{protein}$).
      $c_{total}^{protein} = \frac{M_{total}^{protein}}{\vec{f}^{protein} \cdot \vec{MW}/N_a}$
  5. Calculate counts of each protein ($\vec{c}_{protein}$) by sampling a multinomial distribution $c_{total}^{protein}$ times weighted by the expected distribution of expression ($\vec{f}^{protein}$).
      $\vec{c}_{protein}$ = $\mathrm{multinomial}\ (\vec{f}^{protein}, c_{total}^{protein})$
  Result: Counts of RNA are set at the beginning of the first generation of simulated cells
```

Initializing small molecule counts. The counts of small molecules such as cytoplasmic and membrane constituents are initialized as follows. Expected concentrations of small molecules are either known experimentally, or computed from an FBA biomass reaction (see Section 1.1.6.7) and stored as a reconciled dataset in the KnowledgeBase. The volume of the cell is computed using its mass divided by its density. Therefore adding counts of small molecules to the cell in order to match a concentration will necessarily change the volume of the cell. To reconcile this inconsistency, the counts of each small molecule are recalculated by equating the following two expressions for total mass of small molecules:

$$\begin{cases} m_s = m_t - m_{ns} \\ m_s = \sum_{i=1}^{n_s} m_i \end{cases} \tag{8}$$

where m_s is the total mass of small molecules, m_t is the total mass of the cell, m_{ns} is the total mass of non-small molecules, n_s is the total number of small molecule types, and m_i is the mass of the ith small molecule type. Since the total mass of the cell m_t is equal to $\rho * V$, where ρ is the cell density and V is the cell volume, the first expression can be rewritten as:

$$m_s = \rho * V - m_{ns} \tag{9}$$

The mass of each small molecule type m_i can be described as:

$$m_i = c_i \cdot V * w_i \tag{10}$$

where c_i is the concentration and w_i is the molecular weight of small molecule i. The two expressions in Equation 8 can be rewritten as:

$$\begin{cases} m_s = \rho * V - m_{ns} \\ m_s = V \cdot (w^T \bullet c) \end{cases} \tag{11}$$

where w and c are vectors of length n_s and describe the molecular weights and concentrations of small molecules respectively. Equating the two expressions in Equation 11 and solving for V yields:

$$V = \frac{m_{ns}}{\rho - (w^T \bullet c)} \tag{12}$$

which allows the calculation of small molecule counts as described in Algorithm 3.

```
Algorithm 3: Initializing counts of small molecules
  Input : $MW_k$ molecular weight of small molecule $k = 1$ to $n_s$
  Input : $C_k$ concentration of small molecule $k = 1$ to $n_s$
  Input : $m_{ns}$ total mass of non-small molecules of the cell (ie. only considering RNA,
          protein, and DNA)
  Input : $\rho$ cell density
  Input : $N_a$ Avogadro's number
  1. Calculate the total small molecule mass concentration (partial density).
          $\rho_s = \sum_{k=1}^{n_s} MW_k \cdot C_k$
  2. Calculate the new total cell volume that accommodates the small molecule
     concentrations.
          $V = \frac{m_{ns}}{\rho - \rho_s}$
  3. Calculate the new counts of each small molecule $k$.
          $c_k^s = C_k \cdot V \cdot N_a$
  Result: Consistent small molecule counts $c_k^s$
```

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

First, the number of rounds of replication that on average need to proceed simultaneously can be estimated in an average cell in a population using the length of time required to replicate the chromosome (C period) and the length of time for cytokinesis (D period) as well as the expected doubling time given the environment (τ). The number of simultaneous rounds (n_{limit}) can be calculated with Equation 13 as the ratio of C+D period over the doubling time (*6*).

Because we are considering a specific cell and not an average of a population of cells, the number of rounds of replication needs to be an integer, and we take the floor because a fractional round of chromosome initiation has not yet occurred.

$$n_{limit} = floor(\frac{C + D}{\tau}) \tag{13}$$

For every round of replication proceeding there are a pair of replication forks and a pair of origins of replication. We are assuming that on average a cell after division has inherited one chromosome molecule (i.e. no more than one terC), and that it may have more than one round of replication proceeding on it (i.e. number of oriC ≥ 1). Therefore the number of origins of replication (n_{origin}) is defined by Equation 14.

$$n_{origin} = 2^{n_{limit}} \tag{14}$$

Finally, the position between the oriC and the terC of each replication fork needs to be determined on average. This can be calculated with Equation 15 where f is the fraction of length between the origin and terminus of replication that the replication fork has proceeded for the nth round of replication.

$$f = 1 - \frac{n \cdot \tau - D}{C} \tag{15}$$

Where n is every integer value 1,2,...n_{limit}. The position in nucleotides (l) can then be calculated from Equation 16 where L is the total length of the chromosome in *E. coli*.

$$l = f \cdot \frac{L}{2} \tag{16}$$

Proper initialization of the cell ensures the simulation begins close to the steady state of the system, and in practice the simulation is relatively stable. Perturbations in the ratio of cell mass to number of origins of replication quickly re-converge to steady state for a given environment. A detailed algorithm for chromosome initialization can be found in Algorithm 4.

Algorithm 4: Initializing chromosome state

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

Environments. The model can simulate log-phase growth in three different environments - M9 minimal media with glucose, M9 minimal media with glucose supplemented with amino acids (5x Supplement EZ without VA Vitamin Solution), and M9 minimal media with glucose without oxygen - which map to three different growth rates (doubling times of: 44 minutes, 25 minutes, and 100 minutes, respectively) to demonstrate the ability to shift cell composition. Metabolites included in these environments are glucose, oxygen, ammonium, Na^{+}, PI, K^{+}, sulfate, Fe^{2+}, Ca^{2+}, Cl^{-}, Co^{2+}, Mg^{2+}, Mn^{2+}, Ni^{2+}, Zn^{2+}, water, CO$_2$, MOPS, L-selenocysteine, glucose-D-lactone, cytosine, and the 20 amino acids. Please see Section 1.3.1 for detailed media conditions. In benchmarking our simulations, we also simulated environments that would activate and inactivate each of the transcription factors (not shown).

Exchange flux bounds for the metabolic network are either added or removed based on the presence of metabolites in the simulated media (see Section 1.1.6.7). The files specifying these bounds can be found in the repository (wcEcoli/reconstruction/ecoli/flat/condition/nutrient/). Based on these bounds, the Metabolism process updates cellular concentrations of small molecules, and the Transcription and Translation processes modulate the activities and expression of RNA polymerases and ribosomes to globally shift cell composition. The transcriptional regulatory network (see Section 1.1.6.2) responds to the new small molecule concentrations and adjusts gene expression appropriately.

1.1.6 Processes

The Processes of the *E. coli* model span several major areas of cellular physiology. Each process is a computational representation of chemical reactions or transformations grouped by a physiological function. In this section, the processes are ordered according to their association with the following groups of cellular function: Central Dogma, Metabolism, and Balanced Growth. We modeled Processes using the most appropriate mathematics for their individual network topology and degree of experimental characterization. The inputs and outputs of each Process are the counts of metabolites or macromolecules and the catalytic capacity or configuration of the enzymes that catalyze the reactions in each Process. This section details the model implementation, computational algorithm, associated data, and relevant code for each Process. Some of the data values for these processes are noted with “See GitHub” indicating that they are tabulated or calculated in files in our GitHub repository under wcEcoli/reconstruction/ecoli/.

1.1.6.1 Transcription

Model implementation. The *E. coli* model assumes RNA polymerase exists in two states: free and actively transcribing. Every time step, free RNA polymerase transitions to the actively transcribing state to maintain an experimentally-observed active fraction of RNA polymerase. This is a simplification compared to *M. genitalium* model, which modeled RNA polymerase as existing in 4 states: free, non-specifically bound on a chromosome, bound to a promoter, and actively transcribing a gene. The *E. coli* model does not yet include sigma, elongation or termination factors. The *E. coli* model also currently treats each gene as its own transcription unit. Transcription occurs through the action of two processes in the model: $\mathrm{TranscriptInitiation}$ (Algorithm 5) and $\mathrm{TrancriptElongation}$ (Algorithm 6).

Initiation. $\mathrm{TranscriptInitiation}$ models the binding of RNA polymerase to each gene. The number of initiation events per gene is determined in a probabilistic manner and dependent on the number of free RNA polymerases and each gene's synthesis probability. The number of RNA polymerases to activate in each time step is determined such that the average fraction of RNA polymerases that are active throughout the simulation matches measured fractions, which are dependent on the cellular growth rate. This is done by assuming a steady state concentration of active RNA polymerases (and therefore a constant active fraction):

$$\frac{dR_{act}}{dt} = p_{act} \cdot R_{free} - r \cdot R_{act} = 0 \tag{17}$$

$$p_{act} = \frac{r \cdot R_{act}}{R_{free}} \tag{18}$$

where R_{act} is the concentration of active RNA polymerases, R_{free} is the concentration of free RNA polymerases, p_{act} is the activation probability and r is the expected termination rate for active RNA polymerases. Using the definition of the active fraction, $f_{act} = \frac{R_{act}}{R_{act}+R_{free}}$, p_{act} can be defined in terms of the desired active fraction:

$$p_{act} = \frac{r \cdot f_{act}}{1 - f_{act}} \tag{19}$$

This activation probability is then used to determine how many free RNA polymerases will initiate. These newly initiated RNA polymerases are distributed to individual genes based on the synthesis probability for each gene, which is determined based on another steady state assumption for each mRNA concentration:

$$\frac{dm_i}{dt} = v_{synth,i} - m_i \cdot \left(\frac{\ln 2}{\tau} + \frac{\ln 2}{t_{\frac{1}{2},i}} \right) = 0 \tag{20}$$

$$v_{synth,i} = m_i \cdot \left(\frac{\ln 2}{\tau} + \frac{\ln 2}{t_{\frac{1}{2},i}} \right) \tag{21}$$

where $v_{synth,i}$ is the synthesis rate of each mRNA, m_i is the concentration of each mRNA, τ is the doubling time and $t_{\frac{1}{2},i}$ is the half life for each mRNA (see Section 1.1.6.3). Using RNA expression data for m_i, the rate of synthesis for each gene can be determined. Synthesis rates are then normalized as below to determine a synthesis probability for each gene:

$$p_{synth,i} = \frac{v_{synth,i}}{\sum\limits_j v_{synth,j}} \tag{22}$$

where $p_{synth,i}$ is the synthesis probability for each gene. Gene synthesis probabilities are further dependent on transcription factor binding and regulation as discussed in the next section (Section 1.1.6.2).

Elongation. $\mathrm{TranscriptElongation}$ models nucleotide polymerization into RNA molecules by RNA polymerases. Polymerization occurs across all polymerases simultaneously and resources are allocated to maximize the progress of all polymerases up to the limit of the expected polymerase elongation rate and available nucleotides. The termination of RNA elongation occurs once a RNA polymerase has reached the end of the annotated gene.

```
Algorithm 5: RNA polymerase initiation on DNA
  Input : $f_{act}$ fraction of RNA polymerases that are active
  Input : $r$ expected termination rate for active RNA polymerases
  Input : $p_{\text{synth,i}}$ RNA synthesis probability for each gene where $i = 1$ to $n_{gene}$
  Input : $c_{RNAP,f}$ count of free RNA polymerase
  Input : $\mathrm{multinomial}()$ function that draws samples from a multinomial distribution
  1. Calculate probability ($p_{act}$) of a free RNA polymerase binding to a gene.
        $p_{act} = \frac{r \cdot f_{act}}{1 - f_{act}}$
  2. Calculate the number of RNA polymerases that will bind and activate ($c_{RNAP,b}$).
        $c_{RNAP,b} = p_{act} \cdot c_{RNAP,f}$
  3 Sample multinomial distribution $c_{RNAP,b}$ times weighted by $p_{synth,i}$ to determine
    which genes receive a RNA polymerase and initiate ($n_{init,i}$).
        $n_{init,i} = \mathrm{multinomial}(c_{RNAP,b}, p_{synth,i})$
  4 Assign $n_{init,i}$ RNA polymerases to gene $i$. Decrement free RNA polymerase counts.
  Result: RNA polymerases bind to genes based on the number of free RNA
          polymerases and the synthesis probability for each gene.
```

Algorithm 6: mRNA elongation and termination

```
Input : $e$ expected RNA polymerase elongation rate in given environment
Input : $L_i$ length of each gene $i = 1$ to $n_{gene}$ for each coding gene.
Input : $p_j$ gene position of RNA polymerase $j = 1$ to $n_{RNAP}$
Input : $c_{nt,k}$ counts of nucleotides $k = 1$ to $4$ for each nucleotide type (A, C, G, U)
Input : $\Delta t$ length of current time step
/* Elongate RNA transcripts up to limits of sequence or nucleotides */
for each RNA polymerase j on gene i do
    1. Based on RNA polymerase position $p_j$ on a gene $i$ and maximal elongation rate $e$
       determine "stop condition" ($s_j$) for RNA polymerase $j$ assuming no nucleotide
       limitation.
            $s_j = \min(p_j + e \cdot \Delta t,\, L_i)$
    Stop condition is either maximal elongation rate scaled by the time step or the full
       length of sequence (i.e. the RNA polymerase will terminate in this time step).
    2. Derive sequence between RNA polymerase position ($p_j$) and stop condition ($s_j$).
    3. Based on derived sequence calculate the number of nucleotides required to
       polymerize sequence $c_{nt,k}^{req}$.
    4. Elongate up to limits:
    if $\mathrm{all}(c_{nt,k}^{req} < c_{nt,k})$ then
        Update the position of each polymerase to stop position
            $p_j = s_j$
    else
        4a. Attempt to elongate all RNA fragments.
        4b. Update position of each polymerase to maximal position given the
            limitation of $c_{nt,k}$.
    5. Update counts of $c_{nt,k}$ to reflect polymerization usage.
/* Terminate RNA polymerases that have reached the end of their gene */
for each RNA polymerase j on gene i do
    if $p_j$ == $L_i$ then
        1. Increment count of RNA that corresponds to elongating RNA transcript that
           has terminated.
        2. Increment free RNA polymerase counts.
Result: Each RNA transcript is elongated up to the limit of available gene sequence,
        expected elongation rate, or nucleotide limitation. RNA polymerases that reach
        the end of their genes are terminated and released.
```

Associated data

Parameter	Symbol	Units	Value	Reference
Active fraction of RNAP	f_{act}	-	[0.17, 0.30]	(*7*)
RNA synthesis probability	p_{synth}	-	[0, 0.015]	See GitHub
RNAP elongation rate	e	nt/s	[39, 55]	(*7*)

Table 1: Table of parameters for Transcript Initiation and Elongation processes.

Associated files

wcEcoli Path	File	Type
wcEcoli/models/ecoli/processes	transcript_initiation.py	process
wcEcoli/models/ecoli/processes	transcript_elongation.py	process
wcEcoli/reconstruction/ecoli/dataclasses/process	transcription.py	data
wcEcoli/reconstruction/ecoli/flat	rnas.tsv	raw data
wcEcoli/reconstruction/ecoli/flat	growthRateDependentParameters.tsv	raw data

Table 2: Table of files for transcription.

1.1.6.2 Transcription regulation

Model implementation. There are two aspects to modeling transcriptional regulation: (1) modeling the activation or inhibition of a transcription factor (e.g., by a ligand), and (2) given an active transcription factor, modeling its effect on RNA polymerase recruitment to a promoter site. The the enhanced coverage of the regulatory network - 438 regulatory interactions described by 22 transcription factors that regulate 355 genes - is a significant difference from the *M. genitalium* model. To incorporate this network, regulation is represented by three different classes of transcription regulators: zero-component systems, one-component systems and two-component systems.

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

1. **Zero-component systems**: transcription factors that are considered to be active whenever they are expressed. Examples include the Fis and Hns proteins. These two proteins,

for instance, are important in maintaining higher-order DNA structure and likely have complex feedback loops modulating their activity. Because this complexity is not yet fully understood, we make the simplifying assumption that these proteins are always active unless they are knocked out. Zero-component systems are modeled in the TfBinding process in the model, which handles transcription factor binding to promoters.

2. **One-component systems**: transcription factors that are directly activated or inhibited by a small molecule ligand. Examples of this class include the repressor TrpR which binds tryptophan, and the inducer AraC which binds arabinose. One-component systems are modeled in the TfBinding and Equilibrium processes in the model, which handle transcription factor binding to promoters and transcription factor binding to ligands, respectively.

3. **Two-component systems**: transcription factors that are paired with a separate sensing protein that responds to an environmental stimulus (these are simple analogs to the vast, complicated signaling networks that exist in eukaryotic cells). The sensing protein phosphorylates the cognate transcription factor in a condition-dependent fashion. Examples include ArcA which is phosphorylated by its cognate ArcB in anaerobic conditions, and NarL which responds to the presence of nitrate when phosphorylated by its cognate sensor NarX. Two-component systems are modeled in the TfBinding, Equilibrium, and TwoComponentSystems processes in the model, which handle transcription factor binding to promoters, transcription factor binding to ligands, and phosphotransfer reactions of signaling pathways, respectively.

Zero-component systems. We assume all transcription factors of this class will bind to available promoter sites.

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

$$nL + T \rightleftharpoons T^* \tag{23}$$

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

With the dissociation constant K_d defined as:

$$K_d = \frac{L^n \cdot T}{T^*} \tag{24}$$

we have:

$$\frac{T^*}{T_T} = \frac{L^n}{L^n + K_d} \tag{25}$$

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

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

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

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

$$L + S_f \rightleftharpoons S_b \tag{26}$$

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

$$S_f + ATP \overset{k_A}{\rightarrow} S_{fp} + ADP + H^+ \tag{27}$$

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

$$S_b + ATP \overset{k_B}{\rightarrow} S_{bp} + ADP + H^+ \tag{28}$$

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

$$S_{fp} + T \overset{k_C}{\rightarrow} S_f + T_p \tag{29}$$

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

$$S_{bp} + T \overset{k_D}{\rightarrow} S_b + T_p \tag{30}$$

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

$$T_p + H_2O \overset{k_E}{\rightarrow} T + P_i \tag{31}$$

Ligand binding is simulated in a fashion identical to the one-component systems. By assuming mass-action kinetics, we can represent the rest of this system mathematically using ordinary differential equations:

$$\frac{dS_f}{dt} = -k_A \cdot S_f \cdot ATP + k_C \cdot S_{fp} \cdot T \tag{32}$$

$$\frac{dS_b}{dt} = -k_B \cdot S_b \cdot ATP + k_D \cdot S_{bp} \cdot T \tag{33}$$

$$\frac{dT}{dt} = -k_C \cdot S_{fp} \cdot T - k_D \cdot S_{bp} \cdot T + k_E \cdot T_p \cdot H_2O \tag{34}$$

$$\frac{dS_{fp}}{dt} = -\frac{dS_f}{dt} \tag{35}$$

$$\frac{dS_{bp}}{dt} = -\frac{dS_b}{dt} \tag{36}$$

$$\frac{dT_p}{dt} = -\frac{dT}{dt} \tag{37}$$

This system of equations is simulated using a numerical ODE integrator (see Algorithm 8).

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

$$v_{\text{synth},j} = \alpha_j + \sum_i p_{T,i} \Delta r_{ij} \tag{38}$$

where α_j represents basal recruitment of RNA polymerase and the second term is dependent on transcription factor activity: the probability that the i^{th} transcription factor is DNA-bound is $p_{T,i}$, and the recruitment effect of the i^{th} transcription factor on the j^{th} gene is Δr_{ij}. The α and

Δr values are computed prior to simulation based on gene expression values from conditions that modulate transcription factor activity. Values for p_T are calculated as described in Table 3.

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

Table 3: Formulas used to compute the probability that a transcription factor is promoter-bound. T^* is the active form of a one-component system transcription factor, while T_p is the phosphorylated form of a two-component system transcription factor, and T is the inactive or unphosphorylated form of a transcription factor.

```
Algorithm 7: Equilibrium binding
Input : $c_m$ counts of molecules where $m = 1$ to $n_{molecules}$
Input : $S$ matrix describing reaction stoichiometries where $S[i, j]$ describes the
        coefficient for the $i^{th}$ molecule in the $j^{th}$ reaction
Input : $reactants_j$ set of indices for $c_m$ of reactant molecules that participate in the $j^{th}$
        reaction
Input : $product_j$ index for $c_m$ of the product molecule formed by the $j^{th}$ reaction
Input : $f$ conversion factor to convert molecule counts to concentrations
Input : $K_{d,j}$ dissociation constant where $j = 1$ to $n_{reactions}$
1. Dissociate all complexes in $c$ into constituent molecules to get total reactants ($d$)
  since some reactants participate in multiple reactions:
    $d = c$
    for each ligand-binding reaction, j do
        for each molecule, i do
            $d_i = d_i + c_{product_j} \cdot S[i, j]$
        $d_{product_j} = 0$
2. Find the number of reactions to perform ($n_j$) to minimize the distance from $K_{d,j}$,
  where $r$ is a positive integer and not greater than the total products that can be formed
  by the reactants:
    for each ligand-binding reaction, j do
        $n_j = \underset{r}{\operatorname{argmin}} \left| \frac{\prod_{i \in \text{reactants}_j} (f \cdot (d_i - r))^{S[i,j]}}{f \cdot r} - K_{d,j} \right|$
3. Update counts ($c$) based on number of reactions that will occur. Starting from the
  dissociated counts, reactants will decrease by the number of reactions and their
  stoichiometry and one product will be formed for each reaction.
    $c = d$
    for each ligand-binding reaction, j do
        for each molecule in $reactants_j$, i do
            $c_i = c_i - S[i, j] \cdot n_j$
        $c_{product_j} = n_j$
Result: Ligands are bound to or unbound from their binding partners in a fashion that
        maintains equilibrium.
```

Algorithm 8: Two-component systems

```
Input : $\Delta t$ length of current time step
Input : $c_m$ counts of molecules where $m = 1$ to $n_{molecules}$
Input : $k_A$ rate of phosphorylation of free histidine kinase
Input : $k_B$ rate of phosphorylation of ligand-bound histidine kinase
Input : $k_C$ rate of phosphotransfer from phosphorylated free histidine kinase to
        response regulator
Input : $k_D$ rate of phosphotransfer from phosphorylated ligand-bound histidine kinase
        to response regulator
Input : $k_E$ rate of dephosphorylation of phosphorylated response regulator
Input : $\mathrm{solveToNextTimeStep}()$ function that solves two-component system ordinary
        differential equations to the next time step and returns the change in molecule
        counts ($\Delta c_m$)
1. Solve the ordinary differential equations describing phosphotransfer reactions to
  perform reactions to the next time step ($\Delta t$) using $c_m$, $k_A$, $k_B$, $k_C$, $k_D$ and $k_E$.
     $\Delta c_m = \mathrm{solveToNextTimeStep}(c_m, k_A, k_B, k_C, k_D, k_E, \Delta t)$
2. Update molecule counts.
     $c_m = c_m + \Delta c_m$
Result: Phosphate groups are transferred from histidine kinases to response regulators
        and back in response to counts of ligand stimulants.
```

Algorithm 9: Transcription factor binding

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

Associated data

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

Table 4: Table of parameters for equilibrium binding, two-component systems, and transcription factor binding Processes. HK: histidine kinase, TF: transcription factor, HK-P: phosphorylated histidine kinase, TF-P: phosphorylated transcription factor. Note in this and future tables we reference the source code for our model, which will be freely available at GitHub as noted in the main text.

Associated files

wcEcoli Path	File	Type
wcEcoli/models/ecoli/processes	equilibrium.py	process
wcEcoli/models/ecoli/processes	tf_binding.py	process
wcEcoli/models/ecoli/processes	two_component_system.py	process
wcEcoli/reconstruction/ecoli/dataclasses/process	equilibrium.py	data
wcEcoli/reconstruction/ecoli/dataclasses/process	transcription_regulation.py	data
wcEcoli/reconstruction/ecoli/dataclasses/process	two_component_system.py	data
wcEcoli/reconstruction/ecoli/flat	equilibriumReactions.tsv	raw data
wcEcoli/reconstruction/ecoli/flat	foldChanges.tsv	raw data
wcEcoli/reconstruction/ecoli/flat	tfIds.tsv	raw data
wcEcoli/reconstruction/ecoli/flat	tfOneComponentBound.tsv	raw data
wcEcoli/reconstruction/ecoli/flat	twoComponentSystems.tsv	raw data
wcEcoli/reconstruction/ecoli/flat	twoComponentSystemTemplates.tsv	raw data

Table 5: Table of files for transcription regulation.

1.1.6.3 RNA degradation

Model Implementation. The *E. coli* model provides a more detailed, mechanistic representation of RNA degradation compared to the *M. genitalium* model. Unlike the previous model, the gene functionality of endoRNase and exoRNase is mechanistically integrated to evaluate: (1) rates of RNA degradation due to endo-nucleolytic cleavage, and (2) rates of nucleotides digested by exoRNases. These mechanisms are implemented in the $\mathrm{RnaDegradation}$ process (detailed in Algorithm 10).

Endo-nucelolytic Cleavage. RNAs are cleaved by nine different endoRNases, each of which are assumed to have the same rate of cleavage but can have a different specificity for cleavage of mRNAs, tRNAs or rRNAs. The rate of cleavage for each RNA is determined with a Michaelis-Menten kinetic equation:

$$r_i = k_{cat,endo} \cdot c_{endo,i} \cdot f_i \tag{39}$$

where i indicates the RNA from each gene, r_i is the rate of cleavage of each RNA species, $k_{cat,endo}$ is the rate of cleavage for a single endoRNase, $c_{endo,i}$ is the count of endoRNases specific to each RNA species, and f_i is the saturation of endoRNases for each RNA and is defined:

$$f_i = \frac{\frac{c_{RNA,i}}{K_{M,i}}}{1 + \sum\limits_j \frac{c_{RNA,j}}{K_{M,j}}} \tag{40}$$

where $c_{RNA,i}$ is the count of each RNA and $K_{M,i}$ is the Michaelis constant for each RNA, i. The saturation fraction accounts for competitive binding of each RNA species for the available endoRNases. The Michaelis constant for each RNA is determined by setting Eq. 39 equal to the first order approximation based on measured (or assumed, if measurement data is not available)

RNA half lives:

$$k_{cat,endo} \cdot c_{endo,i} \cdot \frac{\frac{c_{RNA,i}}{K_{M,i}}}{1 + \sum\limits_{j} \frac{c_{RNA,j}}{K_{M,j}}} = \frac{\ln 2}{\tau_{RNA,i}} \cdot c_{RNA,i} \tag{41}$$

where $\tau_{RNA,i}$ is the half life for each RNA. Expected counts of endoRNases and RNA transcripts for an average cell are used to solve the non-linear equation for each $K_{M,i}$.

During simulations, individual RNA counts are low so r_i from Eq. 39 would be very low ($\ll$1) for most RNA. To get integer counts to degrade within a timestep, the total number of RNAs expected to be degraded is first determined. Then, samples are drawn from a multinomial distribution of available RNAs to degrade until this total is reached. This is done separately for each RNA group (mRNA, tRNA and rRNA) based on the known endoRNase affinity for each group:

$$R_{endo,group} = \sum_{i \in group} r_i \cdot \Delta t \tag{42}$$

where r_i is as defined in Eq. 39, Δt is the length of the simulation timestep, and $i \in group$ means that the RNA is in a specific RNA group (mRNA, tRNA or rRNA). The multinomial distribution for each RNA to degrade is sampled with a probability for each RNA species based on the saturated fraction of that species:

$$p_{deg,group,i} = \frac{f_i \cdot \mathbf{1}_{i \in group}}{\sum\limits_{j \in group} f_j} \tag{43}$$

where $\mathbf{1}_{i \in group}$ is an indicator function that is 1 if i in in $group$ and 0 otherwise.

Exo-nucleolytic Digestion. Endo-nucleolytic cleavage produces non-functional RNA fragments, which are then degraded to individual nucleotides via nine different exoRNases. Once degraded, these nucleotides can be recycled by the $\mathrm{Metabolism}$ process. ExoRNase capacity is determined as the following:

$$R_{exo} = k_{cat,exo} \cdot c_{exo} \cdot \Delta t \tag{44}$$

where R_{exo} is the exoRNase digestion capacity, $k_{cat,exo}$ is the rate of cleavage for a single exoRNase, c_{exo} is the total count of exoRNases, and Δt is the length of the simulation timestep. Since non-functional RNA fragments are modeled in aggregate and not individually like functional RNA molecules, we do not include a K_M term for the digestion capacity and assume full saturation.

Algorithm 10: RNA degradation: endo-nucelolytic cleavage and exo-nucleolytic digestion

```
Input : $K_{M,i}$ Michaelis constants of each mRNA transcript binding to endoRNases
        where $i = 1$ to $n_{RNA}$
Input : $k_{cat,endo}$, $k_{cat,exo}$ catalytic rate of endoRNase and exoRNase
Input : $c_{endo}$, $c_{exo}$ count of endoRNases and exoRNases
Input : $c_{frag,i}$ count of nucleotides in non-functional RNA fragments where $i = 1$ to $4$
        for AMP, CMP, GMP, UMP
Input : $c_{nt,i}$ count of free nucleotides where $i = 1$ to $4$ for AMP, CMP, GMP, UMP
Input : $c_{mRNA}$, $c_{tRNA}$, $c_{rRNA}$ count of each mRNA, tRNA and rRNA
Input : $c_{H_2O}$, $c_{PPi}$, $c_{proton}$ count of small molecules
Input : $\mathrm{multinomial}()$ function that draws samples from a multinomial distribution
Input : $\mathrm{countNTs}()$ function that returns counts of AMP, CMP, GMP, and UMP for a
        given non-functional RNA fragment
Input : $\mathrm{lengthFragments}()$ function that returns the total number of bases of all RNA
        fragments
/* Endo-nucleolytic cleveage */
1. Calculate fraction of active endoRNases ($f_i$) that target each RNA where $i = 1$ to
   $n_{gene}$
      $f_i = \frac{\frac{c_{RNA,i}}{K_{M,i}}}{1+\sum_j \frac{c_{RNA,j}}{K_{M,j}}}$
2. Calculate total counts of RNAs to be degraded ($R_{endo,group}$)
      $R_{endo,mRNA} = \sum_{i \in mRNA} k_{cat,endo} \cdot c_{endo,i} \cdot f_i \cdot \Delta t$
      $R_{endo,tRNA} = \sum_{i \in tRNA} k_{cat,endo} \cdot c_{endo,i} \cdot f_i \cdot \Delta t$
      $R_{endo,rRNA} = \sum_{i \in rRNA} k_{cat,endo} \cdot c_{endo,i} \cdot f_i \cdot \Delta t$
3. Determine probabilities for multinomial distributions of RNAs to degrade for each
   RNA group ($p_{deg,group,i}$)
      $p_{deg,mRNA,i} = \frac{f_i \cdot \mathbf{1}_{i \in mRNA}}{\sum_{j \in mRNA} f_j}$
      $p_{deg,tRNA,i} = \frac{f_i \cdot \mathbf{1}_{i \in tRNA}}{\sum_{j \in tRNA} f_j}$
      $p_{deg,rRNA,i} = \frac{f_i \cdot \mathbf{1}_{i \in rRNA}}{\sum_{j \in rRNA} f_j}$
4. Sample multinomial distributions for each group $R_{endo,group}$ times, with probability
   determined by relative endoRNase saturation, to determine counts of RNAs that are
   converted into non-functional RNAs ($d_i$)
      $d_i = \mathrm{multinomial}(R_{endo,mRNA}, p_{deg,mRNA,i})$
          + $\mathrm{multinomial}(R_{endo,tRNA}, p_{deg,tRNA,i})$
          + $\mathrm{multinomial}(R_{endo,rRNA}, p_{deg,rRNA,i})$
```

```
5. Increase number of RNA fragments. Decrease RNA and water counts by amount
  required for RNA hydrolysis and increase pyrophosphate counts for the removal of
  the 5' pyrophosphate
      $c_{frag} = c_{frag} + \text{countNTs}(d_i)$
      $c_{RNA,i} = c_{RNA,i} - d_i$
      $c_{H_2O} = c_{H_2O} - \sum_i d_i$
      $c_{PPi} = c_{PPi} + \sum_i d_i$

/* Exo-nucleolytic digestion */
6. Compute exoRNase capacity ($R_{exo}$)
      $R_{exo} = k_{cat,exo} \cdot c_{exo} \cdot \Delta t$
if $R_{exo} > \sum c_{frag,i}$ then
    Update nucleotide, water and proton counts
        $c_{nt,i} = c_{nt,i} + c_{frag,i}$
        $c_{H_2O} = c_{H_2O} - \text{lengthFragments}(c_{frag})$
        $c_{proton} = c_{proton} + \text{lengthFragments}(c_{frag})$
    Set counts of RNA fragments equal to zero ($c_{frag,i} = 0$)
else
    Sample multinomial distribution $c_{frag}$ with equal probability to determine which
      fragments are exo-digested ($c_{frag,dig}$) and recycled
        $c_{frag,dig,i} = \text{multinomial}(R_{exo}, \frac{c_{frag,i}}{\sum_i c_{frag,i}})$
    Update nucleotide, water, proton counts, and RNA fragments
        $c_{nt,i} = c_{nt,i} + c_{frag,dig,i}$
        $c_{H_2O} = c_{H_2O} - \text{lengthFragments}(c_{frag,dig})$
        $c_{proton} = c_{proton} + \text{lengthFragments}(c_{frag,dig})$
        $c_{frag,i} = c_{frag,i} - c_{frag,dig,i}$
Result: RNAs are selected and degraded by endoRNases, and non-functional RNA
  fragments are digested through exoRNases. During the process water is consumed,
  and nucleotides, pyrophosphate and protons are released.
```

Associated data

Parameter	Symbol	Units	Value	Reference
EndoRNase catalytic rate	$k_{cat,endo}$	RNA counts/s	0.10	See GitHub
ExoRNase catalytic rate	$k_{cat,exo}$	nt digested/s	50	See GitHub
mRNA half-lives$^{(1)}$	τ_{mRNA}	min	[1.30, 31.40]	(*8*)
tRNA, rRNA half-lives	τ_{tRNA}, τ_{rRNA}	hour	48	(*8*)
Michaelis constant	K_M	RNA counts	-	See GitHub
RNAse mechanism of action	-	-	endo-/exo-RNAse	See GitHub
EndoRNase specificity$^{(2)}$	-	-	Specificity for mRNA, tRNA, rRNA	See GitHub

Table 6: Table of parameters for RNA degradation process.
$^{(1)}$Non-measured mRNA half-lives were estimated as the mean mRNA half-life (5.75 min).
$^{(2)}$Matrix relating each endoRNase to the type(s) of RNA that it targets.

Associated files

wcEcoli Path	File	Type
wcEcoli/models/ecoli/processes	rna_degradation.py	process
wcEcoli/reconstruction/ecoli/dataclasses/process	rna_decay.py	data
wcEcoli/reconstruction/ecoli/flat	rnas.tsv	raw data
wcEcoli/reconstruction/ecoli/flat	endoRnases.tsv	raw data

Table 7: Table of files for RNA degradation.

1.1.6.4 Translation

Model implementation. Translation is the process by which the coding sequences of mRNA transcripts are translated by 70S ribosomes into polypeptides that then fold into proteins. This process accounts for more than two thirds of an *E. coli* cell's ATP consumption during rapid growth (*9*) and the majority of macromolecular mass accumulation. In the *E. coli* model translation occurs through the action of two processes in the model: PolypeptideInitiation and PolypeptideElongation.

Initiation. $\mathrm{PolypeptideInitiation}$ models the complementation of 30S and 50S ribosomal subunits into 70S ribosomes on mRNA transcripts. This process is in many ways analogous to the $\mathrm{TranscriptInitiation}$ process - the number of initiation events per transcript is determined in a probabilistic manner and dependent on the number of free ribosomal subunits, each mRNA transcript's translation efficiency, and the counts of each type of transcript. The total number of ribosomes to activate in each time step is determined such that the average fraction of actively translating ribosomes matches experimental values. This is done by assuming a steady state concentration of active ribosomes (and therefore a constant active fraction), similar to what was done for RNA polymerases in $\mathrm{TranscriptInitiation}$:

$$\frac{dR_{70S}}{dt} = p_{act} \cdot \min(R_{30S}, R_{50S}) - r \cdot R_{70S} = 0 \tag{45}$$

$$p_{act} = \frac{r \cdot R_{70S}}{\min(R_{30S}, R_{50S})} \tag{46}$$

where R_{70S} is the concentration of active 70S ribosomes, R_{50S} and R_{30S} are the concentrations of free 50S and 30S ribosomal subunits, respectively, p_{act} is the activation probability and r is the expected termination rate for active 70S ribosomes. Defining the active fraction as $f_{act} = \frac{R_{70S}}{R_{70S}+\min(R_{30S},R_{50S})}$, p_{act} can be defined in terms of the desired active fraction:

$$p_{act} = \frac{r \cdot f_{act}}{1 - f_{act}} \tag{47}$$

This activation probability is then used to determine how many 70S ribosomes will be formed and initiated. These newly initiated 70S ribosomes are distributed to mRNA transcripts based on their translation probabilities, which is computed by normalizing the product of the translational efficiency of each transcript and the counts of each transcript.

$$p_i = \frac{c_{mRNA,i} \cdot t_i}{\sum\limits_{j=1}^{n_{gene}} c_{mRNA,j} \cdot t_j} \tag{48}$$

Here, p_i is the translation probability of the transcript of gene i, $c_{mRNA,i}$ is the count of the transcript of gene i, and t_i is the translation efficiency of the transcript of gene i. The translational efficiencies of each transcript were calculated from ribosomal profiling data (*10*). For transcripts whose translation efficiencies were not given in this dataset, the average of the existing efficiency values was used as the translation efficiency. Full 70S ribosomes are formed on mRNA transcripts by sampling a multinomial distribution with the p_i's calculated above as the probabilistic weights (See Algorithm 11).

This process is implemented similarly in the *M. genitalium* model with a few key differences. Unlike the *M. genitalium* model, the *E. coli* model is not yet gene complete, and hence does not check for initiation factors. A major advancement over the *M. genitalium* model is that the probability of ribosome initiation on a transcript is now proportional to the product of the mRNA count and its translational efficiency. In the *M. genitalium* model translational efficiencies were not taken into account.

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

Under our simulation conditions, we empirically observe that the rate of translation elongation is always limited by the supply of amino acids, not by the elongation rates of ribosomes or the supply of GTP. Thus, the rate at which amino acids are supplied to translation largely determines the synthesis rates of proteins, which in turn is highly correlated with the growth rate of the cell. We have therefore added an option to add Gaussian noise to this supply rate of amino acids to translation, in cases where heterogeneity in growth rates between individual simulations would be a desired outcome.

Unlike the *M. genitalium* model, this process in the *E. coli* model does not account for elongation factors. Additionally, tRNAs and their synthetases are not accounted for explicitly. Instead, the model directly polymerizes amino acids into elongating polypeptides. This avoids computational issues with the simulation time step, tRNA pool size, and tRNA overexpression that were present in the *M. genitalium* model. There is no implementation of ribosome stalling or tmRNAs at this point. The polymerization resource allocation algorithm is the same as in the *M. genitalium* model.

```
Algorithm 11: Ribosome initiation on mRNA transcripts
  Input : $f_{act}$ fraction of ribosomes that are active
  Input : $r$ expected termination rate for active ribosomes
  Input : $t_i$ translational efficiency of each mRNA transcript where $i = 1$ to $n_{gene}$
  Input : $c_{mRNA,i}$ count of each mRNA transcript where $i = 1$ to $n_{gene}$
  Input : $c_{30S}$ count of free 30S ribosomal subunit
  Input : $c_{50S}$ count of free 50S ribosome subunit
  Input : $\text{multinomial}()$ function that draws samples from a multinomial distribution
  1. Calculate probability ($p_{act}$) of free ribosomal subunits binding to a transcript.
      $p_{act} = \frac{r \cdot f_{act}}{1 - f_{act}}$
  2. Calculate the number of 70S ribosomes that will be formed and initiated ($c_{70S,b}$).
      $c_{70S,b} = p_{act} \cdot \min(c_{30S}, c_{50S})$
  3. Calculate probability ($p_i$) of forming a ribosome on each mRNA transcript weighted
     by the count and translational efficiency of the transcript.
      $p_i = \frac{c_{mRNA,i} \cdot t_i}{\sum_{j=1}^{n_{gene}} c_{mRNA,j} \cdot t_j}$
  4. Sample multinomial distribution $c_{70S,b}$ times weighted by $p_i$ to determine which
     transcripts receive a ribosome and initiate ($n_{init,i}$).
      $n_{init,i} = \text{multinomial}(c_{70S,b}, p_i)$
  5. Assign $n_{init,i}$ ribosomes to mRNA transcript $i$. Decrement 30S and 50S counts.
      $c_{30S} = c_{30S} - \sum_{i=1}^{n_{gene}} n_{init,i}$
      $c_{50S} = c_{50S} - \sum_{i=1}^{n_{gene}} n_{init,i}$
  Result: 70S ribosomes are formed from free 30S and 50S subunits on mRNA
          transcripts scaled by the count of the mRNA transcript and the transcript's
          translational efficiency.
```

Algorithm 12: Peptide chain elongation and termination

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

Associated data

Parameter	Symbol	Units	Value	Reference
Active fraction of ribosomes	f_{act}	-	0.8	(*7*)
Translational efficiency[(1)]	t_i	ribosomes/mRNA	[0, 5.11]	(*10*)
Ribosome elongation rate	e	aa/s	18 (growth-dependent)	(*7*)
Protein counts (validation data)	$c_{protein}$	protein counts	[0, 250000]	(*11*)

Table 8: Table of parameters for translation process.
[(1)] Non-measured translational efficiencies were estimated to be equal to the average translational efficiency (1.11 ribosomes/mRNA).

Associated files

wcEcoli Path	File	Type
wcEcoli/models/ecoli/processes	polypeptide_initiation.py	process
wcEcoli/models/ecoli/processes	polypeptide_elongation.py	process
wcEcoli/reconstruction/ecoli/dataclasses/process	translation.py	data
wcEcoli/reconstruction/ecoli/flat	proteins.tsv	raw data
wcEcoli/reconstruction/ecoli/flat	translationEfficiency.tsv	raw data
wcEcoli/validation/ecoli/flat	schmidt2015_javier_table.tsv	validation data

Table 9: Table of files for translation.

1.1.6.5 Protein degradation

Model Implementation. The ProteinDegradation process accounts for the degradation of protein monomers. It uses the N-end rule (*12*) to assign half-lives for each protein monomer with half-lives for select protein determined experimentally as noted in the main text. Specific proteins to be degraded are selected as a Poisson process. The *E. coli* model is not yet gene complete, hence this process does not take into account the activities of specific proteases and does not specifically target prematurely aborted polypeptides. In addition, protein unfolding and refolding by chaperones is not accounted for by this process. The exclusion of these effects is a difference from the *M. genitalium* model and may be the reason for the discrepancies in protein half-lives observed in Figure 5.

```
Algorithm 13: Protein degradation
    Input : $t_{1/2,i}$ Protein half-lives for each monomer where $i = 1$ to $n_{protein}$
    Input : $L_i$ length of each protein monomer where $i = 1$ to $n_{protein}$
    Input : $c_{aa,i,j}$ number of each amino acid present in the protein monomer where $i = 1$
            to $n_{protein}$ and $j = 1$ to $21$ for each amino acid
    Input : $c_{protein,i}$ the number of each protein present in the cell
    1. Determine how many proteins to degrade based on the degradation rates and counts
       of each protein.
            $n_{protein,i} = \text{poisson}(\frac{ln(2)}{t_{1/2,i}} \cdot c_{protein,i} \cdot \Delta t)$
    2. Determine the number of hydrolysis reactions ($n_{rxns}$) that will need to occur.
            $n_{rxns} = \sum_i (L_i - 1) \cdot n_{proteins,i}$
    3. Determine the number of amino acids ($n_{aa,j}$) that will be released.
            $n_{aa,j} = \sum_i c_{aa,i,j} \cdot n_{proteins,i}$
    4. Degrade selected proteins, release amino acids from those proteins back into the cell,
       and consume $H_2O$ according to the number required for the hydrolysis reactions.
    Result: Proteins are selected and degraded. During the process water is consumed, and
       amino acids are released.
```

Associated data

Parameter	Symbol	Units	Value	Reference
N-end rule protein half-lives	$t_{1/2}$	min	2 or 600	(*12*)
Measured protein half-lives	$t_{1/2}$	hr	[0.6, 39.7]	This study

Table 10: Table of parameters for protein degradation process. The measured protein half-lives are for DcuR, BioD, Rph, CarA, Pnp, GshA, and CdsA. Note: CdsA was unable to be measured exactly but was observed to be longer than 2 minutes (which is its expected value according to the N-end rule) so was assigned 10 hours.

Associated files

wcEcoli Path	File	Type
wcEcoli/models/ecoli/processes	protein_degradation.py	process
wcEcoli/reconstruction/ecoli/dataclasses/process	translation.py	data
wcEcoli/reconstruction/ecoli/flat	protein_half_lives.tsv	raw data

Table 11: Table of files for protein degradation.

1.1.6.6 Complexation

Model implementation. This process models the formation of all macromolecular complexes except for the formation of 70S ribosomes from 30S and 50S subunits, which is performed by $\mathrm{Translation}$. Macromolecular complexation is done by identifying complexation reactions that are possible (which are reactions that have sufficient counts of all sub-components), performing one randomly chosen possible reaction, and re-identifying all possible complexation reactions. This process assumes that macromolecular complexes form spontaneously, and that complexation reactions are fast and complete within the time step of the simulation. This approach is very similar to the *M. genitalium* model of complexation with the exception that the selection of a complexation reaction was weighted by a multinomial distribution parameterized by substrate availability rather than a uniform distribution. We found that the choice of distribution had no major effect on behavior of the process. Additionally, the *M. genitalium* simulations describe 201 macromolecular complexes, whereas over 5 times as many are implemented in the *E. coli* model.

Algorithm 14: Macromolecular complexation

```
Input : $c_i$ counts of molecules where $i = 1$ to $n_{molecules}$
Input : $S$ matrix describing reaction stoichiometries where $S_{i,j}$ describes the
        coefficient for the $i^{th}$ molecule in the $j^{th}$ reaction
Input : getPossibleReactions function that takes $c_i$ and $S$ and returns all reactions that
        are possible
Input : chooseRandomReaction function that takes all possible reactions and returns
        one randomly chosen reaction
while possible reactions remaining do
    1. Get all possible reactions ($r$)
           $r = \mathrm{getPossibleReactions}(S, c_i)$
    2. Choose a random possible reaction ($r_{choice}$) to perform
           $r_{choice} = \mathrm{chooseRandomReaction}(r)$
    3. Perform $r_{choice}$ by incrementing product counts and decrementing reactant counts
Result: Macromolecule complexes are formed from their subunits according to their
        known stoichiometries.
```

Associated data

Stoichiometric coefficients that define 1,023 complexation reactions from EcoCyc (*13*).

Associated files

wcEcoli Path	File	Type
wcEcoli/models/ecoli/processes	complexation.py	process
wcEcoli/reconstruction/ecoli/dataclasses/process	complexation.py	data
wcEcoli/reconstruction/ecoli/flat	complexationReactions.tsv	raw data

Table 12: Table of files for complexation.

1.1.6.7 Metabolism

Model implementation. Our challenge in modeling metabolism at the large scale is best reflected in the two major approaches to modeling metabolic networks. Both approaches begin by writing a number of ordinary differential equations to reflect conservation of mass for each of the metabolites in the system. In a kinetics-based approach, the terms of the ODEs are represented in terms of enzyme and small molecule concentrations. The difficulty in implementing this approach at a large scale is that (1) a majority of the parameters are not known, and (2) the link between the parameter values - each measured with their own error - is rigid in this approach, such that it is generally impossible to build a model even at a smaller scale without abandoning most of the experimental data and performing extensive model fitting.

The second approach is called flux balance analysis (FBA), and is a common way to model large-scale metabolic network behavior with a low parameter requirement. However, traditional implementations of FBA are inappropriate for whole-cell modeling due to the dynamic nature of whole-cell simulation and fixed nature of the classic FBA objective function. Moreover, in cases such as *E. coli*, where many parameter values have been measured, it would be unfortunate not to use these measurements.

The method we describe here incorporates the best parts of both kinetics- and flux-balance-based modeling. This enables the model to run stably at a large scale, incorporating known parameter values without additional fitting, and in a manner that is readily integrable with the rest of a whole-cell modeling approach. This is done by using an alternative objective function that involves a multi-objective minimization for homeostatic metabolite composition and reaction kinetics that extends previous work by Birch *et al.* (*14*). The effect of this multi-objective function is twofold: (1) to maintain cellular concentrations of small molecule metabolites and (2) to enforce constraints on metabolic fluxes calculated from Michaelis-Menten kinetics based on metabolite concentrations and curated kinetic parameters. A weighting factor is used to balance the contribution from the two objectives.

We used the metabolic network reconstruction from Weaver *et al.* (*15*) because it was well-connected to the rest of EcoCyc's resources and data which we relied on. This network reconstruction was based on the Orth model (*16*). Different nutrient conditions (minimal M9, +amino acids, -oxygen, etc.) can be specified by changing bounds on metabolite import reactions, and shifts between these nutrient conditions can be programmatically varied.

Homeostatic objective. The homeostatic objective attempts to maintain small molecule metabolite concentrations at a constant value. For example, if during a time-step the net effect of other $\mathrm{Process}$ execution transforms ATP to ADP, the concentration of ATP will be lower and ADP higher. The homeostatic objective ensures that the metabolic network will attempt to increase the ATP concentration and decrease the ADP concentration using chemical transformations available in the network.

A total of 140 metabolite set-point concentrations are specified in the objective ($C_{o,i}$ in Equation 49). A list of those metabolites with concentrations in the model is given in Table S7. The homeostatic objective minimizes the deviation from these measured concentrations and can be specified as:

$$\text{minimize} \sum_i \left| 1 - \frac{C_i}{C_{o,i}} \right| \tag{49}$$

where C_i is the concentration of metabolite i and $C_{o,i}$ is the measured set-point concentration for metabolite i. Cytoplasmic concentrations were chosen based on data from Bennett *et al.* (*17*), and other components of biomass have set-point concentrations specified based on the overall composition of the cell (lipids, metal ions, etc.) (*18*) and can be dependent on the media environment of the simulation.

This approach is an improvement over the *M. genitalium* simulations, where metabolites were produced in a fixed ratio at every time step regardless of the behavior of the rest of the simulated cell—this could lead to pooling or depletion of metabolites. Furthermore, if one metabolite could not be produced, none of the metabolites could be produced. Our homeostatic objective fixes both of these shortcomings.

Kinetics objective. The *E. coli* model simulates both metabolic enzyme expression via transcription and translation and dynamically maintains 140 metabolite concentrations. This enables the use of Michaelis-Menten kinetic equality constraints on metabolic fluxes using Equation 50:

$$v_{o,j} = k_{cat} \cdot E \cdot \left(\frac{C_1}{C_1 + K_{m,1}} \right) \cdot \left(\frac{C_2}{C_2 + K_{m,2}} \right) \cdots \left(\frac{C_n}{C_n + K_{m,n}} \right) \tag{50}$$

where $v_{o,j}$ is the kinetic target for the flux through reaction j that has n substrates, k_{cat} is the catalytic turnover rate for enzyme E, $K_{m,n}$ is the saturation constant for substrate n, E is the concentration of metabolic enzyme, and C_n is the concentration of substrate n in reaction j.

Kinetics data was reviewed from over 12,000 papers identified from BRENDA (*19*). We filtered out papers that did not have a k_{cat} and which did not use a lab strain, or which concerned non-metabolic enzymes. The result was roughly 1200 papers which we manually curated due to our and others' observation that about 20% of the values in the BRENDA database are copied incorrectly from their primary source papers (*20*). From this set, we further removed enzymes that were not included in our metabolic model and consolidated the results of multiple studies (usually preferring experiments conducted at 37°C and/or the report with the highest k_{cat} value). In all, 179 constraints with a K_M and k_{cat} and 216 constraints with only a k_{cat} are used to constrain a total of 380 reactions (with some reactions having multiple constraints). Although some additional constraints were identified, they are not currently being used in the model. In particular, constraints were found for tRNA charging (18 reactions) but not used since tRNA charging is not explicitly included in the model. Additionally, as discussed in the main text, constraints for four reactions (succinate dehydrogenase, NADH dehydrogenase, inorganic pyrophosphatase, and glutathione reductase) were identified that, when included, caused a much higher glucose uptake rate than observed without kinetic constraints and higher than what has been experimentally measured. In related analysis also outlined in the main text, two more reactions (isocitrate dehydrogenase and glyoxylate reductase) were found to be under additional control that is not modeled so their constraints were excluded. Based on this, constraints related to these 24 reactions were excluded from the model. A complete list of the 431 reactions with curated kinetic parameters, including those mentioned above that were excluded from the model, is found in Table S8.

In cases where the enzyme parameters were recorded at non-physiological temperatures, we used the following scaling relationship to adjust the k_{cat}:

$$k_{cat,adj} = 2^{\frac{37-T}{10}} \cdot k_{cat} \tag{51}$$

where T is the reported temperature (in °C) for the experimental conditions—this increases the kinetic rate by a factor of 2 for every 10 °C below 37 °C. The factor 2 comes from an approximation of the change in rates determined by the Arrhenius equation:

$$k = A \cdot e^{-\frac{E_a}{R \cdot T}} \tag{52}$$

where k is the calculated rate, A is a constant for a given reaction, E_a is the activation energy for a given reaction, R is the universal gas constant and T is the temperature. Assuming $E_a \approx$ 50,000 $\frac{J}{mol}$ and the temperature increases 10 °C near 25 °C ($T_1 = 293$ K to $T_2 = 303$ K), we get an approximate doubling in the rate for the 10 °C increase:

$$\frac{k_2}{k_1} = \frac{A \cdot e^{-\frac{E_a}{R \cdot T_2}}}{A \cdot e^{-\frac{E_a}{R \cdot T_1}}} = \frac{e^{-\frac{50000}{8.314 \cdot 303}}}{e^{-\frac{50000}{8.314 \cdot 293}}} \approx 2.0 \tag{53}$$

Similar to the homeostatic objective, the kinetics objective minimizes deviation from the fluxes determined by the kinetic equations that are calculated at each time step based on the enzyme and metabolite concentrations. Formally:

$$\text{minimize} \sum_j \left| 1 - \frac{v_j}{v_{o,j}} \right| \tag{54}$$

where v_j is the flux through reaction j and $v_{o,j}$ is the target flux for reaction j calculated from Equation 50. This represents a soft kinetic constraint on reactions for which we have kinetic parameters and depends on concentrations of enzymes and metabolites in the model.

Combined objective. The two objectives above are linked in the combined objective by a factor λ, which is chosen to be $\ll$1 to prioritize the homeostatic concentration objective function's ability to produce growth at known doubling times, over fit the kinetic data. λ is not a biological parameter such as those found in Table S1 (it is better thought of as a “hyperparameter”), and thus it is not included in that table.

Finally, a hard kinetic constraint of no flux through a reaction is implemented for reactions that have no enzyme present by setting $v_{max} = 0$.

This results in the following linear optimization problem that is solved at each time step with GLPK to determine fluxes for each reaction which are used to update the counts of metabolites through exchange reaction fluxes included in the problem:

$$\begin{array}{ll}\text{minimize} & (1-\lambda)\sum_i \left|1-\frac{C_i}{C_{o,i}}\right| + \lambda \sum_j \left|1-\frac{v_j}{v_{o,j}}\right| \\ \text{subject to} & S \cdot v = 0 \\ & v_j \geq v_{min,j} \\ & v_j \leq v_{max,j}\end{array}$$

```
Algorithm 15: Metabolism
  Input : $C_i$ concentration for metabolite $i$
  Input : $C_{o,i}$ concentration target for metabolite $i$
  Input : $k_{cat,j}$ turnover number for enzyme $j$
  Input : $K_{m,i,j}$ Michaelis constant for metabolite $i$ for enzyme $j$
  Input : $E_j$ concentration for enzyme $j$
  Input : $S$ stoichiometric matrix for all reactions
  1. Set physical constraints on reaction fluxes
        For all reactions: $v_{min,j} = -\inf, v_{max,j} = +\inf$
        For thermodynamically irreversible reactions: $v_{min,j} = 0$
        If required enzyme not present: $v_{min,j} = v_{max,j} = 0$
  2. Calculate kinetic target ($v_{o,j}$) for each reaction $j$ based on the enzymes $j$ and
     metabolites $i$ associated with each reaction for a soft flux constraint
        $v_{o,j} = k_{cat,j} \cdot E_j \cdot \prod_i (\frac{C_i}{K_{m,i,j}+C_i})$
  3. Solve linear optimization problem
        minimize    $(1-\lambda)\sum_i \left|1 - \frac{C_i}{C_{o,i}}\right| + \lambda \sum_j \left|1 - \frac{v_j}{v_{o,j}}\right|$
        subject to  $S \cdot v = 0$
                    $v_j \geq v_{min,j}$
                    $v_j \leq v_{max,j}$
  4. Update concentrations of metabolites based on the solution to the linear optimization
     problem
  Result: Metabolites are taken up from the environment and converted into other
          metabolites for use in other processes
```

Associated data

Parameter	Symbol	Units	Value	Reference
Metabolic network	S	-	Stoichiometric coefficients	(*21*)
Metabolic target fluxes	v_o	$\mu M/s$	[0, 87000]	See GitHub
Metabolic fluxes (validation)	v_v	$\mu M/s$	[82, 1500]	(*22*)
Enzyme turnover number	k_{cat}	$1/s$	[0.00063, 38000]	See GitHub
Enzyme Michaelis constant	K_m	μM	[0.035, 550000]	See GitHub
Metabolite target concentration	C_o	μM	[0.063, 97000]	(*17*)
Kinetic objective weight	λ	-	1e-6	See GitHub

Table 13: Table of parameters for metabolism process.

Associated files

wcEcoli Path	File	Type
wcEcoli/models/ecoli/processes	metabolism.py	process
wcEcoli/reconstruction/ecoli/dataclasses/process	metabolism.py	data
wcEcoli/reconstruction/ecoli/flat	biomass.tsv	raw data
wcEcoli/reconstruction/ecoli/flat	enzymeKinetics.tsv	raw data
wcEcoli/reconstruction/ecoli/flat	metaboliteConcentrations.tsv	raw data
wcEcoli/reconstruction/ecoli/flat	reactions.tsv	raw data
wcEcoli/validation/ecoli/flat	toya_2010_central_carbon_fluxes.tsv	validation data

Table 14: Table of files for metabolism.

1.1.6.8 Energy requirements of cell maintenance

As was the case in our *M. genitalium* simulations, and in many flux balance analysis models, not all of the energy consumed by metabolic pathways, macromolecular polymerization, or other growth and non-growth associated processes is accounted for explicitly in our *E. coli* model. This is primarily due to a lack of experimental data and/or knowledge accounting for its usage. Similar to the *M. genitalium* model, we have incorporated reactions in the metabolic model with two parameters, Growth Associated Maintenance (GAM) and Non-Growth Associated Maintenance (NGAM), which modulate energy consumption associated with growth and cell maintenance.

Associated data

Parameter	Symbol	Units	Value	Reference
Growth associated maintenance	GAM	mmol ATP/g DCW	59.81	(*23*)
Non-growth associated maintenance	NGAM	mmol ATP/g DCW/h	8.39	(*23*)

Table 15: Table of parameters for energy requirements of cell maintenance.

1.1.6.9 Chromosome replication

Model implementation. Chromosome replication occurs through three steps that are implemented in the ChromosomeFormation and ChromosomeElongation processes. First, a round of replication is initiated at a fixed cell mass per origin of replication and generally occurs once per cell cycle (see Algorithm 16). This is in contrast to the DnaA based mechanistic model included in the *M. genitalium* model but allows for stable replication over multiple generations and in different growth conditions. Second, replication forks are elongated up to the maximal expected elongation rate, dNTP resource limitations, and template strand sequence but elongation does not take into account the action of topoisomerases or the enzymes in the replisome (see Algorithm 17). Finally, replication forks terminate once they reach the end of their template strand and the chromosome immediately decatenates forming two separate chromosome molecules (see Algorithm 18).

Algorithm 16: DNA replication initiation

```
Input : $m_{cell}$ cell mass
Input : $m_{critical}$ critical initiation mass
Input : $n_{origin}$ number of origins of replication
Input : $n_{fork,f}$ number of replication forks on forward strand
Input : $n_{fork,r}$ number of replication forks on reverse strand
Input : $C$ length of C period
Input : $D$ length of D period
if $\frac{m_{cell}}{n_{origin}} > m_{critical}$ then
    $n_{fork,f} = n_{fork,f} + n_{origin}$
    $n_{fork,r} = n_{fork,r} + n_{origin}$
    $n_{origin} = 2 \cdot n_{origin}$
Result: When cell mass is larger than critical initiation mass $m_c$ another round of
        replication is initiated with correct number of replication forks
```

Algorithm 17: DNA replication elongation

```
Input : $e$ maximal elongation rate of replication fork
Input : $p_i$ position of forks on chromosome where $i = 1$ to $n_{fork}$
Input : $\Delta t$ length of current time step
Input : $c_{dNTP,j}$ counts of dNTP where $j = 1$ to $4$ for dCTP, dGTP, dATP, dTTP
Input : $L_k$ total length of each strand of chromosome from origin to terminus where
        $k = 1$ to $4$ for forward/complement and reverse/complement.
for each replication fork $i$ on sequence $k$ do
    1. Based on replication fork position $p_i$ and maximal elongation rate $e$ determine
       "stop condition" ($s_i$) for replication fork assuming no dNTP limitation.
           $s_i = \min(p_i + e \cdot \Delta t,\, L_k)$
    Stop condition is either maximal elongation rate scaled by the time step or the full
       length of sequence (i.e. the fork will terminate in this time step).
    2. Derive sequence between replication fork position ($p_i$) and stop condition ($s_i$).
    3. Based on derived sequence calculate the number of dNTPs required to
       polymerize sequence $c^{req}_{dNTP,i}$.
    4. Elongate up to limits:
    if $\mathrm{all}(c^{req}_{dNTP,i} < c_{dNTP,j})$ then
        Update the position of each replication fork to stop position
            $p_i = s_i$
    else
        Attempt to equally elongate each replication fork update position of each fork
          to maximal position given the limitation of $c_{dNTP,j}$.
    5. Update counts of $c_{dNTP,j}$ to reflect polymerization usage.
Result: Each replication fork is elongated up to the limit of available sequence,
        elongation rate, or dNTP limitation
```

Algorithm 18: DNA replication termination

```
Input : $p_i$ position of forks on chromosome where $i = 1$ to $n_{fork}$
Input : $L_k$ total length of each strand of chromosome from origin to terminus where
        $k = 1$ to $4$ for forward/complement and reverse/complement
Input : $d_{queue}$ a double ended queue data structure that stores time(s) cell division
        should be triggered
Input : $D$ D-period of cell cycle (time between completion of chromosome replication
        and cell division)
Input : $t$ Current simulation time
for each replication fork i on strand k do
    if $p_i == L_k$ then
        1. Delete replication fork
        2. Divide remaining replication forks and origins of replication appropriately
         across the two new chromosome molecules
        3. Calculate time cell should trigger division based on current time of
         chromosome termination and push onto queue data structure
             $d_{queue}.\mathrm{push}(t + D)$
Result: Replication forks that have terminated are removed. A new chromosome
        molecule is created separating all remaining replication forks. Timer for
        D-period is started.
```

Associated data

Parameter	Symbol	Units	Value	Reference
Chromosome sequence	-	-	-	(*24*)
Replication fork elongation rate	e	nt/s	967	(*25*)
Mass per origin at DNA replication initiation[(1)]	$m_{critical}$	origin/fg	[600,975]	(*26*)
C period	C	min	40	(*27*)
D period	D	min	20	(*27*)

Table 16: Table of parameters for chromosome replication process.
[(1)] 600 is used for anaerobic conditions where the cell mass is lower and was fit to achieve the appropriate D period. All other growth conditions use 975.

Associated files

wcEcoli Path	File	Type
wcEcoli/models/ecoli/processes	chromosome_formation.py	process
wcEcoli/models/ecoli/processes	chromosome_replication.py	process
wcEcoli/reconstruction/ecoli/dataclasses/process	replication.py	data
wcEcoli/reconstruction/ecoli/flat	genes.tsv	raw data
wcEcoli/reconstruction/ecoli/flat	sequence.fasta	raw data

Table 17: Table of files for chromosome replication.

1.1.6.10 Cell division

Model implementation. Cell division is modeled by the generalized divide_cell function and the *E. coli*-specific ChromosomeReplication process and CellDivision listener in the model. A Helmstetter-Cooper type model of chromosome replication initiation is coupled to cell division, inspired by work from Wallden *et al.* (*28*). Chromosome replication initiation occurs at a fixed mass per origin of replication. Each initiation event is coupled to a cell division event after a constant period of time consisting of one round of chromosome replication (C period) and cytokinesis (D period). When a round of chromosome replication is completed, the ChromosomeReplication process adds the length of the D period to the current time and pushes this time to a queue. The CellDivision listener checks this queue and the current time at every timestep, and triggers cell division when the current time passes the earliest time in this queue.

Cell division itself done by the divide_cell function and is modeled as a binomial process where each daughter cell has an equal probability of inheriting the contents of the mother cell. The exception to this is if two chromosomes are present before cell division—each daughter is guaranteed to get one. Because the *E. coli* model is not yet gene complete, certain mechanistic details of cell division (eg. cytokinesis, septation, and chromosome segregation) are not yet modeled explicitly.

Due to the coupled nature of chromosome replication and this division model as mentioned above, Algorithm 18 (previous section) is used to create the doubled ended queue referenced in the cell division implementation provided in Algorithm 19 below.

Algorithm 19: Cell division

```
Input : $d_{queue}$ a double ended queue data structure that stores the time(s) cell division
        should be triggered
Input : $c_i$ counts of all molecules in simulation at cell division where $i = 1$ to $n_{species}$
Input : $p$ binomial partition coefficient
Input : $n_{chrom}$ number of chromosome molecules
Input : $\mathrm{rand}()$ returns a random number from a uniform distribution between 0 and 1
Input : $\mathrm{randint}()$ returns a random integer either 0 or 1
if $t > d_{queue}.\mathrm{peek}()$ then
    1. Trigger division and remove division time.
          $d_{queue}.\mathrm{pop}()$
    2. Divide bulk contents of cell binomially. Number partitioned into daughter one is
      stored in $n_{daughter,1}$ and to daughter two in $n_{daughter,2}$.
    for $i = 1$ to $n_{species}$ do
        $n_{daughter,1} = 0$
        for $j = 1$ to $c_i$ do
            if $\mathrm{rand}() > p$ then
                $n_{daughter,1} = n_{daughter,1} + 1$
        $n_{daughter,2} = c_i - n_{daughter,1}$
    3. Divide chromosome in binary manner. All replication forks and origins of
      replication associated with a chromosome molecule are partitioned as well.
      Number of chromosome molecules partitioned into daughter one is stored in
      $n_{chrom,daughter,1}$ and to daughter two in $n_{chrom,daughter,2}$.
    if $\mathrm{mod}(n_{chrom}, 2)$ then
        $n_{chrom,daughter,1} = \frac{n_{chrom}}{2}$
    else
        $n_{chrom,daughter,1} = \mathrm{floor}(\frac{n_{chrom}}{2}) + \mathrm{randint}()$
    $n_{chrom,daughter,2} = n_{chrom} - n_{chrom,daughter,1}$
Result: Cell division is triggered at C+D time after DNA replication initiation.
        Contents of mother cell is divided between two daughter cells conserving mass.
```

Associated files

wcEcoli Path	File	Type
wcEcoli/models/ecoli/processes	chromosome_replication.py	process
wcEcoli/wholecell/sim	divide_cell.py	function
wcEcoli/models/ecoli/listeners	cell_division.py	listener

Table 18: Table of files for transcription regulation.

1.2 Simulation analysis methods

This section describes how the simulations were used to perform the analysis described in the main text. For convenience, we have organized this section into subsections corresponding to each figure in the main text, together with its related figure in the supplemental materials. Moreover, our GitHub repository (https://github.com/CovertLab/WholeCellEcoliRelease) contains a file titled runscripts/paper/paper_runs.sh which includes all of the commands to run the simulations and a file titled runscripts/paper/paper_figures.sh which includes all of the commands required to generate our figures from the simulation data, for ease of reproduction. Parameter values used for the simulations can be found in the reconstruction/ecoli/flat directory, and the literature sources used for Movie S1 can be found at paper/movie_data_sources.tsv.

1.2.1 Multiple-generation simulations of the cellular response to an environmental shift (related to Fig. 1 and Fig. S1)

For environmental shifts such as the one shown in Fig. 1 and Fig. S1, we initialized our model using the aerobic glucose minimal media environment, and then at time 11,000 seconds (labeled "Environmental Shift"), we changed the environmental conditions to aerobic glucose minimal media supplemented with twenty amino acids (represented by the gray regions). Each environment was set as described above in the "Environments" sub-section. As mentioned in the main text, this version of the model includes changes to the parameter values related to ribosome and RNA polymerase expression (see Section 1.2.2), which differ depending on the media conditions (which is incorporated at cell initiation described in Section 1.1.5), as well as the parameter values for kinetics and gene expression of certain metabolic enzymes (described in Section 1.2.3).

For the "Integrated Simulation Output" of Fig. 1 and Fig. S1, eight simulations of a single cell growing through nine generations were run, each with its own random seed. At each division, one of the daughters was randomly chosen to continue the simulation, meaning that nine division cycles were captured per simulation. Each simulation required a total of 22,000 seconds of simulation, depending on when division actually occurred for each generation. All eight simulations were inspected and found to be qualitatively similar; only one is used for the panels in Fig. 1 and Fig. S1. For this simulation, the final division event occurred at 21,242 seconds.

Below we list the properties graphed in Fig. 1 and Fig. S1, together with a brief description of how this property was obtained or calculated:

Cell mass. Cell mass (blue) was directly extracted from the simulation outputs, and plotted over the cells' time. Red circles indicate the timepoints when chromosome division is initialized, which is determined under the condition:

$$\frac{m_{cell}}{n_{origin}} > m_{critical} \tag{55}$$

where m_{cell} is the simulated mass of the cell at a given time point, n_{origin} is the current number of origins of replication in the cell, and $m_{critical}$ is the critical mass of initiation - a parameter in the model. (For more details, consult Section 1.1.6.9)

Instantaneous Growth Rate. The growth rate was directly extracted from the simulation outputs, converted from 1/sec to 1/min, and plotted over time.

Active Ribosome Concentration. The concentration of active ribosomes per time step was determined from counts in the simulation as follows:

$$\text{Active Ribosome Concentration} = \frac{(\frac{1}{\text{Avogadro's Number}}) \cdot \text{Active Ribosome Count}}{\frac{1}{\text{Cell Density}} \cdot \text{Cell Mass}} \tag{56}$$

where the Active Ribosome Count, cell density and cell mass were extracted directly from the simulation outputs.

DNA Polymerase Position. As a replication fork is made, the model monitors the position of the DNA polymerase as it moves in both directions on the chromosome both on the leading and lagging strands. For the plot in Fig. 1 and S1A, the DNA polymerase position was down-sampled so that the position is plotted once every 10 time steps.

Relative Rate of dNTP Polymerization. This plot displays the number of pairs of replication forks over time. This value is calculated by dividing the number of replicating strands (which is computed from the DNA polymerase position) by four. As shown in the inset diagram in Fig. S1A, a value of 1 indicates a single chromosome is replicating once (there are 4 strands total in this situation, a leading and lagging strand for both the forward and reverse forks).

TrpR promoter occupancy; TrpA mRNA counts, translation events and protein counts; internal Trp concentration. To illustrate the change in gene expression due to the simulated environmental shift, we show several aspects of transcriptional regulation and its downstream effects for the case of tryptophan biosynthesis and regulation of TrpA expression. First, the promoter occupancy of transcriptional inhibitor TrpR is extracted from the simulation and shown as a moving average over time. The number of TrpA mRNA, the number of completed TrpA translation events, and the number of TrpA proteins (including both the monomer and complex) were directly extracted from the simulation outputs at every time step. The internal tryptophan concentration was obtained as a number of cytoplasmic molecules, and converted to a concentration using this equation:

$$\text{Internal Tryptophan Concentration} = \frac{(\frac{1}{\text{Avogadro's Number}}) \cdot \text{Internal Tryptophan Count}}{\frac{1}{\text{Cell Density}} \cdot \text{Cell Mass}} \tag{57}$$

Average Instantaneous Growth Rate. For each time step, the masses of the cell, protein and RNA were extracted. The instantaneous growth rate for each mass fraction was calculated as follows, over each time step:

$$\text{Instantaneous Growth Rate} = \frac{\frac{\Delta m}{dt}}{m_{initial}} \tag{58}$$

where Δm refers to the change in mass across a time step, dt is the length of the time step and $m_{initial}$ refers to the mass at the start of the time step. A moving average was computed and plotted over time.

Metabolic Fluxes. For the metabolic fluxes shown in Fig. S1B and S1C, time courses of key fluxes were obtained directly from the simulation output, and modified using a moving average for visual clarity. Panel B shows several flux time courses superimposed on the metabolic network. In cases where multiple enzymes catalyze a single metabolic reaction, the sum of their fluxes is displayed. For the network diagram (Fig. S1B), a window of 45 minutes before and after the shift is shown; the larger plots (Fig. S1C) show the fluxes for all nine generations.

1.2.2 Simulation of physiological measurements (related to Fig. 2 and Fig. S2)

This section describes the investigation of the first inconsistency identified in the main text: that the total output of the ribosomes and RNA polymerases was not sufficient for the simulated cell to reproduce measured doubling times. After characterizing the inconsistency, gene groups and parameter types were identified as favorable candidates for adjustment. From these results, an iterative parameter estimation approach was developed and applied to the transcript synthesis probabilities of RNA polymerase and ribosome subunits to address the inconsistency, and resulted in the doubling time compatible cell. Please see the main text for the context of this investigation in the overall study.

Original Parameters Resulted in a Doubling Time Inconsistency: To obtain the histogram of the simulated doubling times shown in Fig. 2A, 256 simulations, each following 4 generations of cells, were simulated in aerobic glucose minimal medium with the original synthesis probabilities. To avoid bias from initialization, only the third and fourth generations of cells were analyzed. The experimentally measured doubling time of 44 minutes was informed (*29, 30*). Note that in all doubling time histograms presented in this work (Figures 2A, 2C, 2D, and S2C), the main source of heterogeneity in doubling times between individual simulations is the Gaussian noise factor that is added to the supply rate of amino acids in the translation process (See Section 1.1.6.4 for details).

Identification of Candidate Gene Groups via Parameter Sensitivity Analysis: It was challenging to determine the best way to analyze the parameter values *in toto*. All of the parameters included in Table S1 carry independent information in the model, as they are derived from separate datasets and measurement types and are implemented in different parts of the model. Thus, even though some parameter categories can have similar effects on model states (e.g., either increasing translation efficiency or increasing protein half lives will lead to higher protein counts), the dynamics will differ (e.g., increasing translation efficiency will increase the rate of protein synthesis while increasing protein half lives will decrease the rate of protein degradation). Additionally, while there might be some correlation between certain values (e.g., two genes on the same transcription unit would be expected to show correlated expression across measurements and conditions), the extent of this correlation remains unclear in most cases. We therefore required methods that could simultaneously capture the parameters' independence as well as potential interactions with regard to influence on simulation output.

To determine the impact of each parameter on the simulated cell's growth rate, 20,000 simulations with varied parameters were run for 10 seconds each. In each simulation, 10% of the parameters were randomly chosen and increased by a factor of 5 and another 10% were randomly chosen and decreased by a factor of 5 (0.2x original value). Then, the average growth rate was calculated for each simulation using the last 5 time steps (to allow an initial adjustment period). The effect (e_i) of each parameter on the simulated growth rate was determined by computing the difference in growth rates when the parameter was adjusted to the higher value and the lower value. A z-score for each parameter can be calculated by using the mean (μ) and standard deviation (σ) of the effect across all parameters. Therefore, the z-score (z_i) for parameter i with effect e_i will be:

$$z_i = \frac{e_i - \mu}{\sigma}$$

Because many parameters are being adjusted simultaneously, the z-score cutoff was determined with a p-value of 0.05 that was adjusted for multiple hypothesis testing for the 17,822 parameters that were adjusted. A z-score between -4.54 and 4.54 should contain the data with a probability of $1 - \frac{0.05}{17822}$ as determined with the inverse error function below:

$$z_{cutoff} = \sqrt{2} \cdot \text{erf}^{-1}(2p - 1) = \sqrt{2} \cdot \text{erf}^{-1}\left(2\left(1 - \frac{0.05}{17822}\right) - 1\right) = 4.54$$

The parameter sensitivity analysis described here identified candidate gene groups that are most likely to impact the simulated cell's growth rate: genes related to the expression of RNA polymerase, ribosomes, and RNAses (Fig. 2B).

Iterative Parameter Estimation Approach: The iterative parameter estimation approach described here is used to adjust the transcript synthesis probabilities of specified gene groups such as subunits of RNA polymerase or ribosomes (as in Fig. 2C) and their combination (as in Fig. 2D).

First, we present an equation which can be used to link either ribosomal or RNA polymerase concentration to the production of protein or RNA, respectively. For a given species (either protein or RNA) with concentration x_i, the rate of production depends on the ribosomal or RNA polymerase concentration dedicated to production of that species c_i. Each species will also have a length l_i in dimensions of monomers per product, whether in terms of nucleic acid or amino acid. Finally, ribosomes and RNA polymerases both have a given elongation rate r (in dimensions of monomers per polymerase per unit time). The loss rate of the protein or RNA is given as β_i, which is a function of x_i (obtained from the cellular growth rate, and first-order decay for protein, or Michaelis-Menten kinetics for RNA, see Section 1.1.6.3 above), with dimensions of product concentration per unit time. Taking these terms together, we have the following equation:

$$\frac{dx_i}{dt} = l_i^{-1} r c_i - \beta_i \tag{59}$$

Setting $dx_i/dt = 0$ and solving for c_i gives us the amount of polymerase needed for each product:

$$c_i = \frac{\beta_i l_i}{r} \tag{60}$$

To determine the ribsomal or RNA polymerase requirements for production of all species (total protein or RNA), we can write the following aggregate equation for the *total* amount of polymerase C, which is the sum over the components:

$$C = r^{-1} \beta \cdot l \tag{61}$$

Here $\beta \cdot l$ is the dot-product.

It is important to remember that the production of ribosomes and RNA polymerases also depends on their own concentration, making them unique amongst the protein and RNA species. To obtain the final determination of ribosomes and RNA polymerases required to produce cellular products, we require an iterative algorithm. We first update the vector x such that the concentration of either the ribosomal or RNA polymerase species is updated with the result C. Changing x also changes the total mass of that component (the total protein or RNA mass), which we wish to maintain. Thus, we normalize the abundances according to their molecular weights m and target total mass M to achieve x':

$$x' = \frac{M}{m \cdot x} x \tag{62}$$

Here $m \cdot x$ is the dot-product. Now we must recompute β, as it is a function of x with the general form:

$$\beta(x) = \mu x + kx + v(x) \tag{63}$$

where μ is the specific growth rate (dilution rate constant), k is a first-order decay constant, and v is some Michaelis-Menten kinetic rate law for degradation. We recompute $\beta \to \beta\left(x'\right)$, and then use this new value of β starting with Equation 61. This procedure is repeated until the change in x over the iteration (Δx) becomes trivially small; in practice we choose the termination condition

$$\sum_i \sqrt{\Delta x_i^2} < 10^{-9} \tag{64}$$

This condition is met in approximately 5 iterations. Note that, in the simplest cases, it may be possible to find a closed-form solution for this problem (i.e. a solution that does not require iteration). However this procedure is carried out in parallel for both the ribosomes and RNA polymerases, and furthermore the expression of β can be difficult to invert. Thus, this iterative approach, which requires minimal back-calculation, is preferable.

From the iterative parameter estimation approach presented here, we obtain x to determine the RNA synthesis probabilities. Similarly to Fig. 2A, 256 simulations, each following 4 generations of cells, were simulated in aerobic glucose minimal medium in each of the following conditions: new RNA polymerase synthesis probabilities (Fig. 2C, left), new ribosome-related synthesis probabilities (Fig. 2C, middle), and new synthesis probabilities for both the RNA polymerases and ribosomes (Fig. 2D). To avoid bias from initialization, only the third and fourth generations of cells are analyzed. Fig. S2C shows similar histograms for aerobic glucose medium supplemented with amino acids and for anaerobic glucose minimal medium, both with the new synthesis probabilities for RNA polymerases and ribosomes (as in Fig. 2D). The fold-changes for all genes following the application of this procedure to both RNA polymerases and ribosomes (versus the original expression levels) are provided in Table S2B.

To estimate new mRNA half lives (Fig. S2A, column 9), the loss rate β_i in Equation 59 was specified as:

$$\frac{dx_i}{dt} = l_i^{-1} r c_i - \left(\frac{\ln 2}{\tau} + k_d\right) x_i \tag{65}$$

where k_d is the mRNA degradation rate ($= \ln 2/\text{half life}$) and τ is the doubling time of the cell. Then, rather than isolating the concentration c_i as was done to reach Equation 60, k_d was isolated:

$$k_d = l_i^{-1} r c_i x_i^{-1} - \frac{\ln 2}{\tau} \tag{66}$$

Equation 61 is used to find the target RNA polymerase abundance, which is used as an input to Equation 66 to find mRNA degradation rates (for subunits of RNA polymerase and ribosomes only) that minimizes differences between the number of RNA polymerases that can be formed and the number of RNA polymerases needed to maintain steady state concentrations of RNAs. During the iterations, the degradation rates have a lower limit of zero (ie. cannot become negative) and, similarly to before, the same termination condition (Equation 64) is applied.

Validation and Parameter Variability Investigation: To validate the state of the cells simulated by the new synthesis probabilities for both the RNA polymerases and ribosomes (Fig. 2D), the abundances of RNA polymerase and ribosomes at the conclusion of the iterative parameter estimation approach described by Equation 59 through 64 (and represented schematically in Fig. S2B) were compared with those reported in literature (*7*) in Fig. 2E.

With the understanding that the cell state in Fig. 2D is well-validated, we pursued a mechanistic understanding (with regard to the model) of why the pre-calibration cell state (depicted in Fig. 2A) was growth-rate-incompatible (ie. does not double in the expected doubling time). To do so, all parameters which could possibly contribute to the RNA polymerase and ribosome counts were considered for their potential to impact the growth-rate of the simulated cells. With the exception of parameters that were assumed to have little variability across data sources (the molecular masses of mRNA transcripts and protein monomers, and their lengths in number of nucleotides or amino acids), all other parameters participated in the analysis summarized in Fig. S2A. These parameters were evaluated according to three criteria: consistency with (1) literature reports, (2) expected abundances of RNA polymerase and ribosome, and (3) expected doubling time.

First, we considered literature consistency. Parameter values that have been reported multiple times in the literature give us an opportunity not only to assess the possible variability of the value itself, but also to run simulations using any of the values - in order to determine how much the simulation output is affected. Thus, when available, direct measurements were compared to the parameter values as validation (Table S2A). Validation data could not be identified for three parameters: R-protein half lives, RNA polymerase elongation rate, and RNA Polymerase active fraction. For such cases, instead of changing the original simulation value with a new literature measurement ("swap"), we performed a more detailed computational analysis. For the R-protein half-lives, the N-end rule only provides two half life possibilities. Thus, the R-protein half lives, which had been assigned 10 hours, were swapped for 2 minutes. For the RNA polymerase elongation rates, we assigned stable RNAs the faster 85 nucleotides per second rate described by Bremer and Dennis 1996 (*31*) (the reason for this rate difference between stable and unstable RNA is unknown, but Bremer and Dennis suggest it may be related to differences in polymerase pausing). For the RNA polymerase active fraction, we simply used the theoretical maximum (=1). Moreover, two parameters could not be validated by the validation data (mRNA expression and mRNA half lives). In these two cases of highest variability, we used the parameter estimation method described above in this section (on page 65) to determine new parameter values. After either parameter swapping or redetermining, cell simulations were performed. The abundances of RNA polymerase and ribosome and the doubling times were then observed from the cell simulations.

We found that only after mRNA expression redetermination, cell simulations could simultaneously meet the expected RNA polymerase and ribosome abundances, and expected doubling time (Fig. S2A, column "mRNA Expression"). None of the other parameter changes simultaneously satisfied all criteria, suggesting that the growth-rate-compatible cell state produced by mRNA expression redetermination is the most supported representation of the cell (highlighted in Fig. 2F and G). The Pearson correlation coefficient and p-value shown in Fig. 2F were computed by the scipy.stats.pearsonr program from the SciPy package. The expected experimental values shown in Fig. 2G (as orange lines) were determined from interpolation of the per-cell abundances (reported in (*7*)) of RNA polymerase and ribosome across doubling times. Interpolation was performed by the interpolate.splrep and interpolate.splev programs from SciPy.

For the validation comparisons made in Fig. 2H, we used the same sets of simulations for each medium that were used to generate Fig. 2D and Fig. S2C, with the new synthesis probabilities. The lengths of the error bars in the simulated values for all four plots indicate the standard deviations of each property calculated over the 1024 cells that were simulated for each medium. For the cellular properties plotted in Fig. 2H, the measured values ("experiment") were obtained from (*31*), while the simulation values were calculated from the simulation outputs as follows:

- RNA Mass: The total mass of RNA at each simulation time step was directly extracted from the simulation outputs, and averaged over the duration of the cell cycle.

- Number of origins at initiation: The number of origins of replication at each simulation time step was directly extracted from the simulation outputs. Then, we extracted the number that corresponds to when chromosome replication was initiated in the simulation.

- Ribosome elongation rate: The effective elongation rate of ribosomes, defined as the number of amino acids elongated by an active ribosome per second, was directly extracted from the simulation outputs, and averaged over the duration of the cell cycle.

- rRNA initiation rate: The number of transcription initiation events for all ribosomal RNAs at each simulation timestep was directly extracted from the simulation outputs. Then, these numbers were divided by the length of each simulation timestep in seconds to yield the initiation rate. The resulting rates were summed up for all rRNAs, and averaged over the duration of the cell cycle. For proper comparison with the values reported in literature, the units of the values were converted from 1/sec to 1/min, and divided by the number of rRNA types (=3).

We could also observe that by adjusting the expression of RNA polymerases and ribosomes, our simulations could recapitulate another experimental measurement that was not used to determine any parameters of our model. Scott et al., 2010 (*32*) showed that the RNA/protein mass ratio and the growth rate of *E. coli* cells always display a fixed linear relationship, regardless of the strain of *E. coli* used or the growth environment. To test whether our simulations agreed with this measurement, we additionally ran four generations of simulations with 256 random seeds for each media condition, using the raw parameters without adjustments in RNA polymerase and ribosome expression, and compared the results with the previous set of simulations run with adjusted parameters. Only the simulation results from the third and fourth generation of cells were selected to avoid bias from initialization. The results in Fig. S2D show that by adjusting these expression parameters, the mean values for the RNA/protein mass ratios and the growth rates calculated from our simulations move significantly closer to the proposed linear relationship. This comparison further verifies that the original set of parameters is unable to lead to balanced growth, and that our adjustments are able to resolve this discrepancy.

The same set of simulations were also used for the adder-sizer behavior comparison plots in Fig. S2E. Our model couples each chromosome replication initiation event to a cell division event which occurs after a fixed period of time for DNA replication and cytokinesis (the so-called C and D periods, occurring zero, one or two generations in the future depending on the growth rate), as inspired by recent work (*33*). We found that these simulations were capable of running for many generations with stable cell size distributions under multiple conditions - in dramatic contrast to our original *M. genitalium* model, which could only grow stably for one cell cycle (*34*). Again, among the four generations of simulations only the simulations from the third and fourth generation of cells were selected to avoid bias from initialization. From each simulation output, the initial mass of the cell was subtracted from the final mass to compute the added mass of the cell, and was normalized by the mean of all added masses to yield normalized added masses. Likewise, the initial masses extracted from the simulation outputs were normalized by the mean initial mass to yield normalized initial masses. The resulting plot shows that our simulation displays adder behavior under fast growth conditions (minimal glucose plus 20 amino acids), and sizer behavior under slower growth conditions (minimal glucose and anaerobic minimal glucose), in agreement with observations by recent studies (*4, 5, 33, 35*)

In Fig. S2F, we directly compared the cell size outputs of our model to Figure 7B of Wallden et al., 2016 (*33*), where the authors plotted the birth and added volumes of *E. coli* MG1655 strain JJC5350 growing under three different nutrient conditions. Our simulations were able to qualitatively reproduce the observed adder-sizer behavior in this figure just as they did in Figure S2E - there was a higher correlation between the birth and added volumes for cells under slower growth - but the quantitative values for the cell volumes did not compare well to experimental data. This is mainly because the cell volumes of our model are calculated from parameters measured for *E. coli* strain B/r, which was shown to have a significantly smaller cell volume compared to strain MG1655 which was used by the referenced experiment. Additionally, the experiment used different nutrient conditions than the three upon which our model is based, further preventing us from making a meaningful comparison.

1.2.3 Analysis of the metabolic network sub-model. (Related to Fig. 3 and Fig. S3.)

The consistency of the metabolic network and reaction kinetic parameters was assessed in Fig. 3 and Fig. S3. Plots for these figures were obtained as follows:

CdsA downstream metabolite concentrations. The concentrations of the metabolites phosphatidylethanolamine (PE) and phosphatidylglycerol (PG) in Fig. 3A were determined using the metabolite counts from eight cells, each simulated through 32 division cycles. The concentration at each time step was calculated as follows:

$$\text{Metabolite Concentration} = \frac{\frac{1}{\text{Avogadro's Number}} \cdot \text{Metabolite Counts}}{\frac{1}{\text{Cell Density}} \cdot \text{Cell Mass}} \tag{67}$$

where the metabolite counts, cell density and cell mass were extracted directly from the simulation outputs. “Low CdsA” indicates simulations were run with the original CdsA half-life of 2 minutes, while “High CdsA” indicates simulations were run with the updated CdsA half-life of 10 hours.

Glucose Yield Glucose yield (Fig. 3B and 3G) was determined at each time step from simulations of four cells, each simulated for one division cycle. It was calculated as follows:

$$\text{Yield} = \frac{\frac{\text{Mass Added}}{\Delta t}}{\text{Glucose Uptake Rate} \cdot \text{Cell Dry Mass}} \tag{68}$$

where Mass Added is the mass added to the cell in a time step, Δt is the length of the time step, Glucose Uptake Rate is the glucose flux, and Cell Dry Mass is the dry mass of the cell at the given time step, which are all taken from the simulations. “Disabled Succ/Fum Constraints” indicates the simulations were run with all kinetic constraints enabled, except for those for tRNA synthetases, succinate dehydrogenase and fumarate reductase. “Enabled Succ/Fum Constraints” indicates the simulations were run with all kinetic constraints enabled, except for those for tRNA synthetases. “New Disabled Constraints” indicates the simulations were run with all kinetic constraints except for those for tRNA synthetase, succinate dehydrogenase, NADH dehydrogenase, inorganic pyrophosphatase, glutathione reductase, isocitrate dehydrogenase and glyoxylate reductase. See Main Text and Section 1.1.6.7 for more details on the constraints selected.

Fluxome Validation. To validate the model flux with data that was not included in the model, we compared the model fluxome to a measured fluxome dataset (*22*) in Fig. 3C and 3H. The simulated flux is the average flux through each reaction over time within a selected cell cycle. Error bars show the standard deviation for the simulated and measured fluxome. Correlation values were determined with (blue and orange) and without (blue only) the identified outlier reactions.

Single constraint flux impact. To determine the impact of perturbing the kinetic reaction target in the metabolic objective on the flux through either outlier reactions identified from the fluxome validation (succinate or isocitrate dehydrogenase), the following global analysis was performed (Fig. 3D). A simulation was run for 10 seconds; at each time step, the metabolism linear optimization problem was solved with all but one reaction target included (ie. solved n times, each time with one of n reaction targets excluded). This strategy results in a different flux solution for each optimization problem, which reveals the impact of removing a single target. Specifically, identifying the constraints that reduce the previously identified outlier fluxes (succinate and isocitrate dehydrogenases) towards their measured values upon perturbation suggests where kinetic parameters may be inconsistent. The relative change in the desired flux (e.g. succinate dehydrogenase, Fig. 3D, or isocitrate dehydrogenase, Fig. S3H) at each time step, t, can then be determined:

$$v_{rel,i}^t = \frac{v_{-i}^t - v_{all}^t}{v_{all}^t} \tag{69}$$

Where $v_{rel,i}^t$ is the relative flux change when removing the target for reaction i, v_{-i}^t is the flux of interest when all targets except reaction i are included in the objective and v_{all}^t is the flux of interest when all targets are included in the objective.

Finding the average relative flux, $v_{rel,i}$, for each reaction over the simulation results in a distribution. A z-score (Fig. 3D), z_i, for each reaction can then be determined by finding the mean, $\mu_{v_{rel}}$, and standard deviation, $\sigma_{v_{rel}}$, for all $v_{rel,i}$.

$$v_{rel,i} = \frac{\sum\limits_{t} v_{rel,i}^{t}}{\text{\# times teps}} \tag{70}$$

$$z_i = \frac{v_{rel,i} - \mu_{v_{rel}}}{\sigma_{v_{rel}}} \tag{71}$$

where a negative z-score for a reaction indicates that removing its target from the kinetic objective results in a lower flux for the flux of interest.

Factorial design for constraint interactions. Individual effects identified with the z-score above does not reveal the interaction effects between kinetic target effects on fluxes of interest so a factorial design experiment was performed to identify any potential interactions between identified kinetic constraints Fig. 3E). A set of 256 constraint conditions (all the combinations of including or removing each of the eight kinetic constraints) were simulated, with four cells for one division cycle for each constraint condition. Glucose uptake flux was computed directly from the simulation, taken as an average over the course of all simulations for a given constraint condition. The comparison to validation data is calculated from the simulation and validation data:

$$\log_2 \frac{\text{Simulation Flux}}{\text{Validation Flux}} \tag{72}$$

where Simulation Flux is the average succinate dehydrogenase flux from all simulations for a given constraint condition and Validation Flux is the succinate dehydrogenase flux from the literature (*22*).

k_{cat} **predictions from simulations.** Using the simulated fluxes, and concentrations of enzymes and metabolites, a predicted k_{cat} (Fig. 3F) can be determined at each time step:

$$k_{cat} = \frac{\text{Simulation Flux}}{\text{Enzyme Concentration} \cdot \text{Enzyme Saturation}} \tag{73}$$

where Simulation Flux is the flux for the given reaction, Enzyme Concentration is the concentration of the catalyzing enzyme for the given reaction and Enzyme Saturation is:

$$\text{Enzyme Saturation} = \begin{cases} \frac{\text{Metabolite Concentration}}{K_M + \text{Metabolite Concentration}} & \text{if } K_M \text{ and Metabolite Concentration are available} \\ 1 & \text{otherwise} \end{cases} \tag{74}$$

where Metabolite Concentration is the concentration of a metabolite with a known K_M value and K_M is a curated Michaelis-Menten constant for the given reaction. If a K_M value or metabolite concentration is not available, the enzyme is assumed to be fully saturated for this calculation.

This analysis is meant to assess how well the new model with relaxed constraints fits the values reported in literature. The calculated k_{cat} values could also be affected by metabolite concentrations, curated K_M values and protein expression but for purposes of comparing the reported k_{cat} values and the model output, it is assumed that enzyme concentration and enzyme saturation are accurate.

Along these same lines, similar distributions could have been generated for metabolite concentrations, K_M values and protein expression (data not shown) but with important caveats. For example, metabolite concentrations and K_M are not as widely reported in literature for all the reactions of interest, meaning that the comparison would be less comprehensive. Moreover, protein expression datasets are often more consistent with each other than reported k_{cat} values, which can vary by an order of magnitude or more and have difficulties in obtaining measurements representative of *in vivo* conditions. Thus the insights to be gained by comparing simulation output to proteomics data are likely to be relatively limited.

Normalized Growth. To produce the plot in Fig. 3I, S3A and S3B, the masses of different cell fractions (protein, tRNA, rRNA, mRNA, DNA and small molecules) were normalized to the starting masses of the respective fraction at each time point.

Flux PCC. To compare the flux values that would have been calculated using Michaelis-Menten kinetic equations (“Target Flux”) with the values obtained by the simulation (“Simulated Flux”) in Fig. 3J, S3C and S3D, we used the kinetic parameters, together with the enzyme and small molecule concentrations where applicable, to calculate the target flux for each time point of each cell cycle. The simulated flux at each time point was obtained directly from the simulation. These were then averaged over all time points and cells. The Pearson correlation between the log of the average target flux (+10^{-6} to prevent $log(0)$) and the log of the average simulated flux (+10^{-6}) was determined for each weighting parameter set. The correlation was determined for all reactions (blue and gray) and for only reactions with a non-zero simulated flux at some point in the simulation (blue only).

Determining value of the kinetic weighting parameter. To determine an appropriate tradeoff between the homeostatic and kinetic objectives in the metabolism optimization problem, we ran ten simulation sets, where each set contained eight separate simulations of a cell over eight generations. As described above, one daughter cell was chosen to continue at each division, such that each simulation included eight life cycles. Each simulation had its own random seed. Each of the simulation sets was run with a different weighting for the kinetics part of the objective (λ), and the homeostatic weighting was always set equal to 1-λ. The optimal value of λ was determined by maximizing an objective function that included various metrics that are expected to tradeoff as the weighting changes. All values were normalized to be between 0 and 1 and weighted equally in the objective. The metrics selected can be grouped as related to satisfying the homeostatic objective (viable sims, growth rate, concentration correlation and concentration deviations) or satisfying the kinetic objective (flux correlation, flux deviations, nonzero flux correlation, nonzero fluxes and validation correlation). As seen in Fig. S3A and S3B, increasing λ causes the cell to fail to meet the homeostatic objective and as a result growth suffers. As seen in Fig. S3C and S3D, increasing λ also increases the number of reactions that have non-zero flux so that additional information (in the form of kinetic parameters and fluxes) can be contained in the model. The results shown in Fig. 3I and 3J show that the value of λ selected through this objective produces desired results related to both the homeostatic and kinetic objectives. The objective to determine the value of λ can be formalized as below with values coming from simulations at different values of λ as shown in Fig. S3E:

$$\max_{\lambda}(v + \mu + r_{conc} + f_{conc} + r_{flux} + f_{flux} + r_{nonzero} + f_{nonzero} + r_{val})$$

$$\begin{aligned}
\text{Viable sims:}\quad & v = \frac{n_{viable\_sims}}{n_{total\_sims}} \\
\text{Growth rate:}\quad & \mu = \frac{\bar{\mu}}{\mu_{expected}} \\
\text{Concentration correlation:}\quad & r_{conc} = \text{pearson}(\bar{C}, C_o) \\
\text{Concentration deviations:}\quad & f_{conc} = 1 - \frac{n_{conc\_off\_axis}}{n_{total\_conc}} \\
\text{Flux correlation:}\quad & r_{flux} = \text{pearson}(\bar{v}, v_o) \\
\text{Flux deviations:}\quad & f_{flux} = 1 - \frac{n_{flux\_off\_axis}}{n_{total\_fluxes}} \\
\text{Nonzero flux correlation:}\quad & r_{nonzero} = \text{pearson}(\bar{v}, v_o) \\
\text{Nonzero fluxes:}\quad & f_{nonzero} = \frac{n_{flux>0}}{n_{total\_fluxes}} \\
\text{Validation correlation:}\quad & r_{val} = \text{pearson}(\bar{v}, v_{val})
\end{aligned}$$

Objective Tradeoff. The plot in Fig. S3F shows the value for each objective component (homeostatic and kinetics) at each kinetic weighting parameter. The objective components were averaged across all time points and cells within each set of simulations that had a different weighting parameter.

1.2.4 Sub-generational expression analysis (related to Fig. 4 and Fig. S4).

To perform the sub-generational expression analyses depicted in Fig. 4, we performed a simulation of 32 generations of log-phase growth on glucose minimal media under aerobic conditions (with one daughter selected at the end of each cell cycle as before). The panels in Fig. 4 all arise from these simulated generations.

Independent validation - comparison to proteome data previously withheld from model construction. To perform a global comparison of simulation-predicted protein expression with a proteomics dataset that was withheld from model construction (Fig. 4A), we first obtained the measured values as given in Schmidt et al. Table S5, Column H (Glucose) (*11*), and determined the log10 of the average count per cell (after incrementing by +1 to avoid log10 of zero) for each protein. To obtain the simulated values, we used the first generation of the simulation set described above, and determined the log10 (again, after incrementing by +1) of the average count per cell as a time average over the course of the cellular life cycle. The R^2 values were calculated for the log-transformed data separately for the two groups of proteins: For the highly abundant proteins whose measured counts are greater than or equal to 30, R^2 was equal to 0.614. For the low-abundance proteins with measured counts less than 30, R^2 was equal to 0.015. The average log deviations of simulated counts from the measured counts were 2.477 and 1.667, respectively, for the two groups of proteins.

Dynamics of exponential and sub-generational transcription over multiple generations. For Fig. 4B, examples of exponentially and sub-generationally transcribed genes were found by computing ratios of the final count of proteins to the initial count of proteins. Examples of exponential dynamics were identified by filtering for proteins with ratio values between 1.6 and 2.4 for each generation of a 6-generation long window. Examples of sub-generational dynamics were identified by filtering for proteins with ratio values less than 1.1 in the first generation, greater than 10 in the second generation, and less than 1.1 in the remaining generations of the same 6-generation long window. For the 6-generation window analyzed, 30 suitable examples of exponential dynamics and 31 suitable examples of sub-generational dynamics were found; one of each was chosen to illustrate this concept.

Global assessment of sub-generational gene expression. To determine the frequency of observing at least one transcript per generation for all translatable genes in the genome, the simulation set was used to obtain time series data for the mRNA counts. Each mRNA time series was then assessed, generation by generation, to determine whether the mRNA count was ever positive during that cell cycle. The result was a number between 0 and 32 which gave the number of generations with a non-zero mRNA count. This was divided by the total number of generations - 32 - to obtain the frequencies shown in Fig. 4C.

The distribution of essential genes into transcriptional frequency groups shown in Fig. 4D was generated analogously, but only for genes considered to be essential under minimal glucose media by Baba et al. (*36*) and Joyce et al. (*37*). Details on how this list of essential genes was curated is described in the paragraph below. The protein absence plot shown in Fig. 4E was generated by observing the counts of the functional units that are generated from sub-generationally transcribed genes. These functional units include 1915 protein monomers that do not form protein complexes, and 504 protein complexes that are formed from monomers produced from sub-generational genes. The same list of essential genes was used to determine the essentiality of each functional unit and plot the blue bars in Fig. 4E. Note that a protein complex was considered to be essential if any of its subunits are produced from essential genes that are sub-generationally transcribed. Table S4 shows the full list of essential genes whose protein product (monomer or complex) is absent at least once during the 32-generation simulation.

Determining gene essentiality. For Fig. 4D and Fig. 4E, we used a list of 406 genes that were shown to be essential for the growth of *E. coli* cells under minimal glucose media by experiments. The list was curated based on the data in Baba et al., 2006 (*36*), with some modifications to more accurately determine essentiality under our simulation conditions. First, all 300 annotated genes (for strain MG1655) that were shown to be essential for growth in rich LB media by Baba et al. were included in this list. This assumes that genes that are essential for growth under rich media will also be essential for growth under minimal media. Next, 119 genes that were labeled as being conditionally essential under glucose minimal media by Joyce et al., 2006 (*37*) were added to this list. These 119 genes were labeled as such because the knockout strains of these genes were able to grow in LB but displayed the slowest growth under glucose minimal media according to Baba et al. Among these conditionally essential genes, 13 genes that were assumed to be false positives by Joyce et al. (see Figure 5 of the main text) were excluded from the list. These genes are *aceF*, *atpE*, *dnaT*, *lipA*, *lipB*, *lpd*, *pfkA*, *priA*, *ptsH*, *ycaL*, *argB*, *argC*, and *metE*.

Downstream effects of sub-generational transcription. For Fig. 4F, the transcription events from the *pabA* and *pabB* promoter, as well as the resulting counts of mRNA, PabA and PabB monomers, the PabAB heterodimer, and the final reaction flux, were all obtained directly from the simulation. The counts of the final product metabolite, 5,10-methylene-THF, were also obtained directly from the simulation.

Sensitivity of the 5,10-methylene-THF depletion phenotype to *pabB* mRNA transcript synthesis probability. To show that the PabAB heterodimer and 5,10-methylene-THF depletion phenotypes we observed in Fig. 4F depend on the synthesis probability of the mRNA transcripts, we performed a parameter sweep analysis above and below the wild-type transcription probability of *pabB* (Fig. S4). Each of the nine parameter values simulated (shown along the x-axis) was assessed by eight trials of eight generation-long simulations – with each of the eight trials initialized on a different random seed – totaling to a set of 576 (i.e., 9 parameter values × 8 trials × 8 generations) simulations of log-phase growth on glucose minimal media under aerobic conditions. Average fraction of time with zero counts of the PabAB complex, and average fraction of time with 5,10-methylene-THF products below 0.1% of their wild-type target concentrations, were determined and plotted in Fig. S4.

1.2.5 Protein steady-state analysis (related to Fig. 5A)

The steady state analysis was performed to compare expected model output to a validation dataset that was not used in the construction of the model. With the expectation that the model will approach steady state protein concentrations that match the validation proteomics dataset, the difference between the rate of production expected using model parameters (x-axis) and rate of loss using model parameters but at protein concentrations from the validation dataset (y-axis) should be 0. Differences between the rates will result in the model reaching a different steady state concentration than the validation dataset, which suggests discrepancies in the parameters or validation data. The rate of change of protein concentration was described as:

$$\frac{dp}{dt} = e_r \cdot r \cdot RIB :: RNA_{aff} \cdot RIB_{act} - \Big(\frac{\ln(2)}{t_{1/2}} + \frac{\ln(2)}{\tau}\Big)p$$

where the positive term on the right side is the production rate, the negative term is the loss rate, and the variables and their sources are described below:

Variable	Description	Source or Value
e_r	ribosome elongation rate	Bremer & Dennis 1996 (*31*)
r	RNA concentration	This Study
$RIB :: RNA_{aff}$	translational efficiency	Li et al. 2014 (*10*)
RIB_{act}	fraction of active ribosomes	Bremer & Dennis 1996 (*31*)
$t_{1/2}$	protein decay rate	Tobias et al. 1991 (N-end rule) (*12*)
τ	doubling time	45 minutes
p	protein concentration (Not used in model construction)	Schmidt et al. 2016 (*11*)

Note that only genes for which all the data was available (i.e. genes that had RNA-sequence data (this study), translational efficiencies (*10*), and proteomics data (*11*)) could be included in this comparison.

The outliers highlighted in Fig. 5A (excluding CdsA, which was identified for further investigation earlier in the main text) were chosen based on the highest discrepancy between the production and loss rates (not including the genes related to ribosomal or RNA polymerase subunit expression).

The plots for Fig. 5B, 5C, and S5 are described in more detail in Section 1.3.

1.3 Experimental Methods

1.3.1 RNA Sequencing and Analysis

RNA sequencing was performed on MG1655 cells grown in two different environments, in triplicate: (1) M9 minimal media (48.8 mM Na_2HPO_4, 22 mM KH_2PO_4, 8.6 mM NaCl,18.7 mM NH_4Cl, 2 mM $MgSO_4$, 0.1 mM $CaCl_2$) with 0.4% Glucose or (2) M9 minimal media with 0.4% Glucose supplemented with amino acids (5x Supplement EZ without VA Vitamin Solution), prepared as has been described (*38*).

Cells were grown overnight in requisite media at 37°C, on a platform shaker. In the morning, cultures were back diluted to an $OD_{600\,nm}$ 0.02 in 10 mL of media contained in a 125 mL flask, and were grown up at 37°C, on a platform shaker. Cells were harvested at $OD_{600\,nm}$ 0.4. Total

RNA was extracted from 2 mls of each culture using Qiagen RNAeasy Protect Bacterial Mini Kit (Qiagen #74524) with RNAprotect Bacteria Reagent (# 76506) and RNase-Free DNase Set (# 79254) according to manufacturer's instructions. Performed cell lysis using lysozyme from ThermoScientific #90082. NanoDrop was used for RNA quantification. rRNA depletion was performed using RiboZero rRNA Removal Kit (Epicentre # MRZGN126), according to manufacturer's instructions. RNA quality, prior to sequencing, was assessed using an Agilent 2100 BioAnalyzer (# G2938C), according to manufacturer's instructions by the SFGF facility at Stanford University.

Library prep was performed by the SFGF facility at Stanford University according to the TruSeq Stranded Total RNA Sample Preparation Guide. Paired-end sequencing with read lengths of 75 bp was performed by the SFGF facility at Stanford University on an Illumina NextSeq 500. Approximately 20 million reads were obtained per sample.

BBMap 34.33 was used to pre-process sequencing data to trim reads, remove reads for common contaminants, and remove reads that map to non-coding RNA (*39*). RSEM 1.2.19 was used for downstream processing and calculation of gene expression (*40*). Sequencing data is available at GEO with accession number GSE85472.

1.3.2 Protein Half-Life Measurement

Plasmids encoding his-tagged genes of interest (from the ASKA library without GFP (*41*)) were transformed into MG1655 (See Table S5 for plasmids used). Duplicates of bacterial cultures were grown overnight at 37°C in M9 minimal media with 0.4% glucose and 20 µg/mL chloramphenicol for plasmid selection. In the morning, bacterial cultures were diluted to $\text{OD}_{600\text{ nm}}$ 0.03, and incubated at 37°C until they reached $\text{OD}_{600\text{ nm}}$ 0.3. At this point, bacterial cultures were diluted 1:2 in minimal media supplemented with IPTG (0.1mM) to induce protein overexpression. Cultures were grown on a shaker in a 37°C warm room for the requisite time of induction.

After the requisite time of IPTG induction (see Table S5), a 9 mL sample was taken to measure the time 0 protein level. Then, 10 µg/mL tetracycline (the 10 mg/mL stock was made in 95% ethanol) was added to the rest of the culture to inhibit protein synthesis. Culture was then returned to 37°C. 9 mL samples were taken at indicated time points (10 min on ice followed by centrifugation for 10 min at 4000g, 4°C) to measure protein levels (see Table S5). At each time point, culture $\mathrm{OD}_{600\,\mathrm{nm}}$ was measured. Cells were lysed using BugBuster Master mix (Millipore, #71456-3) supplemented with Halt protease/phosphatase inhibitor cocktail (Thermo Scientific #78444), following the manufacturer's instructions.

Protein quantification was performed using Pierce BCA protein assay kit (Thermo Scientific #23225), following the manufacturer's instructions.

For western blot detection of proteins we ran samples on a Simon machine (Protein Simple) following manufacturer's instructions, and standard kit components (Protein Simple, Core Kit #CBS201, Mouse Master Kit #Simon-02-01). An anti-His tag antibody (Novus Biologicals #NB100-64768) was used for protein detection and an anti-RNAP β antibody (BioLegend #663006) for capillary normalization (loading control).

The measured amount of His-tagged protein (N_p) was normalized by the amount of β subunit RNAP loaded as a protein control (*RNAP*), i.e. $N = N_p/RNAP$; and then log-transformed. Linear regression was used to determine the first-order decay rate constant (k_d) as follows:

$$N = N_0 exp(-k_d t), log(N) = log(N_0) - k_d t \tag{75}$$

The half-lives in Fig. 5 and Fig. S5 (control experiments, see figure legend) were then estimated by: $t_{1/2} = log(2)/k_d$.

1.3.3 Immunofluoresence

MG1655 was transformed with His-tagged gene plasmids (from the ASKA library without GFP (*41*)) for CdsA and RpoH. Cells were grown, induced with IPTG, and treated with tetracycline as per the protein half-life measurements carried out by western blotting. At the indicated timepoints (Figure 5C, Supplemental Figure 5B, 5C and 5D) 900 µl of sample was taken and combined with 300 µl of 16% PFA (Fisher, PI-28906) for 20 minutes at room temperature. Samples were centrifuged for 2 minutes at 6000 rpm at room temperature. Pellets were resuspended in 1 ml of PBS, centrifuged, and this process repeated twice before resuspending in 500 µl of PBS. A glass bottom 96 well plate (Fisher Scientific, 164588) was coated with a 0.1% solution of poly-L-lysine (Sigma, P8920) for 1 hour at 37°C and rinsed three times with PBS. Cells were added to the plate at a 1:250 dilution, centrifuged for 5 minutes at 1000 rpm, and rinsed three times with PBS. Cells were then treated as follows with PBS washes between each step: 30 minutes in 0.1% Triton, 45 minutes in 100 µg/ml lysozyme (Pierce, 90082) and 5mM EDTA, 30 minutes in 0.1% Triton, 30 minutes in 3% BSA (Sigma, A7906) and 5% donkey serum (Jackson Immunoresearch (017-000-121), 1 hour in anti-His tag antibody (Novus Biologicals, NB100-64768) at 1:200 dilution, 1 hour in secondary antibody (Jackson Immunoresearch, 715-625-151) at 1:200 dilution, 5 minutes in DAPI. Images were acquired using a Nikon Eclipse Ti fluorescence microscope controlled by Micromanager and using a 60x oil objective with a 1.5x tube lens and 2x2 binning.

1.3.4 Absolute-Quantative PCR

MG1655 cells were grow overnight in M9 + 0.4% glucose at 37°C, then back diluted to $OD_{600\text{ nm}}$ 0.02 in the morning. When cells got to an $OD_{600\text{ nm}}$ 0.2, they were back diluted again to $OD_{600\text{ nm}}$ 0.02 and grown to $OD_{600\text{ nm}}$ 0.4. At this point Total RNA was extracted using the Bacterial RNeasy Protect Bacteria Mini Kit (Qiagen, 74524), according to manufacturers instructions (with on-column DNase treatment (Qiagen, 79254). From total RNA, cDNA was generated using iScript Reverse Transcription Supermix (Biorad, 1708840) according to manufacturers instructions, with 500 ng of input RNA. For absolute quantitative qPCR, standards were created by PCR amplification of the constructs using the same primers used for qPCR. The resultant DNA fragments were then run on a gel and extracted using the QIAquick gel extraction kit (Qiagen, 28704). Standards were diluted to a range of 10^{-1}-10^{7} counts/μL, to create a standard curve, by the following equation: molecules/μL = $\frac{DNA*10^{-9}g/L}{fragment\,length*660g/mol} \cdot 6 * 10^{23} molecules/mol$. qPCR was run using iTaq Universal SYBR (BioRad, 1725120) according to manufacturers instructions using a quantity of cDNA that corresponds to 5 ng of input Total RNA. The results for the standards were fit to a line, and an equation was found. The counts of target RNA were calculated according to the following equation: counts = $10^{(CT-b)/m} \cdot 2$. Results were multiplied by 2 since 2 μL of input cDNA was added for each standard. A total of three independent biological replicates were run.

2 Supplemental Figures

Figure S1

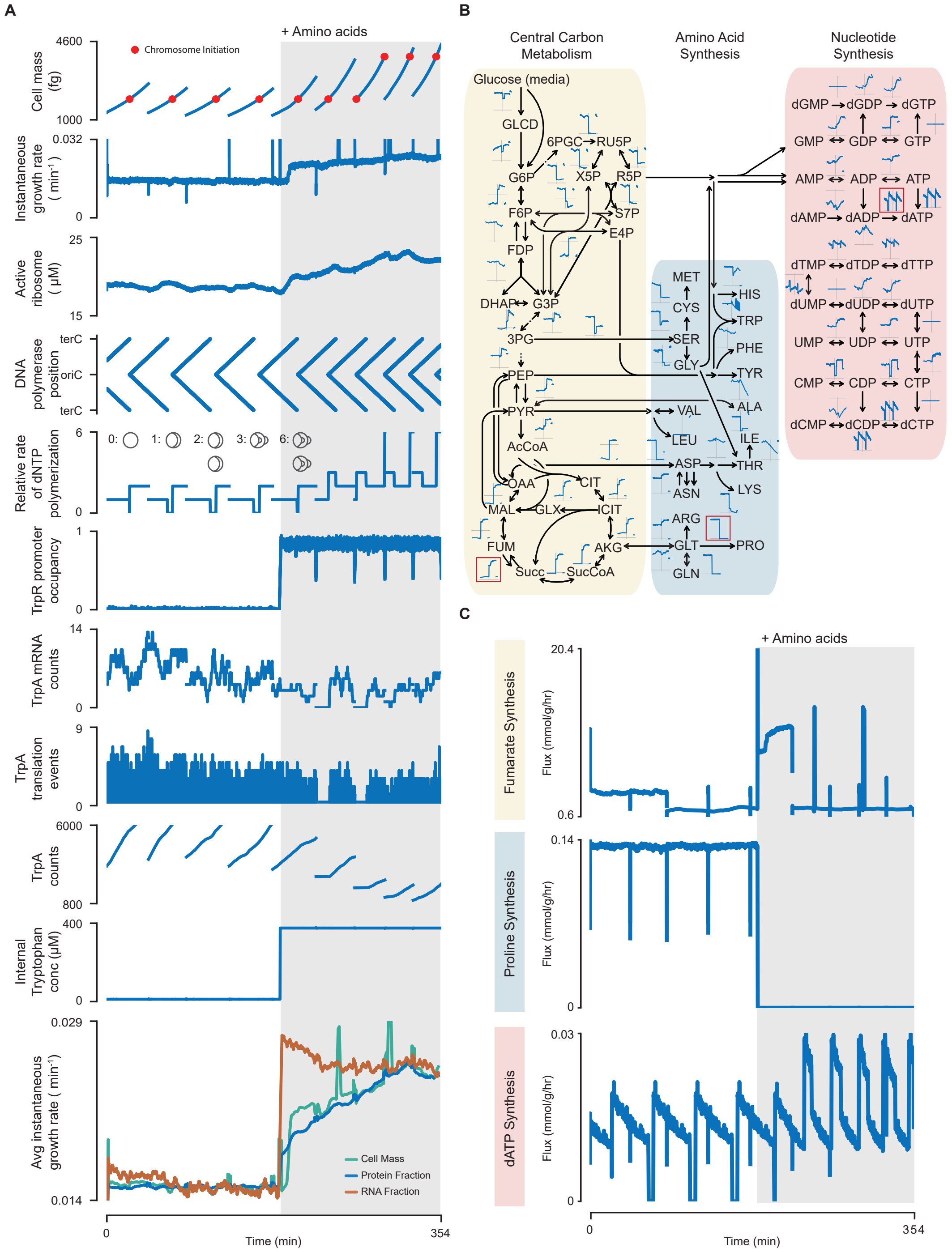


Fig. S1. Fully integrated output from model simulations of several generations of cellular growth on aerobic glucose minimal media, followed by an environmental shift in which amino acids are added to the medium. This figure represents a detailed expansion of the plots shown at the bottom of Fig. 1; note also that the output shown in Fig. 1 and Fig. S1 is only a small fraction of the models output. (A) Plot showing how, as amino acids are added to the medium, the growth rate ("Cell Mass" and "Instantaneous Growth Rate"), number of active ribosomes ("Active Ribosome") and DNA replication rate ("DNA polymerase position", "Relative Rate of dNTP Polymerization", see also the red circles in "Cell Mass") all increase in response to the shift. The underlying mechanism of these events is also modeled; as an example, the shift leads to an increased internal concentration of tryptophan ("Internal TRP conc"), activating the tryptophan biosynthesis repressor TrpR, which then occupies the promoter of biosynthesis enzyme TrpA ("Promoter Occupancy"), decreasing the average counts of the cognate mRNA ("TrpA mRNA counts") and monomer ("TrpA monomer counts", see also "Translation Events"). At bottom, we show that the growth rates of different cell fractions are balanced before a shift but become unbalanced following a shift before returning to balanced growth at a new growth rate. (B) The metabolic flux distributions for central carbon metabolism over several generations, as taken from the same set of simulations. The dynamic changes in each flux are shown for each reaction. (C) Flux dynamics of the three reactions highlighted by red boxes in (B) are shown in detail. Time simulated after the environmental shift to amino acid supplementation is represented by the gray region. See also Movie S2. Full details of the analysis required to generate this figure, as well as a pointer to the generating code, can be found in Section 1.2.

Figure S2A

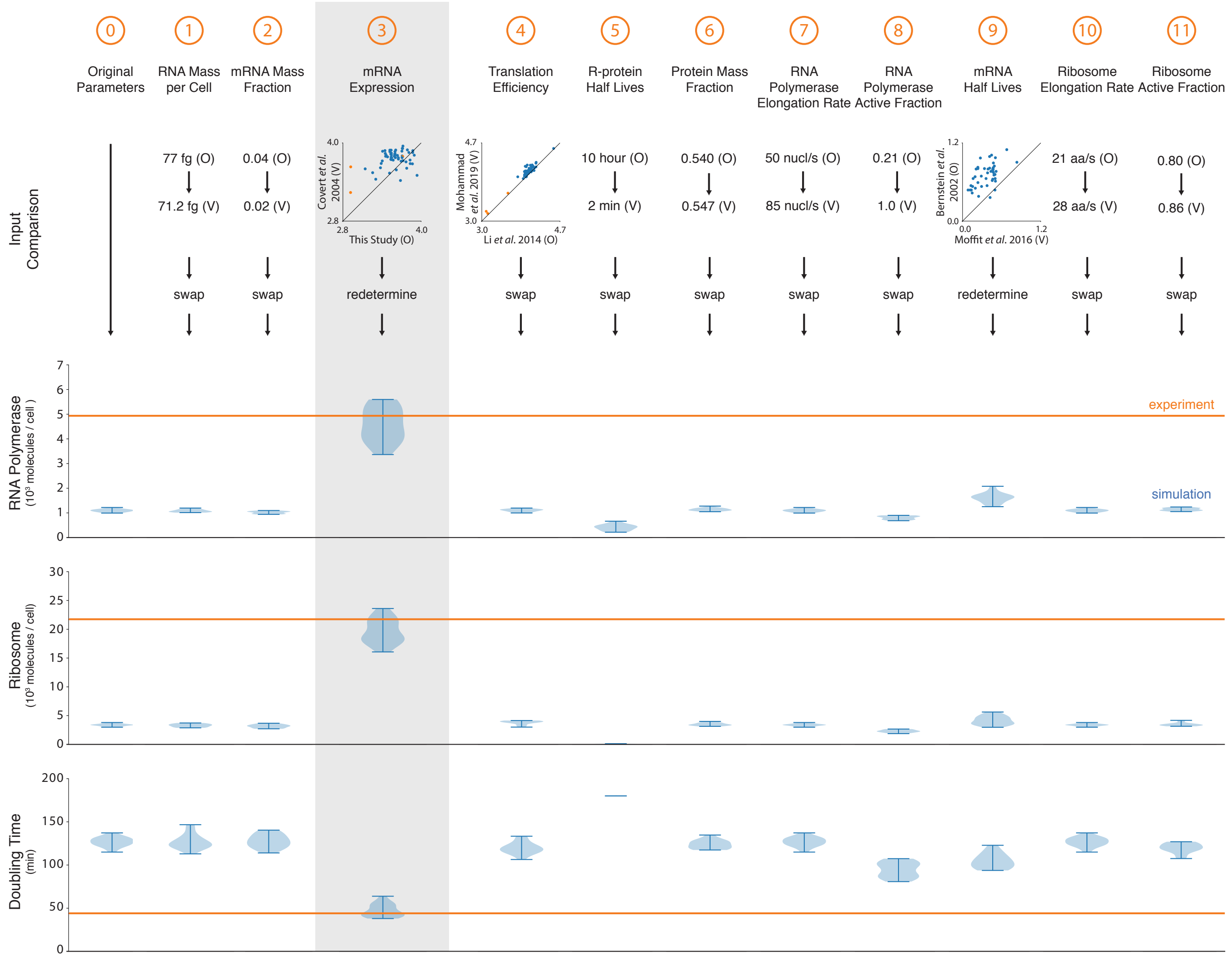
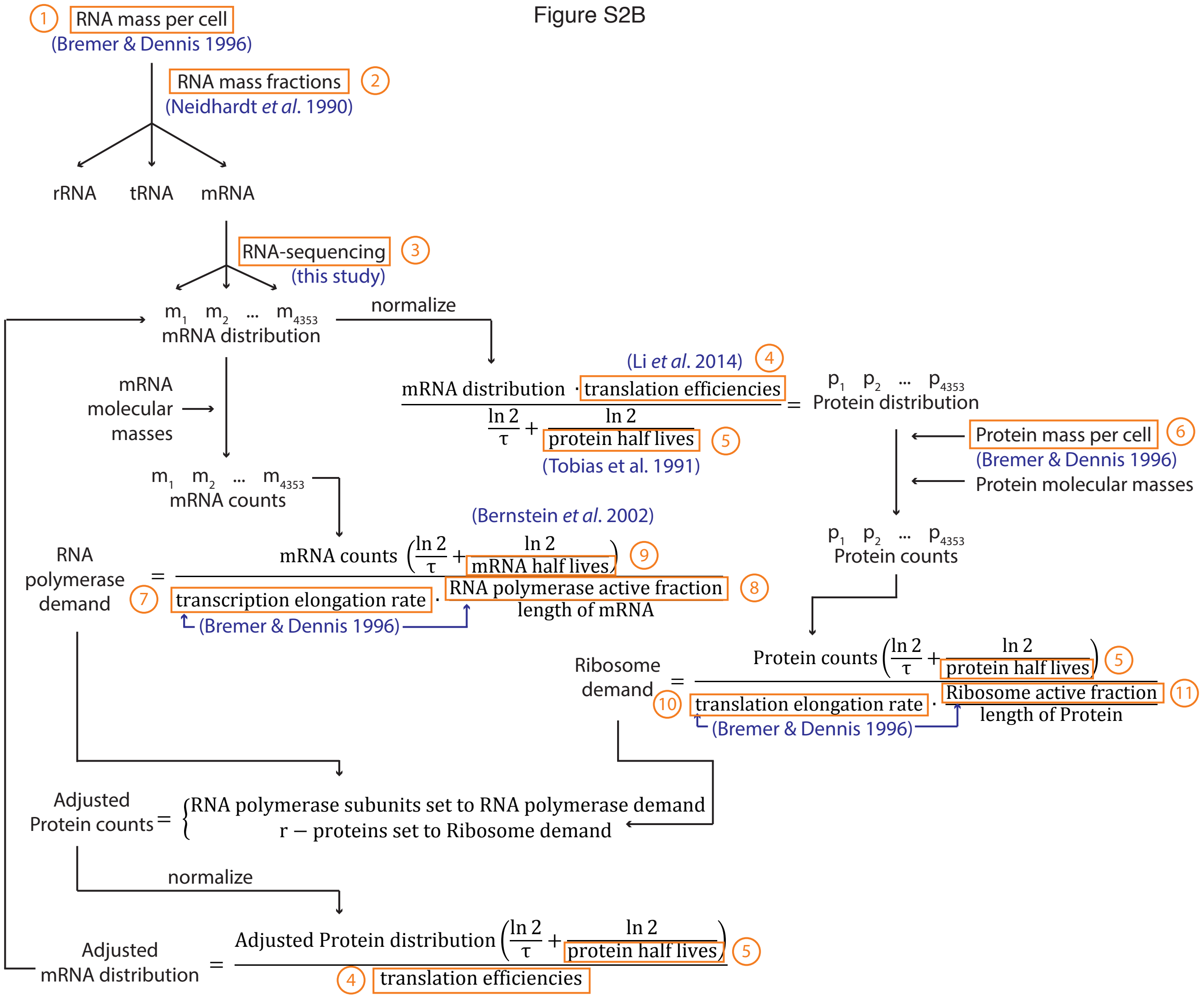

0
Original Parameters
1
RNA Mass per Cell
2
mRNA Mass Fraction
3
mRNA Expression
4
Translation Efficiency
5
R-protein Half Lives
6
Protein Mass Fraction
7
RNA Polymerase Elongation Rate
8
RNA Polymerase Active Fraction
9
mRNA Half Lives
10
Ribosome Elongation Rate
11
Ribosome Active Fraction
Input Comparison
77 fg (O)
71.2 fg (V)
0.04 (O)
0.02 (V)
Covert et al. 2004 (V)
This Study (O)
4.0
2.8
Mohammad et al. 2019 (V)
Li et al. 2014 (O)
4.7
3.0
10 hour (O)
2 min (V)
0.540 (O)
0.547 (V)
50 nucl/s (O)
85 nucl/s (V)
0.21 (O)
1.0 (V)
Bernstein et al. 2002 (O)
Moffit et al. 2016 (V)
1.2
0.0
21 aa/s (O)
28 aa/s (V)
0.80 (O)
0.86 (V)
swap
redetermine
RNA Polymerase
(10³ molecules / cell)
0
1
2
3
4
5
6
7
experiment
simulation
Ribosome
(10³ molecules / cell)
0
5
10
15
20
25
30
Doubling Time
(min)
0
50
100
150
200

Fig. S2A. Parameter variability investigation. Parameters identified in Fig. S2B were investigated for their potential impact on RNA Polymerase and Ribosome abundances, and cell doubling time. The "Input Comparison" row shows the original data (O) used in this study and the validation data (V). Data sources are listed in Table S2A. The data shown for mRNA expression (rna-sequencing), translation efficiency (ribosome density), and mRNA half lives (minutes) are log base 10 transformed values compared on linear axes. Ribosomal proteins (blue) and RNA Polymerase subunits (orange) are highlighted. Parameter values that agreed well with the validation data were swapped with the validation data. Parameter values that did not agree well with the validation data justified a redetermination opportunity which was performed via iterative parameter estimation programs (described in Section 1.2.2) that adjust the mRNA expression (in column three) or mRNA half lives (in column nine) of RNA polymerase subunits and ribosomal proteins. After either a "swap" or "redetermine", 32 cells (8 seeds, 4 generations) were simulated for each column. The abundances of total RNA Polymerase per cell, total Ribosome per cell, and doubling time were identified from the simulations and are depicted as violin plots. As in Fig. 2A (in the main text), the third and fourth generations participate in this analysis. The expected values were determined from interpolation of the per-cell abundances of RNA Polymerase and Ribosome across doubling times (*7*), and are depicted as orange lines. Interpolation was performed by the interpolate.splrep and interpolate.splev programs from SciPy. The numbers circled in orange correspond to the parameter number annotated in Fig. S2B and Table S2A. Full details of the analysis required to generate this figure, as well as a pointer to the generating code, can be found in Section 1.2.

Figure S2B



1 RNA mass per cell
(Bremer & Dennis 1996)
RNA mass fractions 2
(Neidhardt *et al*. 1990)
rRNA
tRNA
mRNA
RNA-sequencing 3
(this study)
$m_1 \quad m_2 \quad \dots \quad m_{4353}$
mRNA distribution
normalize
mRNA molecular masses
$m_1 \quad m_2 \quad \dots \quad m_{4353}$
mRNA counts
(Li *et al*. 2014) 4
$\dfrac{\text{mRNA distribution} \cdot \boxed{\text{translation efficiencies}}}{\dfrac{\ln 2}{\tau} + \dfrac{\ln 2}{\boxed{\text{protein half lives}}}} =$
5
(Tobias et al. 1991)
$p_1 \quad p_2 \quad \dots \quad p_{4353}$
Protein distribution
Protein mass per cell 6
(Bremer & Dennis 1996)
Protein molecular masses
$p_1 \quad p_2 \quad \dots \quad p_{4353}$
Protein counts
(Bernstein *et al*. 2002)
RNA polymerase demand $= \dfrac{\text{mRNA counts}\left(\dfrac{\ln 2}{\tau} + \dfrac{\ln 2}{\boxed{\text{mRNA half lives}}}\right)}{\boxed{\text{transcription elongation rate}} \cdot \dfrac{\boxed{\text{RNA polymerase active fraction}}}{\text{length of mRNA}}}$
9
7
8
(Bremer & Dennis 1996)
Ribosome demand $= \dfrac{\text{Protein counts}\left(\dfrac{\ln 2}{\tau} + \dfrac{\ln 2}{\boxed{\text{protein half lives}}}\right)}{\boxed{\text{translation elongation rate}} \cdot \dfrac{\boxed{\text{Ribosome active fraction}}}{\text{length of Protein}}}$
5
10
11
(Bremer & Dennis 1996)
Adjusted Protein counts $= \begin{cases} \text{RNA polymerase subunits set to RNA polymerase demand} \\ \text{r} - \text{proteins set to Ribosome demand} \end{cases}$
normalize
Adjusted mRNA distribution $= \dfrac{\text{Adjusted Protein distribution}\left(\dfrac{\ln 2}{\tau} + \dfrac{\ln 2}{\boxed{\text{protein half lives}}}\right)}{\boxed{\text{translation efficiencies}}}$
5
4

Fig. S2B. Iterative parameter estimation approach. Flow chart depicts all parameters participating in the iterative parameter estimation approach described by Equations 59 through 64 and used to generate the cell state in Fig. 2D. A measurement of the total RNA mass per cell is partitioned into masses for each RNA type utilizing RNA mass fraction data. The total mRNA mass is further partitioned into the masses of each (gene-specific) mRNA type using RNA-sequencing data. From here, the flow chart splits into two branches: RNA polymerase demand determination (down) and Ribosome demand determination (right). **RNA polymerase branch**: The number of molecules of each mRNA type (one corresponding to each of the 4353 genes) is determined from the mRNA mass distribution and the molecular masses of each mRNA type according to Equation 2. Then, the number of RNA polymerase molecules required to sustain steady-state growth (ie. doubling of all RNA molecules) is determined from Equation 59. **Ribosome branch**: The protein distribution is determined from the normalized mRNA distribution, translation efficiencies, protein half lives, and doubling time as described in Algorithm 2, Step 3. The number of molecules of each protein monomer (one corresponding to each of the 4353 transcripts) is determined from the protein distribution, the molecular masses of each protein monomer, and the total mass of protein per cell according to Equation 2. Then, the number of ribosome molecules required to sustain steady-state growth (ie. doubling of all protein molecules) is determined from Equation 59. **Adjusted protein counts**: The numbers of RNA polymerase and ribosomes determined in the previous two branches is converted to numbers of RNA polymerase subunits and ribosomal proteins using the stoichiometry of the RNA polymerase and ribosome. The number of molecules of each of these protein monomers is then updated in the "Protein counts" array determined previously in the "Ribosome branch". The updated "Protein counts" are then normalized and projected back to the mRNA distribution according to the equation presented in Algorithm 2, Step 3 - rearranged to solve for the mRNA distribution. This mRNA distribution replaces that determined previously, and iterations continue until the mRNA distribution converges according to the terminal condition described by Equation 64. Parameters indicated in orange were identified as potential sources of data variability and investigated further in Fig. S2A. The numbers circled in orange correspond to the parameter number annotated in Fig. S2A and Table S2A. Original data sources are annotated in blue, and correspond to the "Original Data Source" column in Table S2A. Full details of the analysis required to generate this figure, as well as a pointer to the generating code, can be found in Section 1.2.

Figure S2 (cont.)

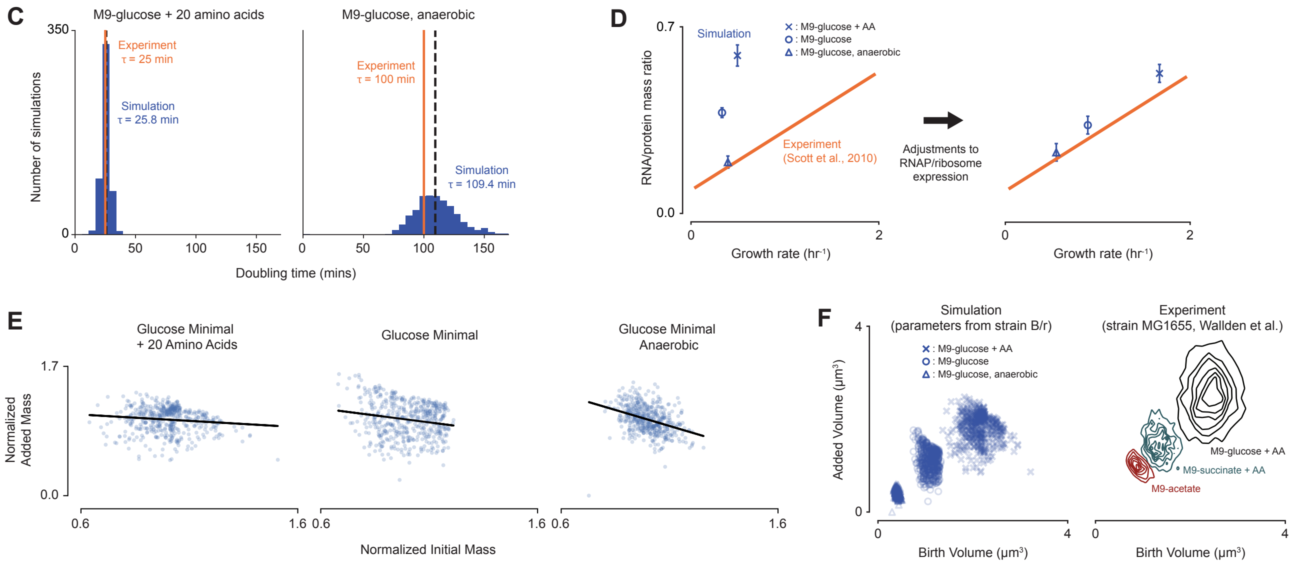


Fig. S2. (continued) (C) Histograms comparing simulated doubling times (blue) to the experimentally determined doubling time (orange line) for aerobic growth in glucose minimal media supplemented with 20 amino acids (left), and anaerobic growth in glucose minimal media (right). Median simulated doubling times are reported (dashed black lines). Both histograms are generated from simulation data using RNA polymerase and ribosome expression calculated from the known doubling time. (D) Comparison of the relationship between the RNA/protein mass ratio and the growth rate predicted by our simulations under the three simulated environments (blue), and the linear relationship that was experimentally determined by Scott et al., 2010 (*32*) (orange). Before adjusting the expression of RNA polymerases and ribosomes (left), our simulations do not agree well with experimental measurements; after the adjustment is made (right), the values from our simulations fall closer to the proposed linear relationship. (E) Comparisons of the normed initial and added mass indicates that the simulations generate adder behavior in the glucose minimal media plus amino acid condition (slope = -0.17, $R^2 = 0.014$, $p = 6.5 \times 10^{-3}$), and sizer behavior in glucose minimal and anaerobic glucose minimal conditions (slope = -0.37, $R^2 = 0.036, p = 2\times10^{-5}$ and slope = -0.84, $R^2 = 0.14, p = 2.3\times10^{-18}$, respectively). These results match observations by recent studies (*4, 5, 33, 35*). (F) Comparison of the distribution of birth volumes and added volumes predicted by our simulations and the experimental distributions reported by Wallden et al., 2016 (*33*). The simulations are able to qualitatively capture the adder-sizer behavior observed in the experiments, but the quantitative values for the cell volumes do not compare well to the experiments due to differences in strains and growth environments. Wallden et al. used strain MG1655 for their measurements, whereas our model calculates cell volumes based on experimental parameters measured for strain B/r. The media conditions used for fast, intermediate, and slow growth in Wallden et al. (M9-glucose + amino acids, M9-succinate + amino acids, and M9-acetate, respectively) also do not fully match the conditions we used in our model (M9-glucose + amino acids, M9-glucose, M9-glucose anaerobic). Full details of the analysis required to generate this figure, as well as a pointer to the generating code, can be found in Section 1.2.

Figure S3

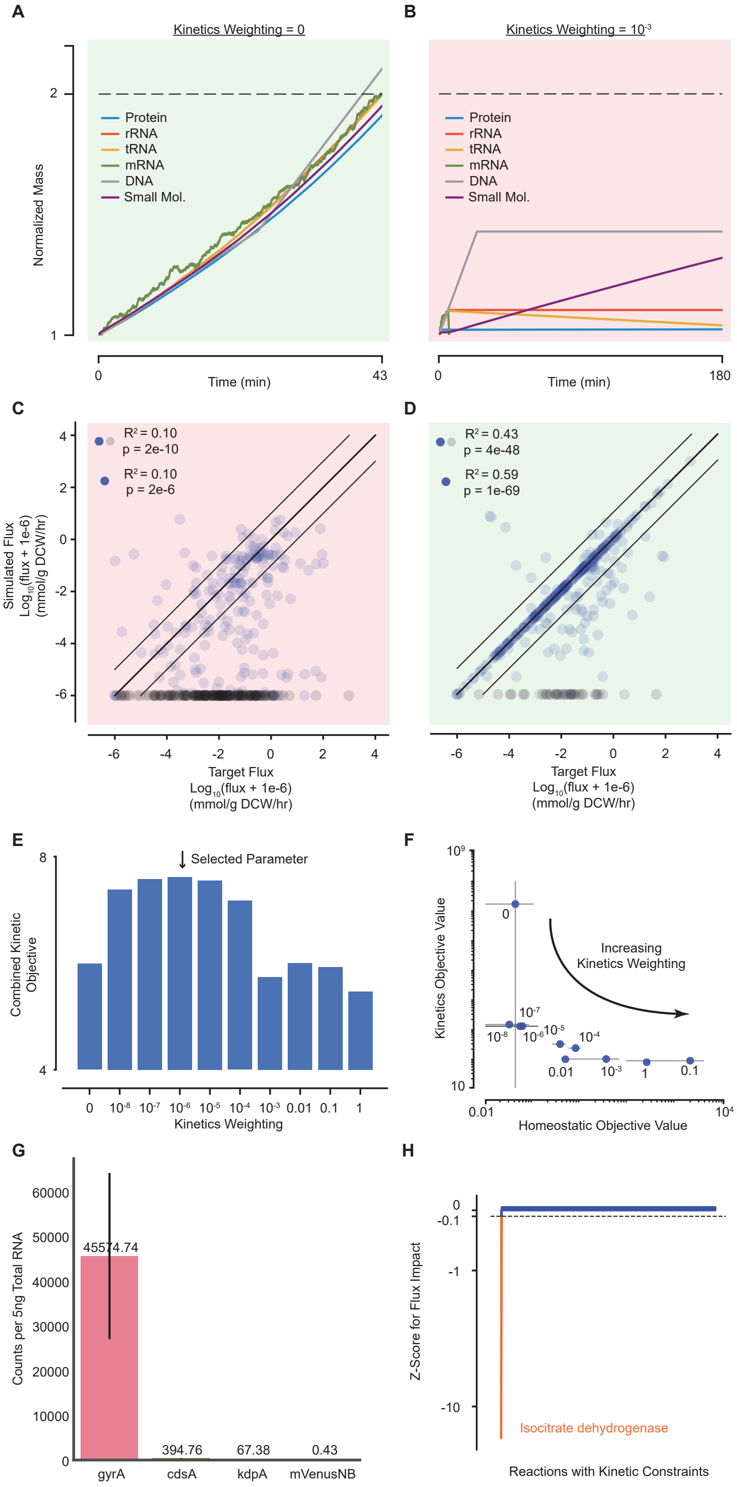

A
Kinetics Weighting = 0
Normalized Mass
2
1
Protein
rRNA
tRNA
mRNA
DNA
Small Mol.
0
Time (min)
43
B
Kinetics Weighting = 10-3
Protein
rRNA
tRNA
mRNA
DNA
Small Mol.
0
Time (min)
180
C
Simulated Flux
Log10(flux + 1e-6)
(mmol/g DCW/hr)
R2 = 0.10
p = 2e-10
R2 = 0.10
p = 2e-6
4
2
0
-2
-4
-6
-6
-4
-2
0
2
4
Target Flux
Log10(flux + 1e-6)
(mmol/g DCW/hr)
D
R2 = 0.43
p = 4e-48
R2 = 0.59
p = 1e-69
-6
-4
-2
0
2
4
Target Flux
Log10(flux + 1e-6)
(mmol/g DCW/hr)
E
Combined Kinetic Objective
8
4
Selected Parameter
0
10-8
10-7
10-6
10-5
10-4
10-3
0.01
0.1
1
Kinetics Weighting
F
Kinetics Objective Value
10^9
10
0
10-7
10-8
10-6
10-5
10-4
0.01
10-3
1
0.1
Increasing
Kinetics Weighting
0.01
10^4
Homeostatic Objective Value
G
Counts per 5ng Total RNA
60000
50000
40000
30000
20000
10000
0
45574.74
394.76
67.38
0.43
gyrA
cdsA
kdpA
mVenusNB
H
Z-Score for Flux Impact
0
-0.1
-1
-10
Isocitrate dehydrogenase
Reactions with Kinetic Constraints

Fig. S3. Analysis of the tradeoff across different kinetics weighting parameters. (A and B) Representative output from simulations showing the increase in mass (normalized to initial mass and over a single life cycle) of six key cellular mass fractions for kinetics weighting of 0 (no kinetics objective) in (A) and for kinetics weighting of 10^{-3} (highly weighted kinetic objective) in (B). Output with kinetics weighting of 10^{-6} (selected parameter) is shown in Fig. 3I. The green background indicates desired behavior (balanced growth) similar to results for the selected parameter (Fig. 3I), while the red background indicates undesired behavior (lack of growth). (C and D) Comparison between the metabolic fluxes calculated directly from the kinetic parameters (target) and the fluxes computed by simulations with the new set of disabled constraints, as summarized by the R^2 value. Gray points correspond to reactions with no simulated flux despite having a target flux. Correlations are shown for all data points (blue and gray) and with gray points excluded (blue only). Correlation for kinetics weighting of 0 (no kinetics objective) is shown in (C), for kinetics weighting of 10^{-3} (highly weighted kinetic objective) in (D) and for kinetics weighting of 10^{-6} (selected parameter) in Fig. 3J. The green background indicates desired behavior (few simulated reactions with 0 flux) similar to results for the selected parameter (Fig. 3J), while the red background indicates undesired behavior (high number of simulated reactions with 0 flux). (E) Objective function used to select the kinetics weighting parameter for all weights tested. It combines desired performance measures into one value to maximize. (F) Tradeoff between the average kinetics objective value and the average homeostatic objective value at different kinetics weightings. Points are labeled with their kinetics weighting. (G) Average counts from qPCR for selected RNA showing that cdsA is present. mVenusNB is not present in the cell and is used to determine the lower limit of detection. (H) Impact of individually disabling each kinetic reaction constraint on the isocitrate dehydrogenase flux in simulations, shown as a z-score representing the average change in flux for removing one constraint compared to the distribution of the average change in flux for removing each constraint. Only disabling the constraint for isocitrate dehydrogenase results in a z-score of $<$-0.1. This is a similar analysis as in Fig. 3D but with the impact on isocitrate dehydrogenase flux instead of succinate dehydrogenase flux. Full details of the analysis required to generate this figure, as well as a pointer to the generating code, can be found in Section 1.2.

Figure S4

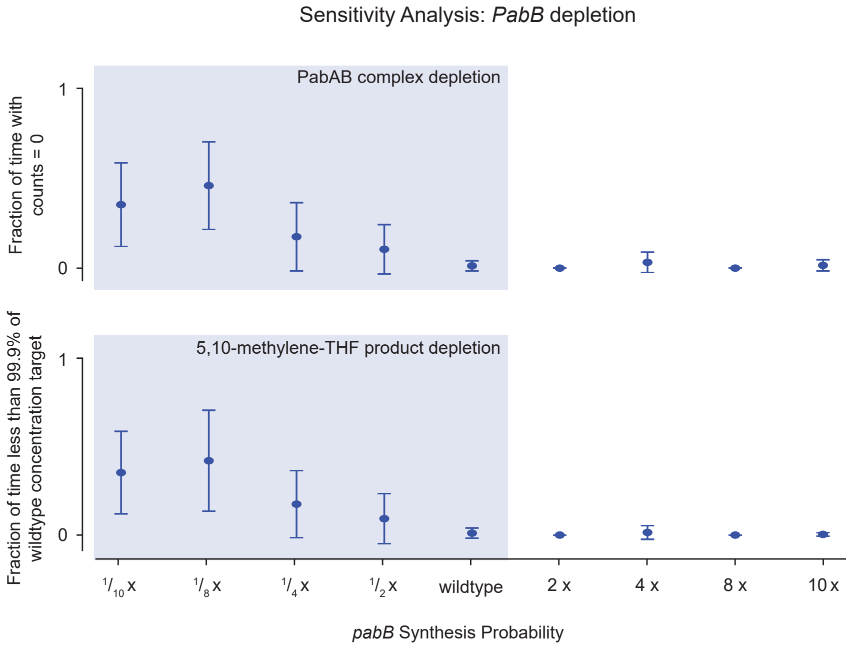
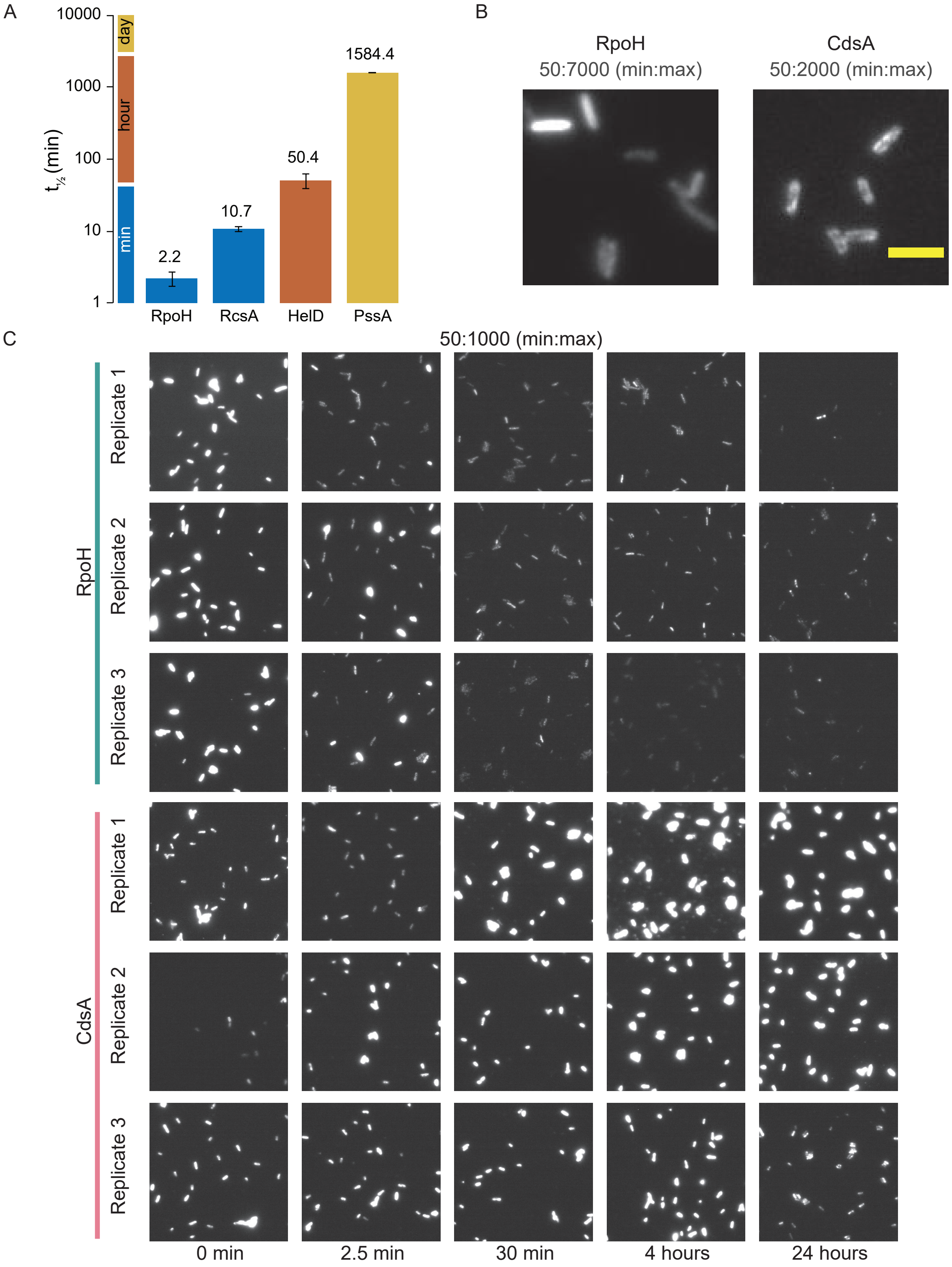


Fig. S4. PabB sensitivity analysis. Sensitivity analysis of the *pabB* transcript synthesis probability was performed by a parameter sweep spanning an order of magnitude above and below the wild-type value. Each of the 9 factors tested (shown along the x-axis) was assessed by 16-generation long serial simulations starting from 8 initial cells (n = 128). Average fraction of time with zero counts of PabAB enzyme complex, and average fraction of time with 5,10-methylene-THF below 99.9% of the wild-type target concentration are shown with error bars representing their standard deviations. Shaded regions represent PabAB and 5,10-methylene-THF depletion. Full details of the analysis required to generate this figure, as well as a pointer to the generating code, can be found in Section 1.2.

Figure S5


A
10000
1000
100
10
1
t½ (min)
day
hour
min
2.2
10.7
50.4
1584.4
RpoH
RcsA
HelD
PssA
B
RpoH
50:7000 (min:max)
CdsA
50:2000 (min:max)
C
50:1000 (min:max)
RpoH
Replicate 1
Replicate 2
Replicate 3
CdsA
Replicate 1
Replicate 2
Replicate 3
0 min
2.5 min
30 min
4 hours
24 hours

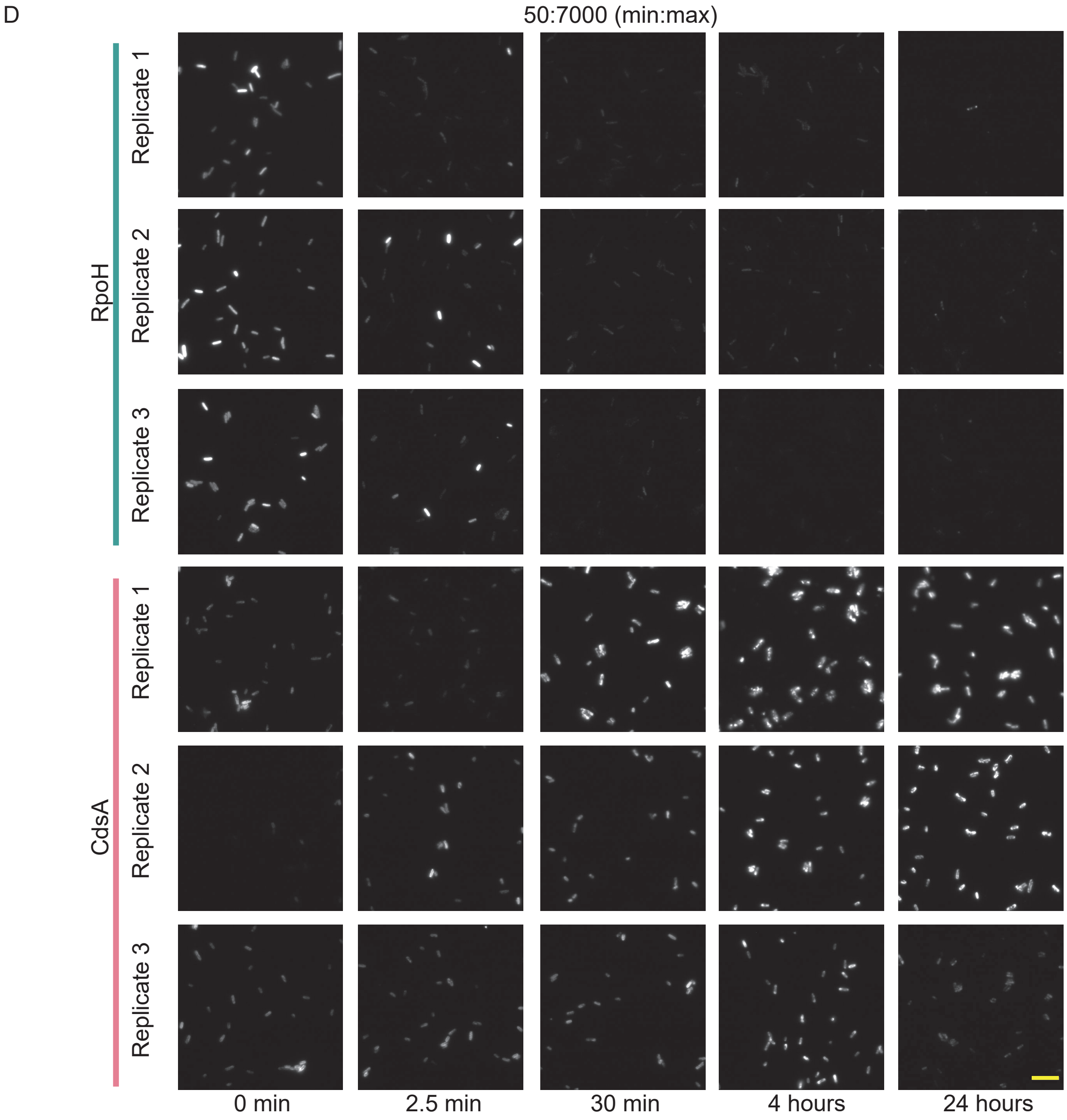

D
50:7000 (min:max)
RpoH
Replicate 1
Replicate 2
Replicate 3
CdsA
Replicate 1
Replicate 2
Replicate 3
0 min
2.5 min
30 min
4 hours
24 hours

Fig. S5. Additional data for protein half life determination (A) We tested four protein half-lives beyond the seven shown in Fig. 5 as controls. First, the half-lives of RpoH and RcsA had previously been reported (*42, 43*) as on the order of minutes, and were also confirmed as such in our measurements. Second, the N-end rules suggests that the half-lives of HelD and PssA should be on the order of hours, and the balance between protein production and loss for these two proteins is shown to be almost perfect in Fig. 5A. We measured half-lives for these proteins to confirm the longer half-lives. (B) Images of RpoH and CdsA prior to addition of tetracycline show cytoplasmic and membrane staining, respectively. Scale bar (yellow) = 5 μm. (C and D) His-tagged RpoH (green panels) or CdsA (pink panels) expression was induced for 1 hour using IPTG followed by the addition of tetracycline to inhibit translation. At the indicated timepoints, aliquots of the culture were harvested, immunofluorescence was carried out using an anti-His antibody, and samples were imaged. Duplicates of the images are shown scaled between (C) 50-1000 AU and (D) 50-7000 AU. His-RpoH protein signal decreased within minutes, while His-CdsA protein signal was maintained or increased over the timecourse. Three independent replicates are shown for each protein. Scale bar (yellow) = 10 μm.

3 Supplemental Tables

Table S1. Estimate of the number of parameters in the model, grouped by category. We are often asked how many parameters are contained our models. Estimating this as a simple number belies the complex and heterogeneous nature of the model, where each sub-model is specified using a different mathematical formalism. At one extreme, every piece of data could be considered a parameter (e.g., every single nucleotide in the chromosome sequence), but we feel this isn't a helpful estimate. Likewise, the stoichiometric coefficients in the metabolic network, molecular masses of each mRNA and protein species, as well as all of the data in our RNA expression databases, could all be treated as parameters—but we don't take them into account when answering this question. While one could therefore count parameters in many different ways, Table S1 provides the breakdown for the over nineteen thousand parameters that we state in the main text.

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

Table S2A. Data sources for parameter investigation in Fig. S2A. Data sources for the parameters identified and investigated in Figures S2A and S2B. The values reported in the "Original Data Source" column are swapped for the values reported in the "Validation Data Source" column only for the purpose of the analysis depicted in Fig. S2A. The values reported in the "Original Data Source" column were used in the model for the remainder of this study, including the iterative parameter estimation approach described in Fig. S2B. The parameter number (first column) corresponds to the orange numbering in Figures S2A and S2B.

Parameter Number	Parameter Name	Original Data Source (used in This Study)	Validation Data Source
1	RNA Mass per Cell	Bremer & Dennis 1996 (*31*)	Pang & Winkler 1994 (*44*)
2	mRNA Mass Fraction	Neidhardt et al. 1990 (*27*)	Kennel 1968 (*45*)
3	mRNA Expression	This Study	Covert et al. 2004 (*46*)
4	Translation Efficiency	Li et al. 2014 (*10*)	Mohammad et al. 2019 (*47*)
5	R-Protein Half Lives	Tobias et al. 1991 (*12*)	Computationally Investigated
6	Protein Mass Fraction	Bremer & Dennis 1996 (*31*)	Beck et al. 2018 (*48*)
7	RNA Polymerase Elongation Rate	Bremer & Dennis 1996 (*31*)	Computationally Investigated
8	RNA Polymerase Active Fraction	Bremer & Dennis 1996 (*31*)	Computationally Investigated
9	mRNA Half Lives	Bernstein et al. 2002 (*8*)	Moffitt et al. 2016 (*49*)
10	Ribosome Elongation Rate	Bremer & Dennis 1996 (*31*)	Pedersen 1984 (*50*)
11	Ribosome Active Fraction	Bremer & Dennis 1996 (*31*)	Dai et al. 2016 (*51*)

Table S2B. The fold-changes in transcription initiation probabilities for all genes, following the described parameter estimation procedure for both RNA polymerase and ribosome expression, under the glucose-minimal growth condition. The changes reported in this table are the only changes between the "Original" and "New" columns in Fig. 2E. See Section 1.2.2 for a description of how these parameters were determined. This table is also included in the GitHub repository: wcEcoli/paper/synthesis_probability_fold_change.tsv.

Cell component	Gene(s)	Fold change
ribosomal protein	*rplY*	0.890
ribosomal protein	*rpmA*	0.890
ribosomal protein	*rpmF,*	0.890
ribosomal protein	*rpmI,*	0.890
ribosomal protein	*rpsF*	0.890
ribosomal protein	*rpsR*	0.890
ribosomal protein	*rpsU*	0.890
ribosomal protein	*rpsN*	0.925
ribosomal protein	*rpmB*	0.930
ribosomal protein	*rpsJ*	0.943
ribosomal protein	*rpsO*	0.965
ribosomal protein	*rpmC*	0.990
ribosomal protein	*rpsT*	0.996
ribosomal protein	*rpsB*	1.026
ribosomal protein	*rpmE*	1.040
ribosomal protein	*rpmG*	1.083
ribosomal protein	*rplJ*	1.155
ribosomal protein	*rplP*	1.174
ribosomal protein	*rplT*	1.185
ribosomal protein	*rpsP*	1.221
ribosomal protein	*rplQ*	1.230
ribosomal protein	*rpsA*	1.244
ribosomal protein	*rpsI*	1.249
ribosomal protein	*rplW*	1.269
ribosomal protein	*rplA*	1.298
ribosomal protein	*rpmJ*	1.319
ribosomal protein	*rpsG*	1.334
ribosomal protein	*rplL*	1.357
ribosomal protein	*rplV*	1.394
ribosomal protein	*rplM*	1.403
ribosomal protein	*rpsS*	1.420
ribosomal protein	*rplC*	1.489
ribosomal protein	*rplU*	1.505
ribosomal protein	*rpsD*	1.549
ribosomal protein	*rpsL*	1.567
ribosomal protein	*rpsQ*	1.666
ribosomal protein	*rplS*	1.689
ribosomal protein	*rplK*	1.699
ribosomal protein	*sra*	1.732
ribosomal protein	*rpsK*	1.782
ribosomal protein	*rpsC*	1.857
ribosomal protein	*rplE*	1.913
ribosomal protein	*rplI*	1.986
ribosomal protein	*rplF*	2.136
ribosomal protein	*rplD*	2.155
ribosomal protein	*rplB*	2.160
ribosomal protein	*rpsM*	2.270
ribosomal protein	*rpmH*	2.272
ribosomal protein	*rplO*	2.480
ribosomal protein	*rpsE*	3.010
ribosomal protein	*rplN*	3.017
ribosomal protein	*rplR*	3.018
ribosomal protein	*rpsH*	3.317
ribosomal protein	*rplX*	3.347
ribosomal protein	*rpmD*	3.525
RNA polymerase	*rpoA*	1.806
RNA polymerase	*rpoB*	2.123
RNA polymerase	*rpoC*	2.186
rRNA	*rrfA, rrlA, rrsA*	0.984
tRNA	*86 genes*	0.984
Misc.	*4230 genes*	0.890

Table S3. Parameters that were modified from their initial values as part of the model fitting process. Two types of manual adjustments were made for certain model parameters. The first was discussed in the main text and is indicated as “Kinetic reaction k_{cat} and K_M” in the adjustments column. The reason for these adjustments is either “Unknown regulation”, which refers to complex regulatory control of the enzymes that is not fully understood, or “Flux validation”, which refers to our observation that the parameter values related to these reactions forced a non-physiological glucose uptake rate and some reaction fluxes that were inconsistent with a validation dataset. Secondly, we do not examine the anaerobic condition in great detail in the main text, but two additional reactions were found (similar to cdsA, described in the main text) to impact simulations negatively due to low expression in anaerobic simulations. For the enzymes catalyzing these reactions, the adjustments column refers to the parameter adjustments that were made where “RNA expression” refers to a factor of 10 increase, “Translation efficiency” refers to a factor of 5 increase, and “Protein degradation rate” refers to a tenfold decrease. These enzymes are indicated by “Metabolite production” in the reason column.

Gene	Gene Name	Adjustments	Reason
ghrB	Glyoxylate reductase	Kinetic reaction k_{cat}	Unknown regulation
icd	Isocitrate dehydrogenase	Kinetic reaction k_{cat} and K_M	Unknown regulation
gor	Glutathione reductase	Kinetic reaction k_{cat}	Flux validation
ppa	Inorganic pyrophosphatase	Kinetic reaction k_{cat} and K_M	Flux validation
nuoABCEF GHIJKLMN	NADH:quinone oxidoreductase	Kinetic reaction k_{cat} and K_M	Flux validation
sdhABCD	Succinate:quinone oxidoreductase	Kinetic reaction k_{cat} and K_M	Flux validation
yibQ	Polysaccharide deacetylase domain-containing protein	RNA expression Translation efficiency Protein degradation rate	Metabolite production
atoB	Acetyl-CoA acetyltransferase	RNA expression Translation efficiency	Metabolite production

Table S4. List of protein products of essential genes that experience a period in which the corresponding protein is absent due to sub-generational gene expression, as indicated in Fig. 4E. Protein products of essential genes with a count of zero at least once during the 32 generation simulation of log-phase growth under aerobic glucose minimal media conditions. Shown below are the 23 protein products of 23 essential genes that fall into this category.

Gene	Description	Protein product(s) whose counts were zero at least once
lpxB	lipid A disaccharide synthase	CPLX0-7415[i]
metR	DNA-binding transcriptional dual regulator MetR	CPLX0-7759[c]
mraY	phospho-N-acetylmuramoyl-pentapeptide-transferase	PHOSNACMURPENTATRANS-MONOMER[i]
pabA	aminodeoxychorismate synthase subunit 2	PABSYNMULTI-CPLX[c] PABASYN-CPLX[c]
pabB	aminodeoxychorismate synthase subunit 1	PABSYNMULTI-CPLX[c] PABASYN-CPLX[c]
pyrE	orotate phosphoribosyltransferase	OROPRIBTRANS-CPLX[c]
tnaB	tryptophan:H+ symporter TnaB	TNAB-MONOMER[i]
tsaD	N6-L-threonylcarbamoyladenine synthase, TsaD subunit	CPLX0-8181[c]
ydfB	Qin prophage; uncharacterized protein YdfB	EG11301-MONOMER[c]
wzyE	putative enterobacterial common antigen polymerase	CPLX0-3976[i]
yibJ	putative RHS domain-containing protein YibJ	EG11766-MONOMER[c]
alsK	D-allose kinase	EG11956-MONOMER[c]
mukF	MukF dimer	CPLX0-7697[c] CPLX0-7698[c]
yhhQ	putative queuosine precursor transporter	EG12217-MONOMER[i]
bcsB	cellulose synthase periplasmic subunit	CPLX0-8125[i]
yafF	putative uncharacterized protein YafF	MONOMER0-2653[c]
ydiL	DUF1870 domain-containing protein YdiL	G6915-MONOMER[c]
tsaB	N6-L-threonylcarbamoyladenine synthase, TsaB subunit	CPLX0-8181[c]
yqgD	DUF2684 domain-containing protein YqgD	G7523-MONOMER[c]
mqsA	DNA-binding transcriptional repressor MqsA	CPLX0-7822[c]
tdcF	predicted enamine/imine deaminase	CPLX0-7987[c]
yhbV	ubiquinone biosynthesis protein UbiV	G7653-MONOMER[c]
lptF	lipopolysaccharide transport system protein LptF	ABC-53-CPLX[i]

Table S5. Experimental details of the half-life measurements. $t_{1/2}$ is the expected half-life prior to the experiment. We had two sets of control proteins for which the half-life was determined: (*) proteins with known and relatively short half-lives (*42, 43*), used as a control for our experimental technique; and (**) proteins in which the protein production and loss rates predicted by the model are nearly equal (see Fig. 5A), and which therefore are likely to have a 10 hour half-life, also used as a control. The test proteins are also listed, with (***) denoting the short half-life proteins as predicted by the N-end rule. [1]IPTG concentration used to over-express protein levels. [2]Amount of protein loaded for protein detection. [3]Antibody dilution in anti-His, anti-RNAP. [4]The time points highlighted in bold were used to calculate the decay constant.

Gene	ASKA ID	$t_{1/2}$	Ind.[1] (h)	Protein[2] (μg)	Ab. dilution[3]	Time points[4]
RcsA	JW1935	2-5 min (c*)	2	10	1:100, 1:200	**0, 2, 5, 10, 30** min
RpoH	JW3426	2-5 min (c*)	2	10	1:100, 1:200	**0, 5, 10, 20**, 40 min
PssA	JW2569	10 h (c**)	2	1	1:100, 1:50	**0, 7, 18** h
HelD	JW0945	10 h (c**)	2	4	1:100, 1:100	**0, 1**, 3, 4, 7, 23 h
CarA	JW0030	2 min ***	2	2	1:100, 1:100	**0, 0.17, 1, 4, 18** h
GshA	JW2663	2 min ***	1	1	1:100, 1:100	**0, 0.17, 1, 4, 18** h
Pnp	JW5851	2 min ***	1	4	1:100, 1:100	**0, 18, 24, 48** h
DcuR	JW4085	10 h	2	2	1:100, 1:100	**0, 0.5, 2.5**, 5.5, 18 h
BioD	JW0761	10 h	2	4	1:100, 1:100	**0, 0.67, 2**, 4, 18 h
Rph	JW3618	10 h	2	4	1:100, 1:100	**0, 0.5, 2.5**, 5.5, 18 h

Table S6. Genes that are functionally incorporated in the model and their model related function. All other genes are transcribed and translated but play no further role in the model. The table is also included in the GitHub repository: `wcEcoli/paper/functional_genes.tsv`.

Gene	Function
aas	Metabolism
aat	Metabolism
accA	Metabolism
accC	Metabolism
accD	Metabolism
aceE	Metabolism
aceF	Metabolism
ackA	Metabolism
acnA	Metabolism
acnB	Metabolism
acpH	Metabolism
acpS	Metabolism
acpT	Metabolism
acrA	Metabolism
acrB	Metabolism
acs	Metabolism
add	Metabolism
ade	Metabolism
adhE	Metabolism
adhP	Metabolism
adiA	Metabolism
adk	Metabolism
ahr	Metabolism
aidB	Metabolism
alaA	Metabolism
alaC	Metabolism
alaE	Metabolism
alaS	Metabolism
aldA	Metabolism
aldB	Metabolism
allB	Metabolism
allC	Metabolism
allE	Metabolism
alr	Metabolism
alsA	Metabolism
alsB	Metabolism
alsC	Metabolism
alsE	Metabolism
alsK	Metabolism
amn	Metabolism
amtB	Metabolism
ansA	Metabolism
ansB	Metabolism
ansP	Metabolism
apaH	Metabolism
aphA	Metabolism
appA	Metabolism
appB	Metabolism
appC	Metabolism
apt	Metabolism
aqpZ	Metabolism
araA	Metabolism
araB	Metabolism
araC	Transcription Regulation
araD	Metabolism
araE	Metabolism
araF	Metabolism
araG	Metabolism
araH	Metabolism
arcA	Transcription Regulation
argA	Metabolism
argB	Metabolism
argC	Metabolism
argD	Metabolism
argE	Metabolism
argF	Metabolism
argG	Metabolism
argH	Metabolism
argI	Metabolism
argP	Transcription Regulation
argR	Transcription Regulation
argS	Metabolism
argT	Metabolism
arnA	Metabolism
arnB	Metabolism
aroA	Metabolism
aroB	Metabolism
aroC	Metabolism
aroD	Metabolism
aroE	Metabolism
aroF	Metabolism
aroG	Metabolism
aroH	Metabolism
aroK	Metabolism
aroL	Metabolism
aroP	Metabolism
arsB	Metabolism
arsC	Metabolism
artJ	Metabolism
artM	Metabolism
artP	Metabolism
artQ	Metabolism
asd	Metabolism
asnA	Metabolism
asnB	Metabolism
asnS	Metabolism
aspA	Metabolism
aspC	Metabolism
aspS	Metabolism
astA	Metabolism
astB	Metabolism
astC	Metabolism
astD	Metabolism
astE	Metabolism
atoA	Metabolism
atoB	Metabolism
atoD	Metabolism
atpA	Metabolism
atpB	Metabolism
atpC	Metabolism
atpD	Metabolism
atpE	Metabolism
atpF	Metabolism
atpG	Metabolism
atpH	Metabolism
avtA	Metabolism
azoR	Metabolism
baeR	Transcription Regulation
basR	Transcription Regulation
betA	Metabolism
betB	Metabolism
betT	Metabolism
bglF	Metabolism
bglJ	Transcription Regulation
bioA	Metabolism
bioB	Metabolism
bioC	Metabolism
bioD	Metabolism
bioF	Metabolism
bioH	Metabolism
birA	Metabolism
bisC	Metabolism
brnQ	Metabolism
btuE	Metabolism
cadA	Metabolism
cadB	Metabolism
can	Metabolism
carA	Metabolism
carB	Metabolism
cdd	Metabolism
cdh	Metabolism
cdsA	Metabolism
chaA	Metabolism
cheR	Metabolism
citC	Metabolism
clcA	Metabolism
clsA	Metabolism
clsB	Metabolism
clsC	Metabolism
cmk	Metabolism

Table S6 continued.

Gene	Function
coaA	Metabolism
coaD	Metabolism
coaE	Metabolism
codA	Metabolism
cof	Metabolism
copA	Metabolism
corA	Metabolism
cpdA	Metabolism
cpsB	Metabolism
cpsG	Metabolism
crr	Metabolism
csdA	Metabolism
cueO	Metabolism
cyaA	Metabolism
cycA	Metabolism
cydA	Metabolism
cydB	Metabolism
cydC	Metabolism
cydD	Metabolism
cynT	Metabolism
cyoA	Metabolism
cyoB	Metabolism
cyoC	Metabolism
cyoD	Metabolism
cyoE	Metabolism
cysA	Metabolism
cysC	Metabolism
cysD	Metabolism
cysE	Metabolism
cysG	Metabolism
cysH	Metabolism
cysI	Metabolism
cysJ	Metabolism
cysK	Metabolism
cysM	Metabolism
cysN	Metabolism
cysP	Metabolism
cysQ	Metabolism
cysS	Metabolism
cysU	Metabolism
cysW	Metabolism
cytR	Transcription Regulation
dadA	Metabolism
dadX	Metabolism
dapA	Metabolism
dapB	Metabolism
dapD	Metabolism
dapE	Metabolism
dapF	Metabolism
dauA	Metabolism
dcd	Metabolism
dctA	Metabolism
dcuA	Metabolism
dcuB	Metabolism
dcuR	Transcription Regulation
dcyD	Metabolism
ddlA	Metabolism
ddlB	Metabolism
ddpX	Metabolism
def	Metabolism
deoA	Metabolism
deoB	Metabolism
deoC	Metabolism
deoD	Metabolism
dfp	Metabolism
dgkA	Metabolism
dgoA	Metabolism
dgt	Metabolism
dhaK	Metabolism
dhaL	Metabolism
dhaM	Metabolism
dkgA	Metabolism
dkgB	Metabolism
dld	Metabolism
dmlA	Metabolism
dmsA	Metabolism
dmsB	Metabolism
dmsC	Metabolism
dnaA	Transcription Regulation
dppA	Metabolism
dppB	Metabolism
dppC	Metabolism
dppD	Metabolism
dppF	Metabolism
dsdA	Metabolism
dtpB	Metabolism
dtpD	Metabolism
dut	Metabolism
dxr	Metabolism
dxs	Metabolism
eamA	Metabolism
eamB	Metabolism
eda	Metabolism
edd	Metabolism
efeB	Metabolism
emrE	Metabolism
emtA	Metabolism
eno	Metabolism
entA	Metabolism
entC	Metabolism
entD	Metabolism
entE	Metabolism
epd	Metabolism
epmB	Metabolism
eptB	Metabolism
etk	Metabolism
etp	Metabolism
eutB	Metabolism
eutC	Metabolism
eutD	Metabolism
exuT	Metabolism
fabA	Metabolism
fabB	Metabolism
fabD	Metabolism
fabF	Metabolism
fabG	Metabolism
fabH	Metabolism
fabI	Metabolism
fabZ	Metabolism
fadA	Metabolism
fadB	Metabolism
fadD	Metabolism
fadE	Metabolism
fadI	Metabolism
fadJ	Metabolism
fadK	Metabolism
fadL	Metabolism
fadM	Metabolism
fau	Metabolism
fbaA	Metabolism
fbaB	Metabolism
fbp	Metabolism
fdhF	Metabolism
fdnG	Metabolism
fdnH	Metabolism
fdnI	Metabolism
fdoG	Metabolism
fdoH	Metabolism
fdoI	Metabolism
feaB	Metabolism
feoB	Metabolism
fes	Metabolism
fieF	Metabolism
fis	Transcription Regulation
fklB	Metabolism
fkpA	Metabolism
fkpB	Metabolism
fliY	Metabolism
fmt	Metabolism
fnr	Transcription Regulation
focA	Metabolism
folA	Metabolism
folB	Metabolism
folC	Metabolism
folD	Metabolism
folE	Metabolism
folK	Metabolism
folM	Metabolism
folP	Metabolism
folX	Metabolism
fpr	Metabolism
frc	Metabolism
frdA	Metabolism
frdB	Metabolism
frdC	Metabolism

Table S6 continued.

Gene	Function
frdD	Metabolism
fre	Metabolism
frlB	Metabolism
frlC	Metabolism
frlD	Metabolism
frmA	Metabolism
frmB	Metabolism
ftsW	Metabolism
fucA	Metabolism
fucI	Metabolism
fucK	Metabolism
fucO	Metabolism
fucP	Metabolism
fucU	Metabolism
fumA	Metabolism
fumB	Metabolism
fumC	Metabolism
gabD	Metabolism
gabP	Metabolism
gabT	Metabolism
gadA	Metabolism
gadB	Metabolism
gadC	Metabolism
galE	Metabolism
galK	Metabolism
galM	Metabolism
galP	Metabolism
galT	Metabolism
gapA	Metabolism
garK	Metabolism
garL	Metabolism
garR	Metabolism
gatY	Metabolism
gatZ	Metabolism
gcl	Metabolism
gcvP	Metabolism
gcvT	Metabolism
gdhA	Metabolism
ghrA	Metabolism
ghrB	Metabolism
glcA	Metabolism
glcD	Metabolism
glcE	Metabolism
glcF	Metabolism
gldA	Metabolism
glf	Metabolism
glgC	Metabolism
glk	Metabolism
glmM	Metabolism
glmS	Metabolism
glmU	Metabolism
glnA	Metabolism
glnH	Metabolism
glnP	Metabolism
glnQ	Metabolism
glnS	Metabolism
gloA	Metabolism
gloB	Metabolism
glpA	Metabolism
glpB	Metabolism
glpC	Metabolism
glpD	Metabolism
glpE	Metabolism
glpF	Metabolism
glpK	Metabolism
glpQ	Metabolism
glpX	Metabolism
gltA	Metabolism
gltB	Metabolism
gltD	Metabolism
gltI	Metabolism
gltJ	Metabolism
gltK	Metabolism
gltL	Metabolism
gltP	Metabolism
gltX	Metabolism
glxK	Metabolism
glxR	Metabolism
glyA	Metabolism
glyQ	Metabolism
glyS	Metabolism
gmhB	Metabolism
gmk	Metabolism
gnd	Metabolism
gntK	Metabolism
gntP	Metabolism
gor	Metabolism
gpmA	Metabolism
gpmM	Metabolism
gpp	Metabolism
gpsA	Metabolism
gpt	Metabolism
gshA	Metabolism
gshB	Metabolism
gsiA	Metabolism
gsiB	Metabolism
gsiC	Metabolism
gsiD	Metabolism
gsk	Metabolism
gss	Metabolism
guaA	Metabolism
guaB	Metabolism
guaC	Metabolism
guaD	Metabolism
gutQ	Metabolism
hchA	Metabolism
hdhA	Metabolism
hemA	Metabolism
hemB	Metabolism
hemC	Metabolism
hemD	Metabolism
hemE	Metabolism
hemF	Metabolism
hemG	Metabolism
hemL	Metabolism
hemN	Metabolism
hisA	Metabolism
hisB	Metabolism
hisC	Metabolism
hisD	Metabolism
hisF	Metabolism
hisG	Metabolism
hisH	Metabolism
hisI	Metabolism
hisJ	Metabolism
hisM	Metabolism
hisP	Metabolism
hisQ	Metabolism
hisS	Metabolism
hns	Transcription Regulation
hpt	Metabolism
hycB	Metabolism
hycC	Metabolism
hycD	Metabolism
hycE	Metabolism
hycF	Metabolism
hycG	Metabolism
hyi	Metabolism
hypF	Metabolism
icd	Metabolism
idi	Metabolism
idnD	Metabolism
idnK	Metabolism
idnO	Metabolism
ihfA	Transcription Regulation
ihfB	Transcription Regulation
ileS	Metabolism
ilvA	Metabolism
ilvB	Metabolism
ilvC	Metabolism
ilvD	Metabolism
ilvE	Metabolism
ilvG_1	Metabolism
ilvG_2	Metabolism
ilvH	Metabolism
ilvI	Metabolism
ilvM	Metabolism
ilvN	Metabolism
iscS	Metabolism
ispA	Metabolism
ispB	Metabolism
ispD	Metabolism
ispE	Metabolism
ispF	Metabolism
ispG	Metabolism

Table S6 continued.

Gene	Function
ispH	Metabolism
ispU	Metabolism
katE	Metabolism
katG	Metabolism
kbaY	Metabolism
kbaZ	Metabolism
kbl	Metabolism
kch	Metabolism
kdgK	Metabolism
kdpA	Metabolism
kdpB	Metabolism
kdpC	Metabolism
kdpF	Metabolism
kdsA	Metabolism
kdsB	Metabolism
kdsC	Metabolism
kdsD	Metabolism
kduD	Metabolism
kefB	Metabolism
kefC	Metabolism
kefF	Metabolism
kup	Metabolism
lacA	Metabolism
lacY	Metabolism
lacZ	Metabolism
ldcC	Metabolism
ldhA	Metabolism
leuA	Metabolism
leuB	Metabolism
leuC	Metabolism
leuD	Metabolism
leuE	Metabolism
leuO	Transcription Regulation
leuS	Metabolism
lexA	Transcription Regulation
lhgO	Metabolism
lipA	Metabolism
lipB	Metabolism
livF	Metabolism
livG	Metabolism
livH	Metabolism
livJ	Metabolism
livK	Metabolism
livM	Metabolism
lldD	Metabolism
lldP	Metabolism
lpcA	Metabolism
lpd	Metabolism
lplA	Metabolism
lptA	Metabolism
lptB	Metabolism
lptC	Metabolism
lptD	Metabolism
lptE	Metabolism
lptF	Metabolism

Gene	Function
lptG	Metabolism
lpxA	Metabolism
lpxB	Metabolism
lpxC	Metabolism
lpxD	Metabolism
lpxH	Metabolism
lpxK	Metabolism
lpxL	Metabolism
lpxM	Metabolism
lpxP	Metabolism
lpxT	Metabolism
lrp	Transcription Regulation
lsrA	Metabolism
lsrB	Metabolism
lsrC	Metabolism
lsrD	Metabolism
lsrF	Metabolism
lsrG	Metabolism
lsrK	Metabolism
ltaE	Metabolism
luxS	Metabolism
lysA	Metabolism
lysC	Metabolism
lysP	Metabolism
lysS	Metabolism
lysU	Metabolism
lyxK	Metabolism
maa	Metabolism
maeA	Metabolism
maeB	Metabolism
mak	Metabolism
malP	Metabolism
malQ	Metabolism
malY	Metabolism
malZ	Metabolism
manA	Metabolism
manX	Metabolism
manY	Metabolism
manZ	Metabolism
mazG	Metabolism
mdaB	Metabolism
mdfA	Metabolism
mdh	Metabolism
mdoB	Metabolism
mdtI	Metabolism
mdtJ	Metabolism
mdtM	Metabolism
menA	Metabolism
menB	Metabolism
menC	Metabolism
menD	Metabolism
menE	Metabolism
menF	Metabolism
menH	Metabolism
menI	Metabolism

Gene	Function
metA	Metabolism
metB	Metabolism
metC	Metabolism
metE	Metabolism
metF	Metabolism
metG	Metabolism
metH	Metabolism
metI	Metabolism
metJ	Transcription Regulation
metK	Metabolism
metL	Metabolism
metN	Metabolism
metQ	Metabolism
mgsA	Metabolism
mgtA	Metabolism
mhpD	Metabolism
mhpE	Metabolism
mhpF	Metabolism
miaA	Metabolism
miaB	Metabolism
mltA	Metabolism
mltB	Metabolism
mltC	Metabolism
mltD	Metabolism
mltF	Metabolism
mmuM	Metabolism
mnaT	Metabolism
mntH	Metabolism
mntP	Metabolism
mpaA	Metabolism
mpl	Metabolism
mppA	Metabolism
mqo	Metabolism
mraY	Metabolism
msbA	Metabolism
msrA	Metabolism
msrB	Metabolism
msrC	Metabolism
mtlD	Metabolism
mtn	Metabolism
mtr	Metabolism
murA	Metabolism
murB	Metabolism
murC	Metabolism
murD	Metabolism
murE	Metabolism
murF	Metabolism
murG	Metabolism
murI	Metabolism
murJ	Metabolism
murQ	Metabolism
nadA	Metabolism
nadB	Metabolism
nadC	Metabolism
nadD	Metabolism

Table S6 continued.

Gene	Function
nadE	Metabolism
nadK	Metabolism
nadR	Metabolism
nagA	Metabolism
nagD	Metabolism
nagK	Metabolism
nanA	Metabolism
nanC	Metabolism
nanE	Metabolism
nanK	Metabolism
nanM	Metabolism
nanT	Metabolism
narK	Metabolism
narL	Transcription Regulation
ndh	Metabolism
ndk	Metabolism
nepI	Metabolism
nhaA	Metabolism
nhaB	Metabolism
nhoA	Metabolism
nikA	Metabolism
nikB	Metabolism
nikC	Metabolism
nikD	Metabolism
nikE	Metabolism
nirB	Metabolism
nirC	Metabolism
nirD	Metabolism
nrdA	Metabolism
nrdB	Metabolism
nrdD	Metabolism
nrdE	Metabolism
nrdF	Metabolism
nrdG	Metabolism
nudB	Metabolism
nudF	Metabolism
nudI	Metabolism
nudJ	Metabolism
nudK	Metabolism
nudL	Metabolism
nuoA	Metabolism
nuoB	Metabolism
nuoC	Metabolism
nuoE	Metabolism
nuoF	Metabolism
nuoG	Metabolism
nuoH	Metabolism
nuoI	Metabolism
nuoJ	Metabolism
nuoK	Metabolism
nuoL	Metabolism
nuoM	Metabolism
nuoN	Metabolism
nupC	Metabolism
nupG	Metabolism

Gene	Function
ompC	Metabolism
ompF	Metabolism
oppA	Metabolism
oppB	Metabolism
oppC	Metabolism
oppD	Metabolism
oppF	Metabolism
orn	RNA Decay
otsA	Metabolism
otsB	Metabolism
oxc	Metabolism
paaA	Metabolism
paaB	Metabolism
paaC	Metabolism
paaE	Metabolism
paaF	Metabolism
paaG	Metabolism
paaH	Metabolism
paaI	Metabolism
paaJ	Metabolism
paaK	Metabolism
paaZ	Metabolism
pabA	Metabolism
pabB	Metabolism
pabC	Metabolism
panB	Metabolism
panC	Metabolism
panE	Metabolism
patA	Metabolism
patD	Metabolism
pck	Metabolism
pdxA	Metabolism
pdxH	Metabolism
pdxJ	Metabolism
pdxK	Metabolism
pdxY	Metabolism
pepA	Metabolism
pepB	Metabolism
pepD	Metabolism
pepN	Metabolism
pfkA	Metabolism
pfkB	Metabolism
pgaB	Metabolism
pgi	Metabolism
pgk	Metabolism
pgl	Metabolism
pgm	Metabolism
pgpA	Metabolism
pgpB	Metabolism
pgpC	Metabolism
pgsA	Metabolism
pheA	Metabolism
pheP	Metabolism
pheS	Metabolism
pheT	Metabolism

Gene	Function
phnC	Metabolism
phnD	Metabolism
phnE_1	Metabolism
phnN	Metabolism
phnO	Metabolism
phoA	Metabolism
phoE	Metabolism
pitA	Metabolism
pitB	Metabolism
plaP	Metabolism
pldB	Metabolism
plsB	Metabolism
plsC	Metabolism
pncA	Metabolism
pncB	Metabolism
pncC	Metabolism
pnp	Metabolism, RNA Decay
pntA	Metabolism
pntB	Metabolism
polA	RNA Decay
potA	Metabolism
potB	Metabolism
potC	Metabolism
potD	Metabolism
potE	Metabolism
potF	Metabolism
potG	Metabolism
potH	Metabolism
potI	Metabolism
poxB	Metabolism
ppa	Metabolism
ppc	Metabolism
pphA	Metabolism
pphB	Metabolism
ppiA	Metabolism
ppiB	Metabolism
ppiC	Metabolism
ppk	Metabolism
ppsA	Metabolism
preA	Metabolism
preT	Metabolism
prmC	Metabolism
proA	Metabolism
proB	Metabolism
proC	Metabolism
proP	Metabolism
proS	Metabolism
proV	Metabolism
proW	Metabolism
proX	Metabolism
prpB	Metabolism
prpC	Metabolism
prpD	Metabolism
prpE	Metabolism
prs	Metabolism

Table S6 continued.

Gene	Function
pssA	Metabolism
pstA	Metabolism
pstB	Metabolism
pstC	Metabolism
pstS	Metabolism
psuG	Metabolism
pta	Metabolism
ptsG	Metabolism
purA	Metabolism
purB	Metabolism
purC	Metabolism
purD	Metabolism
purE	Metabolism
purF	Metabolism
purH	Metabolism
purK	Metabolism
purL	Metabolism
purM	Metabolism
purN	Metabolism
purT	Metabolism
purU	Metabolism
putA	Transcription Regulation, Metabolism
putP	Metabolism
puuA	Metabolism
puuB	Metabolism
puuC	Metabolism
puuD	Metabolism
puuE	Metabolism
puuP	Metabolism
pykA	Metabolism
pykF	Metabolism
pyrB	Metabolism
pyrC	Metabolism
pyrD	Metabolism
pyrE	Metabolism
pyrF	Metabolism
pyrG	Metabolism
pyrH	Metabolism
pyrI	Metabolism
qor	Metabolism
rbn	RNA Decay
rbsD	Metabolism
rcnA	Metabolism
rcsB	Transcription Regulation
rdgB	Metabolism
rdoA	Metabolism
relA	Metabolism
rfaD	Metabolism
rfaE	Metabolism
rfbA	Metabolism
rfbB	Metabolism
rfbC	Metabolism
rfbD	Metabolism
rffA	Metabolism
rffE	Metabolism
rffG	Metabolism
rffH	Metabolism
rhaA	Metabolism
rhaB	Metabolism
rhaD	Metabolism
rhaM	Metabolism
rhaT	Metabolism
rhlB	Metabolism
rhtA	Metabolism
rhtC	Metabolism
ribA	Metabolism
ribB	Metabolism
ribC	Metabolism
ribD	Metabolism
ribE	Metabolism
ribF	Metabolism
rihA	Metabolism
rihB	Metabolism
rihC	Metabolism
rna	RNA Decay
rnb	RNA Decay
rnc	RNA Decay
rnd	RNA Decay
rne	Metabolism, RNA Decay
rng	RNA Decay
rnhA	RNA Decay
rnhB	RNA Decay
rnlA	RNA Decay
rnpA	RNA Decay
rnr	RNA Decay
rnt	RNA Decay
rpe	Metabolism
rph	RNA Decay
rpiA	Metabolism
rpiB	Metabolism
rplA	Translation
rplB	Translation
rplC	Translation
rplD	Translation
rplE	Translation
rplF	Translation
rplI	Translation
rplJ	Translation
rplK	Translation
rplL	Translation
rplM	Translation
rplN	Translation
rplO	Translation
rplP	Translation
rplQ	Translation
rplR	Translation
rplS	Translation
rplT	Translation
rplU	Translation
rplV	Translation
rplW	Translation
rplX	Translation
rplY	Translation
rpmA	Translation
rpmB	Translation
rpmC	Translation
rpmD	Translation
rpmE	Translation
rpmF	Translation
rpmG	Translation
rpmH	Translation
rpmI	Translation
rpmJ	Translation
rpoA	Transcription
rpoB	Transcription
rpoC	Transcription
rpsA	Translation
rpsB	Translation
rpsC	Translation
rpsD	Translation
rpsE	Translation
rpsF	Translation
rpsG	Translation
rpsH	Translation
rpsI	Translation
rpsJ	Translation
rpsK	Translation
rpsL	Translation
rpsM	Translation
rpsN	Translation
rpsO	Translation
rpsP	Translation
rpsQ	Translation
rpsR	Translation
rpsS	Translation
rpsT	Translation
rpsU	Translation
rrrD	Metabolism
rsmI	Metabolism
rutG	Metabolism
sad	Metabolism
sbmA	Metabolism
sbp	Metabolism
scpA	Metabolism
scpB	Metabolism
scpC	Metabolism
sdaA	Metabolism
sdaB	Metabolism
sdaC	Metabolism
sdhA	Metabolism
sdhB	Metabolism
sdhC	Metabolism
sdhD	Metabolism
serA	Metabolism
serB	Metabolism

Table S6 continued.

Gene	Function
serC	Metabolism
serS	Metabolism
setA	Metabolism
setB	Metabolism
sgbE	Metabolism
sgbH	Metabolism
sgbU	Metabolism
slt	Metabolism
slyD	Metabolism
sodC	Metabolism
speB	Metabolism
speC	Metabolism
speE	Metabolism
speF	Metabolism
speG	Metabolism
spoT	Metabolism
spr	Metabolism
sra	Translation
srlD	Metabolism
sstT	Metabolism
ssuA	Metabolism
ssuB	Metabolism
ssuC	Metabolism
ssuD	Metabolism
ssuE	Metabolism
sthA	Metabolism
sucA	Metabolism
sucB	Metabolism
sucC	Metabolism
sucD	Metabolism
sufS	Metabolism
surA	Metabolism
surE	Metabolism
tadA	Metabolism
talA	Metabolism
talB	Metabolism
tcdA	Metabolism
tdcB	Metabolism
tdcC	Metabolism
tdcD	Metabolism
tdcG	Metabolism
tdh	Metabolism
tdk	Metabolism
tesA	Metabolism
thiC	Metabolism
thiD	Metabolism
thiE	Metabolism
thiF	Metabolism
thiG	Metabolism
thiH	Metabolism
thiI	Metabolism
thiL	Metabolism
thrA	Metabolism
thrB	Metabolism
thrC	Metabolism

Gene	Function
thrS	Metabolism
thyA	Metabolism
tig	Metabolism
tilS	Metabolism
tktA	Metabolism
tktB	Metabolism
tmk	Metabolism
tnaA	Metabolism
tnaB	Metabolism
tolB	Metabolism
tolC	Metabolism
tpiA	Metabolism
tppB	Metabolism
tqsA	Metabolism
treC	Metabolism
treF	Metabolism
trkG	Metabolism
trkH	Metabolism
trmD	Metabolism
trpA	Metabolism
trpB	Metabolism
trpC	Metabolism
trpD	Metabolism
trpE	Metabolism
trpR	Transcription Regulation
trpS	Metabolism
truB	Metabolism
truC	Metabolism
truD	Metabolism
trxB	Metabolism
tsx	Metabolism
ttdA	Metabolism
ttdB	Metabolism
tyrA	Metabolism
tyrB	Metabolism
tyrP	Metabolism
tyrR	Transcription Regulation
tyrS	Metabolism
uacT	Metabolism
ubiA	Metabolism
ubiC	Metabolism
ubiD	Metabolism
ubiE	Metabolism
ubiF	Metabolism
ubiG	Metabolism
ubiH	Metabolism
udk	Metabolism
udp	Metabolism
ugd	Metabolism
ugpA	Metabolism
ugpB	Metabolism
ugpC	Metabolism
ugpE	Metabolism
ugpQ	Metabolism
ulaD	Metabolism

Gene	Function
ulaE	Metabolism
ulaF	Metabolism
upp	Metabolism
uraA	Metabolism
ushA	Metabolism
uxaA	Metabolism
uxaB	Metabolism
uxaC	Metabolism
uxuA	Metabolism
uxuB	Metabolism
valS	Metabolism
visC	Metabolism
waaA	Metabolism
waaB	Metabolism
waaC	Metabolism
waaF	Metabolism
waaG	Metabolism
waaI	Metabolism
waaJ	Metabolism
waaP	Metabolism
waaQ	Metabolism
waaU	Metabolism
waaY	Metabolism
waaZ	Metabolism
wbbI	Metabolism
wcaJ	Metabolism
wrbA	Metabolism
wza	Metabolism
wzb	Metabolism
wzc	Metabolism
xanP	Metabolism
xanQ	Metabolism
xapA	Metabolism
xapB	Metabolism
xdhA	Metabolism
xdhB	Metabolism
xdhC	Metabolism
xylA	Metabolism
xylB	Metabolism
xylE	Metabolism
xylF	Metabolism
xylG	Metabolism
xylH	Metabolism
yaaH	Metabolism
yagE	Metabolism
yagF	Metabolism
yahK	Metabolism
ybaS	Metabolism
ybbO	Metabolism
ybcF	Metabolism
ybdL	Metabolism
ybgT	Metabolism
ybhA	Metabolism
ybiV	Metabolism
ybjE	Metabolism

Table S6 continued.

Gene	Function
ybjG	Metabolism
ybjI	Metabolism
ycaO	Metabolism
ycbL	Metabolism
ycjG	Metabolism
ycjU	Metabolism
ydbK	Metabolism
yddG	Metabolism
ydeA	Metabolism
ydfG	Metabolism
ydiB	Metabolism
ydiF	Metabolism
ydiO	Metabolism
ydjG	Metabolism
yeaE	Metabolism
yeaG	Metabolism
yecC	Metabolism
yecS	Metabolism
yegS	Metabolism
yeiG	Metabolism
yejA	Metabolism
yejB	Metabolism
yejE	Metabolism
yejF	Metabolism
yfaU	Metabolism

Gene	Function
yfaW	Metabolism
yfbR	Metabolism
yfbT	Metabolism
yfdE	Metabolism
yfdR	Metabolism
yfeX	Metabolism
ygaH	Metabolism
ygaZ	Metabolism
ygfQ	Metabolism
yggF	Metabolism
yghA	Metabolism
yghZ	Metabolism
ygiF	Metabolism
yhbO	Metabolism
yhdE	Metabolism
yhdH	Metabolism
yiaK	Metabolism
yiaY	Metabolism
yibQ	Metabolism
yidA	Metabolism
yieF	Metabolism
yigB	Metabolism
yigL	Metabolism
yihS	Metabolism
yihU	Metabolism

Gene	Function
yihV	Metabolism
yihX	Metabolism
yijE	Metabolism
yjbB	Metabolism
yjcD	Metabolism
yjdL	Metabolism
yjeF	Metabolism
yjeH	Metabolism
yjhG	Metabolism
yjhH	Metabolism
yjjG	Metabolism
yjjX	Metabolism
yliI	Metabolism
yneH	Metabolism
yqhD	Metabolism
yrfG	Metabolism
ytfG	Metabolism
zitB	Metabolism
zntA	Metabolism
znuA	Metabolism
znuB	Metabolism
znuC	Metabolism
zupT	Metabolism
zwf	Metabolism

Table S7. Metabolites that are included in the homeostatic objective in the model. Initial concentration targets come from two sources: source 1 (mass spec data (*17*)) or source 2 (cellular biomass from the FBA model (*15*) combined with growth dependent mass fractions (*31*) and average mass fractions (*52*)). The table is also included in the GitHub repository: wcEcoli/paper/metabolite_pools.tsv.

Metabolite	Source
2-3-DIHYDROXYBENZOATE[c]	1
2-KETOGLUTARATE[c]	1
4-hydroxybenzoate[c]	1
ACETOACETYL-COA[c]	1
ACETYL-COA[c]	1
ACETYL-P[c]	1
ADENINE[c]	1
ADENOSINE[c]	1
ADP-D-GLUCOSE[c]	1
ADP[c]	1
AMP[c]	1
ANTHRANILATE[c]	1
APS[c]	1
ARG[c]	1
ASN[c]	1
ATP[c]	1
BIOTIN[c]	2
CA+2[c]	2
CA+2[p]	2
CARBAMYUL-L-ASPARTATE[c]	1
CHORISMATE[c]	2
CIS-ACONITATE[c]	1
CIT[c]	1
CL-[c]	2
CMP[c]	1
CO+2[c]	2
CO+2[p]	2
CO-A[c]	1
CPD-12115[c]	2
CPD-12261[p]	2
CPD-12575[c]	1
CPD-12819[c]	2
CPD-12824[c]	2
CPD-13469[c]	1
CPD-2961[c]	1
CPD-8260[c]	2
CPD-9956[c]	2
CPD0-939[c]	2
CTP[c]	1
CYS[c]	2
CYTIDINE[c]	1
CYTOSINE[c]	1
DAMP[c]	1
DATP[c]	1
DCTP[c]	1
DEOXYADENOSINE[c]	1
DEOXYGUANOSINE[c]	1
DGMP[c]	1
DGTP[c]	2
DI-H-OROTATE[c]	1
DIHYDROXY-ACETONE-PHOSPHATE[c]	1
ENTEROBACTIN[c]	2
FAD[c]	1
FE+2[c]	2
FE+2[p]	2
FRUCTOSE-16-DIPHOSPHATE[c]	1
FUM[c]	1
G3P[c]	1
GDP[c]	1
GLC-D-LACTONE[c]	1
GLN[c]	1
GLT[c]	1
GLUCONATE[c]	1
GLUTATHIONE[c]	1
GLYCERATE[c]	1
GLYCEROL-3P[c]	1
GLY[c]	2
GMP[c]	1
GTP[c]	1
GUANINE[c]	1
GUANOSINE[c]	1
HISTIDINOL[c]	1
HIS[c]	1
HOMO-CYS[c]	1
ILE[c]	2
K+[c]	2
L-ALPHA-ALANINE[c]	1
L-ASPARTATE[c]	1
L-CITRULLINE[c]	1
L-ORNITHINE[c]	1
L-SELENOCYSTEINE[c]	2
LEU[c]	2
LYS[c]	1
MALONYL-COA[c]	1
MAL[c]	1
METHYLENE-THF[c]	2
MET[c]	1
MG+2[c]	2
MN+2[c]	2
MN+2[p]	2

Table S7 continued.

Metabolite	Source
N-ACETYL-D-GLUCOSAMINE-1-P[c]	1
N-ALPHA-ACETYLORNITHINE[c]	1
NADH[c]	1
NADPH[c]	1
NADP[c]	1
NAD[c]	1
NI+2[c]	2
NI+2[p]	2
OXYGEN-MOLECULE[p]	2
PHENYL-PYRUVATE[c]	1
PHE[c]	1
PHOSPHO-ENOL-PYRUVATE[c]	1
PI[c]	2
PI[p]	2
PPI[c]	2
PROPIONYL-COA[c]	1
PROTOHEME[c]	2
PROTON[c]	2
PRO[c]	1
PUTRESCINE[c]	2
PYRIDOXAL_PHOSPHATE[c]	2
QUINOLINATE[c]	1
REDUCED-MENAQUINONE[c]	2
RIBOFLAVIN[c]	1
S-ADENOSYLMETHIONINE[c]	1

Metabolite	Source
SER[c]	1
SHIKIMATE[c]	1
SIROHEME[c]	2
SPERMIDINE[c]	2
SUC-COA[c]	1
SUC[c]	1
TDP[c]	1
THF[c]	2
THIAMINE-PYROPHOSPHATE[c]	2
THR[c]	1
TRP[c]	1
TTP[c]	1
TYR[c]	1
UDP-GLUCURONATE[c]	1
UDP-N-ACETYL-D-GLUCOSAMINE[c]	1
UDP[c]	1
UMP[c]	2
UNDECAPRENYL-DIPHOSPHATE[c]	2
URIDINE[c]	1
UTP[c]	1
VAL[c]	1
WATER[c]	2
ZN+2[c]	2
ZN+2[p]	2
glycogen-monomer[c]	2

Table S8. Kinetic constraints curated for the model Pubmed ID shows original data reference. Reaction ID is the model ID for the reaction. k_{cat} and K_M show the curated values with Adjusted k_{cat} showing the temperature adjusted k_{cat} (at 37 ° C). Enzyme and Substrates are the model IDs. Substrates for K_M correspond to each listed K_M value. Temp is the temperature that measurements were made at. Excluded indicates if the constraint is excluded from the final model (”True” means that the constraint was excluded). The table is also included in the GitHub repository: wcEcoli/paper/kinetic_constraints.tsv.

Pubmed ID	Reaction ID	k_{cat} (1/s)	K_M (uM)	Adjusted k_{cat} (1/s)	Enzyme	Substrates	Substrates for K_M	Temp (C)	Excluded
11237876	1.1.1.215-RXN	40.7	[]	93.5	CPLX0-235[i]	[’CPD-377[c]’]	[]	25	False
10225425	1.1.1.262-RXN	1.4	[]	1.4	CPLX0-7847[c]	[’4-PHOSPHONOOXY-THREONINE[c]’]	[]	37	False
17557829	1.1.1.39-RXN	82.7	[660, 68.8]	134.3	MALIC-NAD-CPLX[c]	[’MAL[c]’, ’NAD[c]’]	[’MAL[c]’, ’NAD[c]’]	30	False
25160617	1.1.1.83-RXN	11.7	[94]	26.88	G6986-MONOMER[c]	[’CPD-660[c]’, ’NAD[c]’]	[’NAD[c]’]	25	False
11604533	1.8.4.13-RXN-MET/Ox-Thioredoxin/WATER//CPD-8989/Red-Thioredoxin.51.__EG11433-MONOMER	3.7	[]	8.5	EG11433-MONOMER[c]	[’CPD-8989[c]’, ’Red-Thioredoxin[c]’]	[]	25	False
15680231	1.8.4.14-RXN-MET/Ox-Thioredoxin/WATER//CPD-8990/Red-Thioredoxin.51.__EG12394-MONOMER	1.1	[]	2.527	EG12394-MONOMER[c]	[’Red-Thioredoxin[c]’]	[]	25	False
17535911	1.8.4.14-RXN-MET/Ox-Thioredoxin/WATER//CPD-8990/Red-Thioredoxin.51.__G7005-MONOMER	6.9	[]	15.85	G7005-MONOMER[c]	[’CPD-8990[c]’, ’Red-Thioredoxin[c]’]	[]		False
11724562	2-DEHYDROPANTOATE-REDUCT-RXN__2-DEHYDROPANTOATE-REDUCT-MONOMER	40	[20]	40	2-DEHYDROPANTOATE-REDUCT-MONOMER[c]	[’2-DEHYDROPANTOATE[c]’, ’NADPH[c]’]	[’NADPH[c]’]	37	False
15654896	2-DEHYDROPANTOATE-REDUCT-RXN__CPLX0-7643	0.194	[]	0.194	CPLX0-7643[c]	[’2-DEHYDROPANTOATE[c]’]	[]	37	False
9492273	2-OXOPENT-4-ENOATE-HYDRATASE-RXN	450	[]	1034	CPLX0-7951[c]	[’OXOPENTENOATE[c]’]	[]	25	False
11084021	2.3.1.157-RXN	1350	[]	1350	NAG1P-URIDYLTRANS-CPLX[c]	[’ACETYL-COA[c]’]	[]	37	False
18567546	2.5.1.19-RXN	26.4	[100]	60.65	AROA-MONOMER[c]	[’PHOSPHO-ENOL-PYRUVATE[c]’]	[’PHOSPHO-ENOL-PYRUVATE[c]’]	25	False
21928762	2.5.1.64-RXN	0.28	[9.9]	0.6433	CPLX0-7525[c]	[’2-KETOGLUTARATE[c]’]	[’2-KETOGLUTARATE[c]’]	25	False
15379557	2.7.7.60-RXN	48.4	[]	127.7	CPLX0-234[c]	[’2-C-METHYL-D-ERYTHRITOL-4-PHOSPHATE[c]’]	[]	23	False

11686925	2.8.1.6-RXN	13.89	[]	13.89	BIOTIN-SYN-CPLX[c]	['DETHIOBIOTIN[c]']	[]	37	False
2514789	3-DEHYDROQUINATE-SYNTHASE-RXN	150	[]	487.4	AROB-MONOMER[c]	['3-DEOXY-D-ARABINO-HEPTULOSONATE-7-P[c]']	[]	20	False
9003442	3-ISOPROPYLMALDEHYDROG-RXN__3-ISOPROPYLMALDEHYDROG-CPLX	69	[321]	56.05	3-ISOPROPYLMALDEHYDROG-CPLX[c]	['2-D-THREO-HYDROXY-3-CARBOXY-ISOCAPROATE[c]', 'NAD[c]']	['NAD[c]']	40	False
25160617	3-ISOPROPYLMALDEHYDROG-RXN__G6986-MONOMER	0.102	[70]	0.2343	G6986-MONOMER[c]	['2-D-THREO-HYDROXY-3-CARBOXY-ISOCAPROATE[c]', 'NAD[c]']	['NAD[c]']	25	False
16990279	3.1.3.74-RXN[CCO-CYTOSOL]-PYRIDOXAL_PHOSPHATE/WATER//PYRIDOXAL/Pi.53.__EG11239-MONOMER	1	[370]	2.297	EG11239-MONOMER[c]	['PYRIDOXAL_PHOSPHATE[c]']	['PYRIDOXAL_PHOSPHATE[c]']		False
16990279	3.1.3.74-RXN[CCO-CYTOSOL]-PYRIDOXAL_PHOSPHATE/WATER//PYRIDOXAL/Pi.53.__EG11470-MONOMER	12	[1500]	27.57	EG11470-MONOMER[c]	['PYRIDOXAL_PHOSPHATE[c]']	['PYRIDOXAL_PHOSPHATE[c]']		False
16990279	3.1.3.74-RXN[CCO-CYTOSOL]-PYRIDOXAL_PHOSPHATE/WATER//PYRIDOXAL/Pi.53.__G6246-MONOMER	0.58	[680]	1.332	G6246-MONOMER[c]	['PYRIDOXAL_PHOSPHATE[c]']	['PYRIDOXAL_PHOSPHATE[c]']		False
9610360	3.5.1.88-RXN	2100	[]	3411	EG11440-MONOMER[c]	['FORMYL-L-METHIONYL-PEPTIDE[c]']	[]	30	False
2986688	3.6.1.41-RXN	250	[]	250	EG10048-MONOMER[c]	['ADENOSYL-P4[c]']	[]	37	False
17616624	325-BISPHOSPHATE-NUCLEOTIDASE-RXN	11.4	[]	11.4	EG10043-MONOMER[c]	['3-5-ADP[c]']	[]	37	False
15489502	5-NUCLEOTID-RXN[CCO-CYTOSOL]-AMP/WATER//ADENOSINE/Pi.37.__EG11817-MONOMER	4.9	[320]	4.9	EG11817-MONOMER[c]	['AMP[c]']	['AMP[c]']	37	False
16990279	5-NUCLEOTID-RXN[CCO-CYTOSOL]-AMP/WATER//ADENOSINE/Pi.37.__EG12115-MONOMER	1.3	[1800]	2.987	EG12115-MONOMER[c]	['AMP[c]']	['AMP[c]']		False
15489502	5-NUCLEOTID-RXN[CCO-CYTOSOL]-GMP/WATER//GUANOSINE/Pi.37.__EG11817-MONOMER	11	[260]	11	EG11817-MONOMER[c]	['GMP[c]']	['GMP[c]']	37	False
15489502	5-NUCLEOTID-RXN[CCO-CYTOSOL]-UMP/WATER//URIDINE/Pi.35.__EG12115-MONOMER	34.2	[660]	34.2	EG12115-MONOMER[c]	['UMP[c]']	['UMP[c]']	37	False
10231382	5.4.2.10-RXN-GLUCOSAMINE-1P//CPD-13469.26.	7.9	[]	7.9	PHOSGLUCOSAMINEMUT-MONOMER[c]	['GLUCOSAMINE-1P[c]']	[]	37	False

Kruger, N.J. Effects of temperature on the kinetic properties of phosphofructokinase from Escherichia coli Biochem. Soc. Trans. 17 760-761 1989 Escherichia coli ; 0 0	6PFRUCTPHOS-RXN__6PFK-1-CPX	185	[]	185	6PFK-1-CPX[c]	['FRUCTOSE-6P[c]']	[]	37	False
20887711	6PFRUCTPHOS-RXN__6PFK-2-CPX	62	[12]	142.4	6PFK-2-CPX[c]	['FRUCTOSE-6P[c]', 'ATP[c]']	['ATP[c]']		False
2007545	7-ALPHA-HYDROXYSTEROID-DEH-RXN (reverse)	3798	[]	3798	7-ALPHA-HYDROXYSTEROID-DEH-CPLX[c]	['CHOLATE[c]']	[]	37	False
23237860	ACETALD-DEHYDROG-RXN__MHPF-MONOMER	17.5	[90000, 250]	40.2	MHPF-MONOMER[c]	['ACETALD[c]', 'CO-A[c]', 'NAD[c]']	['CO-A[c]', 'NAD[c]']	25	False
9015391	ACETOLACTREDUCTOISOM-RXN (reverse)	0.052	[]	0.1471	CPLX0-7643[c]	['NADP[c]']	[]	22	False
15654896	ACETOLACTREDUCTOISOM-RXN	2.231	[2.53]	2.231	CPLX0-7643[c]	['2-ACETO-LACTATE[c]', 'NADPH[c]']	['NADPH[c]']	37	False
22443469	ACETOLACTSYN-RXN__ACETOLACTSYNI-CPLX	71.3	[]	71.3	ACETOLACTSYNI-CPLX[c]	['PYRUVATE[c]']	[]	37	False
9581571	ACETOOHBUTSYN-RXN__ACETOLACTSYNII-CPLX	66.7	[]	66.7	ACETOLACTSYNII-CPLX[c]	['PYRUVATE[c]']	[]	37	False
16201833	ACETYLORNDEACET-RXN	3800	[1200]	8730	ACETYLORNDEACET-CPLX[c]	['N-ALPHA-ACETYLORNITHINE[c]']	['N-ALPHA-ACETYLORNITHINE[c]']	25	False
15967977	ACNEULY-RXN (reverse)	10.5	[]	24.12	ACNEULY-CPLX[c]	['N-ACETYLNEURAMINATE[c]']	[]	25	False
16101288	ADENOSYLHOMOCYSTEINE-NUCLEOSIDASE-RXN	2.6	[]	7.354	CPLX0-1541[c]	['ADENOSYL-HOMO-CYS[c]']	[]	22	False
12937174	ADENPHOSPHOR-RXN (reverse)	3.32	[46]	7.627	DEOD-CPLX[c]	['ADENOSINE[c]']	['ADENOSINE[c]']	25	False
357906	ADENPRIBOSYLTRAN-RXN	9.33	[20]	9.33	ADENPRIBOSYLTRAN-CPLX[c]	['ADENINE[c]', 'PRPP[c]']	['ADENINE[c]']	37	False
6300054	ADENYLATECYC-RXN	1.67	[1000]	1.67	ADENYLATECYC-MONOMER[c]	['ATP[c]']	['ATP[c]', 'ATP[c]', 'ADP[c]', 'GTP[c]', 'PPI[c]']	37	False
16981730	ADENYLOSUCCINATE-SYNTHASE-RXN	1	[26, 230]	2.828	ADENYLOSUCCINATE-SYN-DIMER[c]	['IMP[c]', 'GTP[c]', 'L-ASPARTATE[c]']	['GTP[c]', 'L-ASPARTATE[c]']	22	False
2549047	ADENYLYLSULFKIN-RXN	0.206	[]	0.4733	ADENYLYLSULFKIN-CPLX[c]	['ADP[c]']	[]	25	False

2549047	ADENYLYLSULFKIN-RXN (reverse)	50	[]	114.9	ADENYLYLSULFKIN-CPLX[c]	[’APS[c]’]	[]	25	False
7918446	ALANINE–TRNA-LIGASE-RXN-ALA-tRNAs/L-ALPHA-ALANINE/ATP/PROTON//Charged-ALA-tRNAs/AMP/PPI.64.	84	[340, 83]	84	ALAS-CPLX[c]	[’L-ALPHA-ALANINE[c]’, ’ATP[c]’]	[’L-ALPHA-ALANINE[c]’, ’ATP[c]’]	37	True
15796715	ALARACECAT-RXN__CPLX0-8202 (reverse)	27.6	[290]	27.6	CPLX0-8202[c]	[’L-ALPHA-ALANINE[c]’]	[’L-ALPHA-ALANINE[c]’]	37	False
15796715	ALARACECAT-RXN__CPLX0-8202	28.3	[]	28.3	CPLX0-8202[c]	[’D-ALANINE[c]’]	[]	37	False
10406936	ALCOHOL-DEHYDROG-RXN__CPLX0-8015 (reverse)	163	[]	374.5	CPLX0-8015[c]	[’ACETALD[c]’]	[]	25	False
10406936	ALCOHOL-DEHYDROG-RXN__CPLX0-8015	67.5	[]	155.1	CPLX0-8015[c]	[’ETOH[c]’]	[]	25	False
16023116	AMINOBUTDEHYDROG-RXN	5.65	[54]	12.98	CPLX0-3641[c]	[’NAD[c]’]	[’NAD[c]’]	25	False
20170126	ANTHRANSYN-RXN	138	[2500]	317	ANTHRANSYN-CPLX[c]	[’GLN[c]’]	[’GLN[c]’]	25	False
9748544	ARGININE–TRNA-LIGASE-RXN-ARG-tRNAs/ARG/ATP/PROTON//Charged-ARG-tRNAs/AMP/PPI.52.	26	[900, 12]	26	ARGS-MONOMER[c]	[’ATP[c]’, ’ARG-tRNAs[c]’, ’ARG[c]’]	[’ATP[c]’, ’ARG[c]’]	37	True
20853825	ASNSYNA-RXN__ASNSYNB-CPLX	0.75	[1200]	0.75	ASNSYNB-CPLX[c]	[’L-ASPARTATE[c]’]	[’L-ASPARTATE[c]’]	37	False
20853825	ASNSYNA-RXN__ASNSYNB-CPLX	0.96	[110]	0.96	ASNSYNB-CPLX[c]	[’ATP[c]’]	[’ATP[c]’]	37	False
12706338	ASNSYNB-RXN	1.56	[850]	1.56	ASNSYNB-CPLX[c]	[’L-ASPARTATE[c]’]	[’L-ASPARTATE[c]’]	37	False
12706338	ASNSYNB-RXN	2.18	[260]	2.18	ASNSYNB-CPLX[c]	[’ATP[c]’]	[’ATP[c]’]	37	False
12706338	ASNSYNB-RXN	2.73	[690]	2.73	ASNSYNB-CPLX[c]	[’GLN[c]’]	[’GLN[c]’]	37	False
11967363	ASPAMINOTRANS-RXN__ASPAMINOTRANS-DIMER	259	[1900, 590]	595	ASPAMINOTRANS-DIMER[c]	[’L-ASPARTATE[c]’, ’2-KETOGLUTARATE[c]’]	[’L-ASPARTATE[c]’, ’2-KETOGLUTARATE[c]’]	25	False
14767072	ASPAMINOTRANS-RXN__TYRB-DIMER	140	[3800, 1300]	321.6	TYRB-DIMER[c]	[’L-ASPARTATE[c]’, ’2-KETOGLUTARATE[c]’]	[’L-ASPARTATE[c]’, ’2-KETOGLUTARATE[c]’]	25	False
11106175	ASPARAGHYD-RXN[CCO-PERI-BAC]-ASN/WATER//L-ASPARTATE/AMMONIUM.46.	24	[15]	55.14	ANSB-CPLX[c]	[’ASN[p]’]	[’ASN[c]’]	25	False
1544480	ASPARAGINE–TRNA-LIGASE-RXN-ASN-tRNAs/ASN/ATP/PROTON//Charged-ASN-tRNAs/AMP/PPI.52.	1.3	[76]	1.3	ASNS-CPLX[c]	[’ATP[c]’]	[’ATP[c]’]	37	True
1544480	ASPARAGINE–TRNA-LIGASE-RXN-ASN-tRNAs/ASN/ATP/PROTON//Charged-ASN-tRNAs/AMP/PPI.52.	1.6	[32]	1.6	ASNS-CPLX[c]	[’ASN[c]’]	[’ASN[c]’]	37	True

Gou, X.j.; Li, S.; Kong, X.d.; Liu, W.; Sun, Y.h.; Zhang, J. Directed evolution of L-aspartase by mobility of domains Chem. Res. Chin. Univ. 20 50-54 2004	ASPARTASE-RXN	180	[1200]	413.5	ASPARTASE-CPLX[c]	[’L-ASPARTATE[c]’]	[’L-ASPARTATE[c]’]		False
15289581	ASPARTATE–TRNA-LIGASE-RXN-ASP-tRNAs/L-ASPARTATE/ATP/PROTON//Charged-ASP-tRNAs/AMP/PPI.60.	12	[91]	12	ASPS-CPLX[c]	[’ATP[c]’, ’L-ASPARTATE[c]’]	[’ATP[c]’]	37	True
15289581	ASPARTATE–TRNA-LIGASE-RXN-ASP-tRNAs/L-ASPARTATE/ATP/PROTON//Charged-ASP-tRNAs/AMP/PPI.60.	30	[130]	30	ASPS-CPLX[c]	[’ATP[c]’, ’L-ASPARTATE[c]’]	[’L-ASPARTATE[c]’]	37	True
Truffa-Bachi, P. Microbial aspartokinases The Enzymes, 3rd Ed. (Boyer, P.D., ed.) 8 509-553 1973	ASPARTATEKIN-RXN__ASPKINIHOMOSERDEHYDROGI-CPLX	56.7	[1500, 4000]	113.4	ASPKINIHOMOSERDEHYDROGI-CPLX[c]	[’L-ASPARTATE[c]’, ’ATP[c]’]	[’L-ASPARTATE[c]’, ’ATP[c]’]	27	False
Sheperdson, M.; Pardee, A.B. Production and crystallization of aspartate transcarbamylase J. Biol. Chem. 235 3233-3237 1960	ASPCARBTRANS-RXN	1670	[]	3837	ASPCARBTRANS-CPLX[c]	[’L-ASPARTATE[c]’]	[]		False
16793549	BIOTIN-CARBOXYL-RXN-BCCP-dimers/HCO3/ATP//Carboxybiotin-BCCP/ADP/Pi/PROTON.55.	0.228	[115.2]	0.228	BIOTIN-CARBOXYL-CPLX[c]	[’ATP[c]’]	[’ATP[c]’]	37	False
16480719	BIOTINLIG-RXN-BCCP-monomers/BIOTIN/ATP//AMP/BCCP-biotin-monomers/PPI/PROTON.62.	0.165	[]	0.165	BIOTINLIG-MONOMER[c]	[’BCCP-monomers[c]’]	[]	37	False
10989422	BRANCHED-CHAINAMINOTRANSFERILEU-RXN	48	[420, 2400]	110.3	BRANCHED-CHAINAMINOTRANSFER-CPLX[c]	[’ILE[c]’, ’2-KETOGLUTARATE[c]’]	[’ILE[c]’, ’2-KETOGLUTARATE[c]’]	25	False

10989422	BRANCHED-CHAINAMINOTRANSFERLEU-RXN__BRANCHED-CHAINAMINOTRANSFER-CPLX (reverse)	78	[2200, 6600]	179.2	BRANCHED-CHAINAMINOTRANSFER-CPLX[c]	[’LEU[c]’, ’2-KETOGLUTARATE[c]’]	[’LEU[c]’, ’2-KETOGLUTARATE[c]’]	25	False
10989422	BRANCHED-CHAINAMINOTRANSFERVAL-RXN	19	[2700, 1700]	43.65	BRANCHED-CHAINAMINOTRANSFER-CPLX[c]	[’VAL[c]’, ’2-KETOGLUTARATE[c]’]	[’VAL[c]’, ’2-KETOGLUTARATE[c]’]	25	False
18458150	CARBPSYN-RXN	2.13	[130]	4.893	CARBPSYN-CPLX[c]	[’GLN[c]’]	[’GLN[c]’]	25	False
18458150	CARBPSYN-RXN	4.79	[44]	11	CARBPSYN-CPLX[c]	[’ATP[c]’]	[’ATP[c]’]	25	False
21998098	CATAL-RXN__CPLX0-1683	200	[]	324.9	CPLX0-1683[c]	[’HYDROGEN-PEROXIDE[c]’]	[]	30	False
374409	CATAL-RXN__HYDROPEROXIDI-CPLX	16333	[]	3.752e+04	HYDROPEROXIDI-CPLX[c]	[’HYDROGEN-PEROXIDE[c]’]	[]	25	False
8831972	CHORISMATEMUT-RXN__CHORISMUTPREPHENDEHYDRAT-CPLX	41.4	[300]	41.4	CHORISMUTPREPHENDEHYDRAT-CPLX[c]	[’CHORISMATE[c]’]	[’CHORISMATE[c]’]	37	False
6395895	CHORISMATEMUT-RXN__CHORISMUTPREPHENDEHYDROG-CPLX	182	[92]	182	CHORISMUTPREPHENDEHYDROG-CPLX[c]	[’CHORISMATE[c]’]	[’CHORISMATE[c]’]	37	False
1644758	CHORPYRLY-RXN	0.82	[9.7]	0.82	CHORPYRLY-MONOMER[c]	[’CHORISMATE[c]’]	[’CHORISMATE[c]’]	37	False
12824188	CITSYN-RXN__CITRATE-SI-SYNTHASE	81	[120]	186.1	CITRATE-SI-SYNTHASE[c]	[’ACETYL-COA[c]’]	[’ACETYL-COA[c]’]		False
21669179	CPM-KDOSYNTH-RXN	12.5	[330]	12.5	CPM-KDOSYNTH-MONOMER[c]	[’CTP[c]’]	[’CTP[c]’]	37	False
16427816	CTPSYN-RXN	12.8	[]	12.8	CTPSYN-CPLX[c]	[’ATP[c]’]	[]	37	False
16427816	CTPSYN-RXN	13.7	[]	13.7	CTPSYN-CPLX[c]	[’UTP[c]’]	[]	37	False
12383057	CTPSYN-RXN	6.1	[320]	6.1	CTPSYN-CPLX[c]	[’GLN[c]’]	[’GLN[c]’]	37	False
15489861	CYSTEINE–TRNA-LIGASE-RXN-CYS-tRNAs/CYS/ATP/PROTON//Charged-CYS-tRNAs/AMP/PPI.52.	2.46	[]	2.46	CYSS-MONOMER[c]	[’CYS-tRNAs[c]’]	[]	37	True
15248753	CYTDEAM-RXN	165	[200]	165	CPLX0-7932[c]	[’CYTOSINE[c]’]	[’CYTOSINE[c]’]	37	False
9056491	DAHPSYN-RXN__AROF-CPLX	29.5	[13]	67.77	AROF-CPLX[c]	[’PHOSPHO-ENOL-PYRUVATE[c]’]	[’PHOSPHO-ENOL-PYRUVATE[c]’]	25	False
9387	DAHPSYN-RXN__AROG-CPLX	122	[5.8]	122	AROG-CPLX[c]	[’PHOSPHO-ENOL-PYRUVATE[c]’]	[’PHOSPHO-ENOL-PYRUVATE[c]’]	37	False
9398312	DAHPSYN-RXN__AROH-CPLX	21	[]	48.25	AROH-CPLX[c]	[’PHOSPHO-ENOL-PYRUVATE[c]’]	[]	25	False
1993184	DALADALALIG-RXN__DALADALALIGA-MONOMER	7.4	[]	17	DALADALALIGA-MONOMER[c]	[’D-ALANINE[c]’]	[]		False
1993184	DALADALALIG-RXN__DALADALALIGB-CPLX	17	[]	39.06	DALADALALIGB-CPLX[c]	[’D-ALANINE[c]’]	[]		False
1092682	DAPASYN-RXN	0.283	[200]	0.283	DAPASYN-CPLX[c]	[’S-ADENOSYLMETHIONINE[c]’]	[’S-ADENOSYLMETHIONINE[c]’]	37	False

12805358	DARAB5PISOM-RXN__CPLX0-1262 (reverse)	157	[]	157	CPLX0-1262[c]	['ARABINOSE-5P[c]']	[]	37	False
12805358	DARAB5PISOM-RXN__CPLX0-1262	255	[]	255	CPLX0-1262[c]	['RIBULOSE-5P[c]']	[]	37	False
16199563	DARAB5PISOM-RXN__CPLX0-3929 (reverse)	218	[]	218	CPLX0-3929[c]	['ARABINOSE-5P[c]']	[]	37	False
16199563	DARAB5PISOM-RXN__CPLX0-3929	242	[]	242	CPLX0-3929[c]	['RIBULOSE-5P[c]']	[]	37	False
17996716	DCTP-DEAM-RXN	1.24	[]	1.24	DCTP-DEAM-CPLX[c]	['DCTP[c]']	[]	37	False
11527960	DCYSDESULF-RXN	72	[]	72	DCYSDESULF-CPLX[c]	['D-CYSTEINE[c]']	[]	37	False
Barbas, C.F.; Wang, Y.F.; Wong, C.H. Deoxyribose-5-phosphate aldolase as a synthetic catalyst J. Am. Chem. Soc. 112 2013-2014 1990	DEOXYRIBOSE-P-ALD-RXN (reverse)	521.1	[]	1197	DEOXYRIBOSE-P-ALD-MONOMER[c]	['DEOXY-RIBOSE-5P[c]']	[]	25	False
2826481	DGTPTRIPHYDRO-RXN	4000	[10]	4000	DGTPTRIPHYDRO-CPLX[c]	['DGTP[c]']	['DGTP[c]']	37	False
2531000	DHBAMPLIG-RXN	5.5	[2.7]	5.5	ENTE-CPLX[i]	['2-3-DIHYDROXYBENZOATE[c]']	['2-3-DIHYDROXYBENZOATE[c]']	37	False
2531000	DHBAMPLIG-RXN	5.83	[1120]	5.83	ENTE-CPLX[i]	['ATP[c]']	['ATP[c]']	37	False
2144454	DHBDEHYD-RXN	92.5	[]	92.5	ENTA-CPLX[c]	['DIHYDRO-DIOH-BENZOATE[c]']	[]	37	False
2689171	DIAMINOPIMEPIM-RXN	132	[]	303.3	CPLX0-7997[c]	['LL-DIAMINOPIMELATE[c]']	[]	25	False
6378903	DIAMINOPIMEPIM-RXN (reverse)	67	[]	153.9	CPLX0-7997[c]	['MESO-DIAMINOPIMELATE[c]']	[]	25	False
15066435	DIHYDRODIPICSYN-RXN	124	[]	201.4	DIHYDRODIPICSYN-CPLX[c]	['PYRUVATE[c]', 'L-ASPARTATE-SEMIALDEHYDE[c]']	[]	30	False
6142052	DIHYDROOROT-RXN	127	[75.6]	206.3	DIHYDROOROT-CPLX[c]	['DI-H-OROTATE[c]']	['DI-H-OROTATE[c]']	30	False
6142052	DIHYDROOROT-RXN (reverse)	195	[1070]	316.8	DIHYDROOROT-CPLX[c]	['CARBAMYUL-L-ASPARTATE[c]']	['CARBAMYUL-L-ASPARTATE[c]']	30	False
8969520	DIMESULFREDUCT-RXN-CPD-7670/CPD-9728/WATER//DMSO/REDUCED-MENAQUINONE.50.	79.9	[]	129.8	DIMESULFREDUCT-CPLX[i]	['DMSO[p]']	[]	30	False
11380254	DTDPGLUCDEHYDRAT-RXN__DTDPGLUCDEHYDRAT-CPLX	4.9	[]	4.9	DTDPGLUCDEHYDRAT-CPLX[c]	['DTDP-D-GLUCOSE[c]']	[]	37	False
346589	DUTP-PYROP-RXN__DUTP-PYROP-CPLX	58.33	[]	94.76	DUTP-PYROP-CPLX[c]	['DUTP[c]']	[]	30	False
16766526	DUTP-PYROP-RXN__G7164-MONOMER	22	[]	22	G7164-MONOMER[c]	['DUTP[c]']	[]	37	False

15938625	DXPREDISOM-RXN (reverse)	29	[0.5]	29	DXPREDISOM-CPLX[c]	[’DEOXYXYLULOSE-5P[c]’, ’NADPH[c]’]	[’NADPH[c]’]	37	False
21119630	DXS-RXN	345	[]	345	CPLX0-743[c]	[’PYRUVATE[c]’, ’GAP[c]’]	[]	37	False
7751290	ERYTH4PDEHYDROG-RXN	169	[74]	169	ERYTH4PDEHYDROG-CPLX[c]	[’NAD[c]’]	[’NAD[c]’]	37	False
15531627	F16ALDOLASE-RXN__FRUCBISALD-CLASSII (reverse)	14.17	[140]	23.02	FRUCBISALD-CLASSII[c]	[’FRUCTOSE-16-DIPHOSPHATE[c]’]	[’FRUCTOSE-16-DIPHOSPHATE[c]’]	30	False
9878448	F16ALDOLASE-RXN__FRUCBISALD-CLASSI (reverse)	0.35	[20]	0.5686	FRUCBISALD-CLASSI[c]	[’FRUCTOSE-16-DIPHOSPHATE[c]’]	[’FRUCTOSE-16-DIPHOSPHATE[c]’]	30	False
19073594	F16BDEPHOS-RXN[CCO-CYTOSOL]-FRUCTOSE-16-DIPHOSPHATE/WATER//FRUCTOSE-6P/Pi.59.__CPLX0-303	5.7	[70]	5.7	CPLX0-303[c]	[’FRUCTOSE-16-DIPHOSPHATE[c]’]	[’FRUCTOSE-16-DIPHOSPHATE[c]’]	37	False
19073594	F16BDEPHOS-RXN[CCO-CYTOSOL]-FRUCTOSE-16-DIPHOSPHATE/WATER//FRUCTOSE-6P/Pi.59.__CPLX0-7776	2.5	[100]	2.5	CPLX0-7776[c]	[’FRUCTOSE-16-DIPHOSPHATE[c]’]	[’FRUCTOSE-16-DIPHOSPHATE[c]’]	37	False
19073594	F16BDEPHOS-RXN[CCO-CYTOSOL]-FRUCTOSE-16-DIPHOSPHATE/WATER//FRUCTOSE-6P/Pi.59.__EG11239-MONOMER	5.3	[2400]	5.3	EG11239-MONOMER[c]	[’FRUCTOSE-16-DIPHOSPHATE[c]’]	[’FRUCTOSE-16-DIPHOSPHATE[c]’]	37	False
19073594	F16BDEPHOS-RXN[CCO-CYTOSOL]-FRUCTOSE-16-DIPHOSPHATE/WATER//FRUCTOSE-6P/Pi.59.__F16B-CPLX	14.6	[20]	14.6	F16B-CPLX[c]	[’FRUCTOSE-16-DIPHOSPHATE[c]’]	[’FRUCTOSE-16-DIPHOSPHATE[c]’]	37	False
2659070	FGAMSYN-RXN	0.05	[64, 51]	0.05	FGAMSYN-MONOMER[c]	[’GLN[c]’, ’ATP[c]’]	[’GLN[c]’, ’ATP[c]’]	37	False
1099093	FORMATEDEHYDROG-RXN-FORMATE/CPD-9728/PROTON//CARBON-DIOXIDE/REDUCED-MENAQUINONE/PROTON.67.__FORMATEDEHYDROGN-CPLX	563.3	[]	915.1	FORMATEDEHYDROGN-CPLX[i]	[’FORMATE[p]’]	[]	30	False
15157072	FRUCTOKINASE-RXN	13	[1800]	29.87	EG11288-MONOMER[c]	[’BETA-D-FRUCTOSE[c]’, ’ATP[c]’]	[’ATP[c]’]	25	False
10821675	FUCPALDOL-RXN (reverse)	19.3	[]	19.3	CPLX0-7633[c]	[’FUCULOSE-1P[c]’]	[]	37	False
8422384	FUMHYDR-RXN__FUMARASE-A	3000	[600]	6892	FUMARASE-A[c]	[’FUM[c]’]	[’FUM[c]’]		False
1329945	FUMHYDR-RXN__FUMARASE-A (reverse)	3100	[700]	7122	FUMARASE-A[c]	[’MAL[c]’]	[’MAL[c]’]	25	False
8203917	FUMHYDR-RXN__FUMARASE-C	3100	[620]	7122	FUMARASE-C[c]	[’FUM[c]’]	[’FUM[c]’]	25	False
12021453	FUMHYDR-RXN__FUMARASE-C (reverse)	595.2	[857]	1367	FUMARASE-C[c]	[’MAL[c]’]	[’MAL[c]’]		False

15723541	GABATRANSAM-RXN__G6646-MONOMER	47.4	[]	108.9	G6646-MONOMER[c]	[’4-AMINO-BUTYRATE[c]’]	[]	25	False
10820011	GALACTURIDYLYLTRANS-RXN	283	[]	566	GALACTURIDYLYLTRANS-CPLX[c]	[’CPD-14553[c]’, ’GLC-1-P[c]’]	[]	27	False
7873670	GALACTURIDYLYLTRANS-RXN (reverse)	960	[200]	1920	GALACTURIDYLYLTRANS-CPLX[c]	[’GALACTOSE-1P[c]’, ’CPD-12575[c]’]	[’CPD-12575[c]’]	27	False
Zhang, Q.; Liu, H.w. Studies of UDP-galactopyranose mutase from Escherichia coli: An unusual role of reduced FAD in its catalysis J. Am. Chem. Soc. 122 9065-9070 2000	GALPMUT-RXN (reverse)	27	[]	27	GALPMUT-MONOMER[c]	[’UDP-D-GALACTO-14-FURANOSE[c]’]	[]	37	False
19384989	GARTRANSFORMYL2-RXN	37.6	[45]	86.38	GARTRANSFORMYL2-MONOMER[c]	[’5-PHOSPHO-RIBOSYL-GLYCINEAMIDE[c]’, ’ATP[c]’, ’FORMATE[c]’]	[’ATP[c]’]	25	False
9772162	GKI-RXN__GKI-MONOMER	2.5	[61, 51]	7.071	GKI-MONOMER[c]	[’ATP[c]’, ’GLYCERATE[c]’]	[’ATP[c]’, ’GLYCERATE[c]’]	22	False
17079236	GLUC1PADENYLTRANS-RXN	370	[590]	370	GLUC1PADENYLTRANS-CPLX[c]	[’ATP[c]’, ’GLC-1-P[c]’]	[’ATP[c]’]	37	False
15157072	GLUCOKIN-RXN	410	[260]	941.9	GLUCOKIN-MONOMER[c]	[’GLC[c]’, ’ATP[c]’]	[’ATP[c]’]	25	False
25484615	GLUCOSE-1-PHOSPHAT-RXN[CCO-CYTOSOL]-GLC-1-P/WATER//Glucopyranose/Pi.45.__EG11850-MONOMER	10.2	[]	10.2	EG11850-MONOMER[c]	[’GLC-1-P[c]’]	[]	37	False
16990279	GLUCOSE-6-PHOSPHATASE-RXN__EG11470-MONOMER	14	[]	32.16	EG11470-MONOMER[c]	[’GLC-6-P[c]’]	[]		False
19678710	GLUCUROISOM-RXN-CPD-15530//CPD-12537.21. (reverse)	196	[]	318.4	UXAC-MONOMER[c]	[’CPD-15530[c]’]	[]	30	False
3003503	GLURS-RXN-GLT-tRNAs/GLT/ATP/PROTON//Charged-GLT-tRNAs/AMP/PPI.52.	0.8	[200, 100]	0.8	GLURS-MONOMER[c]	[’GLT[c]’, ’ATP[c]’]	[’GLT[c]’, ’ATP[c]’]	37	False
20853825	GLUTAMIN-RXN__ASNSYNB-CPLX	6.6	[1700]	6.6	ASNSYNB-CPLX[c]	[’GLN[c]’]	[’GLN[c]’]	37	False
18459799	GLUTAMIN-RXN__CPLX0-7694	91.4	[7300]	210	CPLX0-7694[c]	[’GLN[c]’]	[’GLN[c]’]		False
18459799	GLUTAMIN-RXN__CPLX0-7695	101	[30600]	232	CPLX0-7695[c]	[’GLN[c]’]	[’GLN[c]’]		False
12383057	GLUTAMIN-RXN__CTPSYN-CPLX	4.98	[]	4.98	CTPSYN-CPLX[c]	[’GLN[c]’]	[]	37	False

9657697	GLUTAMINE–TRNA-LIGASE-RXN-GLN-tRNAs/GLN/ATP/PROTON//Charged-GLN-tRNAs/AMP/PPI.52.	2.62	[114]	2.62	GLNS-MONOMER[c]	[’GLN[c]’]	[’GLN[c]’]	37	True
9657697	GLUTAMINE–TRNA-LIGASE-RXN-GLN-tRNAs/GLN/ATP/PROTON//Charged-GLN-tRNAs/AMP/PPI.52.	2.8	[110]	2.8	GLNS-MONOMER[c]	[’ATP[c]’]	[’ATP[c]’]	37	True
15797252	GLUTAMINESYN-RXN	36	[200, 3300]	82.71	GLUTAMINESYN-OLIGOMER[c]	[’ATP[c]’, ’GLT[c]’]	[’ATP[c]’, ’GLT[c]’]		False
1924337	GLUTATHIONE-REDUCT-NADPH-RXN	733.3	[]	1191	GLUTATHIONE-REDUCT-NADPH-CPLX[c]	[’OXIDIZED-GLUTATHIONE[c]’]	[]	30	True
19797049	GLUTDECARBOX-RXN__GLUTDECARBOXB-CPLX	24.85	[2320]	24.85	GLUTDECARBOXB-CPLX[c]	[’GLT[c]’]	[’GLT[c]’]	37	False
23879525	GLUTDEHYD-RXN__GDHA-CPLX	37.3	[18.4]	85.69	GDHA-CPLX[c]	[’NADP[c]’]	[’NADP[c]’]		False
17568739	GLUTRACE-RXN	12.17	[1200]	27.96	GLUTRACE-MONOMER[c]	[’GLT[c]’]	[’GLT[c]’]		False
17568739	GLUTRACE-RXN (reverse)	43.33	[]	99.55	GLUTRACE-MONOMER[c]	[’D-GLT[c]’]	[]		False
12370189	GLUTRNAREDUCT-RXN-GLUTAMATE-1-SEMIALDEHYDE/GLT-tRNAs/NADP//Charged-GLT-tRNAs/NADPH.65.	0.15	[39]	0.15	CPLX0-3741[c]	[’Charged-GLT-tRNAs[c]’, ’NADPH[c]’]	[’NADPH[c]’]	37	False
Thorner, J.W.; Paulus, H. Glycerol and glycerate kinases The Enzymes,3rd ed.(Boyer,P.D.,ed.) 8 487-508 1973	GLYCEROL-KIN-RXN	193.3	[]	444.1	GLYCEROL-KIN-CPLX[c]	[’GLYCEROL[c]’]	[]	25	False
16731973	GLYCOLALD-DEHYDROG-RXN	18.33	[]	42.11	ALD-CPLX[c]	[’GLYCOLALDEHYDE[c]’]	[]		False
21059411	GLYOHMETRANS-RXN-SER/THF//GLY/METHYLENE-THF/WATER.33.	10.67	[300]	17.33	GLYOHMETRANS-CPLX[c]	[’SER[c]’]	[’SER[c]’]	30	False
14556652	GLYOXI-RXN	388	[]	891.4	GLYOXI-CPLX[c]	[’METHYL-GLYOXAL[c]’]	[]		False
16567800	GLYOXII-RXN__CPLX0-3954	1.02	[]	1.02	CPLX0-3954[c]	[’S-LACTOYL-GLUTATHIONE[c]’]	[]	37	False
22173092	GLYOXII-RXN__G6475-MONOMER	61	[]	140.1	G6475-MONOMER[c]	[’S-LACTOYL-GLUTATHIONE[c]’]	[]	25	False
21696459	GLYOXIII-RXN__CPLX0-861	2.62	[]	2.62	CPLX0-861[c]	[’METHYL-GLYOXAL[c]’]	[]	37	False
11237876	GLYOXYLATE-REDUCTASE-NADP+-RXN__CPLX0-235	203	[]	466.4	CPLX0-235[i]	[’GLYOX[c]’]	[]	25	True
21298178	GMP-REDUCT-RXN	0.28	[5.5, 14.7]	0.6433	GMP-REDUCT-CPLX[c]	[’GMP[c]’, ’NADPH[c]’]	[’GMP[c]’, ’NADPH[c]’]	25	False

7775463	GSPAMID-RXN	2.1	[]	2.1	GSP-CPLX[c]	['GLUTATHIONYLSPERMIDINE[c]']	[]	37	False
7775463	GSPSYN-RXN	7	[60]	7	GSP-CPLX[c]	['SPERMIDINE[c]']	['SPERMIDINE[c]']	37	False
9211333	GTP-CYCLOHYDRO-II-RXN	0.15	[30]	0.15	GTP-CYCLOHYDRO-II-CPLX[c]	['GTP[c]']	['GTP[c]']	37	False
16766526	GUANOSINE-DIPHOSPHATASE-RXN__G6580-MONOMER	3.4	[640]	3.4	G6580-MONOMER[c]	['GDP[c]']	['GDP[c]']	37	False
9100006	GUANPRIBOSYLTRAN-RXN__GPT-CPLX	112	[4.3]	257.3	GPT-CPLX[i]	['GUANINE[c]', 'PRPP[c]']	['GUANINE[c]']	25	False
17176045	H2NEOPTERINALDOL-RXN	0.082	[]	0.1884	CPLX0-3936[c]	['DIHYDRO-NEO-PTERIN[c]']	[]	25	False
17698004	H2NEOPTERINP3PYROPHOSPHOHYDRO-RXN	11.6	[]	11.6	H2NEOPTERINP3PYROPHOSPHOHYDRO-MONOMER[c]	['DIHYDRONEOPTERIN-P3[c]']	[]	37	False
9266856	HISTIDINE–TRNA-LIGASE-RXN-HIS-tRNAs/HIS/ATP/PROTON//Charged-HIS-tRNAs/AMP/PPI.52.	34	[560]	34	HISS-CPLX[c]	['ATP[c]']	['ATP[c]']	37	True
9266856	HISTIDINE–TRNA-LIGASE-RXN-HIS-tRNAs/HIS/ATP/PROTON//Charged-HIS-tRNAs/AMP/PPI.52.	7	[8]	7	HISS-CPLX[c]	['HIS[c]']	['HIS[c]']	37	True
16966333	HISTIDPHOS-RXN[CCO-CYTOSOL]-L-HISTIDINOL-P/WATER//HISTIDINOL/Pi.49.	2140	[]	4916	IMIDHISTID-CPLX[c]	['L-HISTIDINOL-P[c]']	[]	25	False
9211277	HOLO-ACP-SYNTH-RXN-apo-ACP/CO-A//3-5-ADP/ACP/PROTON.33.__HOLO-ACP-SYNTH-CPLX	1.17	[50]	1.17	HOLO-ACP-SYNTH-CPLX[c]	['apo-ACP[c]', 'CO-A[c]']	['CO-A[c]']	37	False
Truffa-Bachi, P. Microbial aspartokinases The Enzymes, 3rd Ed. (Boyer, P.D., ed.) 8 509-553 1973	HOMOSERDEHYDROG-RXN-HOMO-SER/NADP//L-ASPARTATE-SEMIALDEHYDE/NADPH/PROTON.53.__ASPKINIHOMOSERDEHYDROGI-CPLX	400	[120]	800	ASPKINIHOMOSERDEHYDROGI-CPLX[c]	['L-ASPARTATE-SEMIALDEHYDE[c]', 'NADPH[c]']	['NADPH[c]']	27	False
8973190	HOMOSERKIN-RXN	18.3	[130]	29.73	HOMOSERKIN-CPLX[c]	['HOMO-SER[c]', 'ATP[c]']	['ATP[c]']	30	False
17442255	HOMSUCTRAN-RXN	25.7	[280]	59.04	HOMSUCTRAN-CPLX[c]	['SUC-COA[c]', 'HOMO-SER[c]']	['SUC-COA[c]']	25	False
9100006	HYPOXANPRIBOSYLTRAN-RXN__GPT-CPLX	54.8	[]	125.9	GPT-CPLX[i]	['HYPOXANTHINE[c]']	[]	25	False
2494074	IGPSYN-RXN	7.2	[]	7.2	PRAI-IGPS[c]	['CARBOXYPHENYLAMINO-DEOXYRIBULOSE-P[c]']	[]	37	False
18990827	IMP-DEHYDROG-RXN	13	[2000]	29.87	IMP-DEHYDROG-CPLX[c]	['IMP[c]', 'NAD[c]']	['NAD[c]']		False
21279421	INOPHOSPHOR-RXN__DEOD-CPLX	6200	[]	6200	DEOD-CPLX[c]	['INOSINE[c]']	[]	37	False
14640961	INORGPYROPHOSPHAT-RXN[CCO-CYTOSOL]-PPI/WATER//Pi/PROTON.34.	390	[0.13]	896	CPLX0-243[c]	['PPI[c]']	['PPI[c]']	25	True

2139795	ISOCHORSYN-RXN__ENTC-MONOMER	1.8	[]	1.8	ENTC-MONOMER[c]	[’ISOCHORISMATE[c]’]	[]	37	False
2139795	ISOCHORSYN-RXN__ENTC-MONOMER (reverse)	2.88	[]	2.88	ENTC-MONOMER[c]	[’CHORISMATE[c]’]	[]	37	False
17240978	ISOCHORSYN-RXN__MENF-CPLX (reverse)	3.55	[]	3.55	MENF-CPLX[c]	[’CHORISMATE[c]’]	[]	37	False
17401542	ISOCITDEH-RXN	106.3	[]	86.34	ISOCITHASE-CPLX[c]	[’THREO-DS-ISO-CITRATE[c]’]	[]	40	True
17401542	ISOCITDEH-RXN	88.1	[39.2]	71.56	ISOCITHASE-CPLX[c]	[’NADP[c]’]	[’NADP[c]’]	40	True
8672449	ISOLEUCINE–TRNA-LIGASE-RXN-ILE-tRNAs/ILE/ATP/PROTON//Charged-ILE-tRNAs/AMP/PPI.52.	80.4	[3.6, 280]	80.4	ILES-MONOMER[c]	[’ILE[c]’, ’ATP[c]’]	[’ILE[c]’, ’ATP[c]’]	37	True
9772162	KDGALDOL-RXN	27.3	[]	77.22	CPLX0-7615[c]	[’5-KETO-4-DEOXY-D-GLUCARATE[c]’]	[]	22	False
15308670	KDO-8PSYNTH-RXN	6.1	[6]	6.1	KDO-8PSYNTH-CPLX[c]	[’PHOSPHO-ENOL-PYRUVATE[c]’, ’ARABINOSE-5P[c]’]	[’PHOSPHO-ENOL-PYRUVATE[c]’]	37	False
11094340	KDPGALDOL-RXN	283.8	[]	652	KDPGALDOL-4OH2OXOGLUTARALDOL-CPLX[c]	[’2-KETO-3-DEOXY-6-P-GLUCONATE[c]’]	[]	25	False
8706750	L-ASPARTATE-OXID-RXN	2.6	[5200]	5.973	L-ASPARTATE-OXID-MONOMER[c]	[’L-ASPARTATE[c]’, ’OXYGEN-MOLECULE[c]’]	[’L-ASPARTATE[c]’]	25	False
20416269	L-GLN-FRUCT-6-P-AMINOTRANS-RXN-FRUCTOSE-6P/GLN//CPD-13469/GLT.31.	12	[230]	27.57	L-GLN-FRUCT-6-P-AMINOTRANS-CPLX[c]	[’GLT[c]’]	[’GLT[c]’]	25	False
17559838	L-GLN-FRUCT-6-P-AMINOTRANS-RXN-FRUCTOSE-6P/GLN//CPD-13469/GLT.31. (reverse)	14.2	[]	14.2	L-GLN-FRUCT-6-P-AMINOTRANS-CPLX[c]	[’FRUCTOSE-6P[c]’]	[]	37	False
18754693	L-RHAMNONATE-DEHYDRATASE-RXN (reverse)	3.2	[]	7.352	CPLX0-7722[c]	[’L-RHAMNONATE[c]’]	[]	25	False
15110746	LEUCINE–TRNA-LIGASE-RXN-LEU-tRNAs/LEU/ATP/PROTON//Charged-LEU-tRNAs/AMP/PPI.52.	5	[240]	5	LEUS-MONOMER[c]	[’ATP[c]’]	[’ATP[c]’]	37	True
15110746	LEUCINE–TRNA-LIGASE-RXN-LEU-tRNAs/LEU/ATP/PROTON//Charged-LEU-tRNAs/AMP/PPI.52.	5.1	[20]	5.1	LEUS-MONOMER[c]	[’LEU[c]’]	[’LEU[c]’]	37	True
21059411	LTAA-RXN__GLYOHMETRANS-CPLX	0.5	[]	0.8123	GLYOHMETRANS-CPLX[c]	[’L-ALLO-THREONINE[c]’]	[]	30	False
21059411	LTAA-RXN__LTAA-CPLX	6.27	[]	10.19	LTAA-CPLX[c]	[’L-ALLO-THREONINE[c]’]	[]	30	False
11741871	LXULRU5P-RXN__G7859-MONOMER	2.9	[]	6.662	G7859-MONOMER[c]	[’L-XYLULOSE-5-P[c]’]	[]	25	False

21278708	LYSDECARBOX-RXN__LYSDECARBOX-CPLX	30	[368]	295.5	LYSDECARBOX-CPLX[c]	[’LYS[c]’]	[’LYS[c]’]	4	False
15362869	LYSINE–TRNA-LIGASE-RXN-LYS/LYS-tRNAs/ATP/PROTON//Charged-LYS-tRNAs/AMP/PPI.52.__LYSS-CPLX	1.8	[2.6]	1.8	LYSS-CPLX[c]	[’ATP[c]’]	[’ATP[c]’]	37	True
10387054	LYSINE–TRNA-LIGASE-RXN-LYS/LYS-tRNAs/ATP/PROTON//Charged-LYS-tRNAs/AMP/PPI.52.__LYSS-CPLX	6.1	[2.7]	6.1	LYSS-CPLX[c]	[’LYS[c]’]	[’LYS[c]’]	37	True
14597191	LYXK-RXN	110	[]	110	LYXK-CPLX[c]	[’L-XYLULOSE[c]’]	[]	37	False
17947381	MALATE-DEH-RXN (reverse)	931	[]	1512	MALATE-DEHASE[c]	[’OXALACETIC_ACID[c]’]	[]	30	False
17557829	MALIC-NADP-RXN	66.6	[3410, 41.5]	108.2	MALIC-NADP-CPLX[c]	[’MAL[c]’, ’NADP[c]’]	[’MAL[c]’, ’NADP[c]’]	30	False
11814333	MALONYL-COA-ACP-TRANSACYL-RXN-ACP/MALONYL-COA//MALONYL-ACP/CO-A.34. (reverse)	1580	[]	3630	MALONYL-COA-ACP-TRANSACYL-MONOMER[c]	[’ACP[c]’, ’MALONYL-COA[c]’]	[]		False
2111176	MANNPDEHYDROG-RXN (reverse)	190	[200]	436.5	MANNPDEHYDROG-MONOMER[c]	[’MANNITOL-1P[c]’, ’NAD[c]’]	[’NAD[c]’]	25	False
2111176	MANNPDEHYDROG-RXN	240	[10]	551.4	MANNPDEHYDROG-MONOMER[c]	[’FRUCTOSE-6P[c]’, ’NADH[c]’]	[’NADH[c]’]	25	False
19370061	METBALT-RXN	1.8	[]	4.135	O-SUCCHOMOSERLYASE-CPLX[c]	[’O-SUCCINYL-L-HOMOSERINE[c]’]	[]	25	False
15049687	METHGLYSYN-RXN	220	[200]	505.4	METHGLYSYN-CPLX[c]	[’DIHYDROXY-ACETONE-PHOSPHATE[c]’]	[’DIHYDROXY-ACETONE-PHOSPHATE[c]’]	25	False
2126467	METHIONINE–TRNA-LIGASE-RXN-Elongation-tRNAMet/MET/ATP/PROTON//Charged-MET-tRNAs/AMP/PPI.61.	3.3	[]	3.3	METG-CPLX[c]	[’Elongation-tRNAMet[c]’]	[]	37	True
9391059	METHIONYL-TRNA-FORMYLTRANSFERASE-RXN-10-FORMYL-THF/L-methionyl-tRNAfmet//THF/N-formyl-L-methionyl-tRNAfmet.70.	41.52	[]	41.52	EG11268-MONOMER[c]	[’L-methionyl-tRNAfmet[c]’]	[]	37	False
10769117	METHYLMALONYL-COA-MUT-RXN (reverse)	0.2	[11.2]	0.2	CPLX0-7741[c]	[’SUC-COA[c]’]	[’SUC-COA[c]’]	37	False
17059210	METHYLTHIOADENOSINE-NUCLEOSIDASE-RXN	4	[]	9.19	CPLX0-1541[c]	[’5-METHYLTHIOADENOSINE[c]’]	[]	25	False
20364820	MHPELY-RXN (reverse)	205.4	[]	471.9	MHPELY-MONOMER[c]	[’ACETALD[c]’]	[]	25	False
10026151	MMUM-RXN	0.38	[45]	0.6173	MMUM-MONOMER[c]	[’HOMO-CYS[c]’]	[’HOMO-CYS[c]’]	30	False

18081839	NACGLCTRANS-RXN	0.93	[150]	0.93	NACGLCTRANS-MONOMER[c]	['UDP-N-ACETYL-D-GLUCOSAMINE[c]']	['UDP-N-ACETYL-D-GLUCOSAMINE[c]']	37	False
11488932	NAD-KIN-RXN	125	[2000]	125	CPLX0-682[c]	['NAD[c]']	['NAD[c]']	37	False
11488932	NAD-KIN-RXN	55	[2500]	55	CPLX0-682[c]	['ATP[c]']	['ATP[c]']	37	False
21832062	NADH-DEHYDROG-A-RXN-NADH/UBIQUINONE-8/PROTON//NAD/CPD-9956/PROTON.46. (reverse)	26	[13]	42.24	NADH-DHI-CPLX[i]	['NADH[c]']	['NADH[c]']	30	True
8083170	NAG1P-URIDYLTRANS-RXN	12.4	[70, 100]	12.4	NAG1P-URIDYLTRANS-CPLX[c]	['N-ACETYL-D-GLUCOSAMINE-1-P[c]', 'UTP[c]']	['N-ACETYL-D-GLUCOSAMINE-1-P[c]', 'UTP[c]']	37	False
15157072	NANK-RXN-N-ACETYL-D-MANNOSAMINE/ATP//N-ACETYL-D-MANNOSAMINE-6P/ADP/PROTON.65.	83	[510]	190.7	NANK-MONOMER[c]	['N-ACETYL-D-MANNOSAMINE[c]', 'ATP[c]']	['ATP[c]']	25	False
24530526	NMNAMIDOHYDRO-RXN	4.1	[]	4.1	CPLXECOLI-7943[c]	['NICOTINAMIDE_NUCLEOTIDE[c]']	[]	37	False
19370061	O-SUCCHOMOSERLYASE-RXN	121	[640]	278	O-SUCCHOMOSERLYASE-CPLX[c]	['O-SUCCINYL-L-HOMOSERINE[c]', 'CYS[c]']	['CYS[c]']	25	False
10757968	OROTPDECARB-RXN	28	[]	64.33	OROTPDECARB-CPLX[c]	['OROTIDINE-5-PHOSPHATE[c]']	[]		False
8422384	OXALOACETATE-TAUTOMERASE-RXN (reverse)	300	[]	689.2	FUMARASE-A[c]	['ENOL-OXALOACETATE[c]']	[]		False
20553497	OXALYL-COA-DECARBOXYLASE-RXN	60.7	[]	98.61	CPLX0-7878[c]	['OXALYL-COA[c]']	[]	30	False
20170126	PABASYN-RXN__PABASYN-CPLX (reverse)	0.67	[520]	1.539	PABASYN-CPLX[c]	['GLN[c]']	['GLN[c]']	25	False
14982443	PABASYN-RXN__PABSYNMULTI-CPLX (reverse)	0.53	[13]	1.218	PABSYNMULTI-CPLX[c]	['CHORISMATE[c]']	['CHORISMATE[c]']	25	False
16990935	PANTOATE-BETA-ALANINE-LIG-RXN	1.4	[1750]	3.216	PANTOATE-BETA-ALANINE-LIG-CPLX[c]	['ATP[c]']	['ATP[c]']	25	False
19307712	PANTOTHENATE-KIN-RXN	1.14	[115]	3.704	PANTOTHENATE-KIN-CPLX[c]	['ATP[c]']	['ATP[c]']	20	False
9927652	PEPCARBOX-RXN	150	[190]	243.7	PEPCARBOX-CPLX[c]	['PHOSPHO-ENOL-PYRUVATE[c]']	['PHOSPHO-ENOL-PYRUVATE[c]']	30	False
8550422	PGLYCDEHYDROG-RXN	27.8	[]	27.8	PGLYCDEHYDROG-CPLX[c]	['3-P-HYDROXYPYRUVATE[c]']	[]	37	False
11736651	PHEAMINOTRANS-RXN__ASPAMINOTRANS-DIMER	13.73	[8000]	5.576	ASPAMINOTRANS-DIMER[c]	['PHE[c]']	['PHE[c]']	50	False
22138634	PHEAMINOTRANS-RXN__ASPAMINOTRANS-DIMER (reverse)	670	[37000]	1539	ASPAMINOTRANS-DIMER[c]	['GLT[c]']	['GLT[c]']	25	False
10989422	PHEAMINOTRANS-RXN__BRANCHED-CHAINAMINOTRANSFER-CPLX	2.9	[890, 260]	6.662	BRANCHED-CHAINAMINOTRANSFER-CPLX[c]	['PHE[c]', '2-KETOGLUTARATE[c]']	['PHE[c]', '2-KETOGLUTARATE[c]']	25	False

16731973	PHENDEHYD-RXN (reverse)	96.83	[]	222.5	PHENDEHYD-CPLX[c]	['PHENYLACETALDEHYDE[c]']	[]		False
20160120	PHENYLALANINE–TRNA-LIGASE-RXN-PHE-tRNAs/PHE/ATP/PROTON//Charged-PHE-tRNAs/AMP/PPI.52.	199	[2]	199	PHES-CPLX[c]	['PHE[c]']	['PHE[c]']	37	True
21046341	PHOSACETYLTRANS-RXN__CPLX0-7912	119.8	[9.5]	194.6	CPLX0-7912[c]	['ACETYL-COA[c]']	['ACETYL-COA[c]']	30	False
21046341	PHOSACETYLTRANS-RXN__CPLX0-7912 (reverse)	415.5	[311.7, 32.7]	675	CPLX0-7912[c]	['ACETYL-P[c]', 'CO-A[c]']	['ACETYL-P[c]', 'CO-A[c]']	30	False
21046341	PHOSACETYLTRANS-RXN__PHOSACETYLTRANS-CPLX (reverse)	227.6	[900, 67.2]	369.7	PHOSACETYLTRANS-CPLX[c]	['ACETYL-P[c]', 'CO-A[c]']	['ACETYL-P[c]', 'CO-A[c]']	30	False
21046341	PHOSACETYLTRANS-RXN__PHOSACETYLTRANS-CPLX	29.6	[44.9]	48.09	PHOSACETYLTRANS-CPLX[c]	['ACETYL-COA[c]']	['ACETYL-COA[c]']	30	False
7696318	PMPOXI-RXN	28	[]	19.8	PDXH-CPLX[c]	['PYRIDOXAMINE-5P[c]']	[]	42	False
7860596	PNPOXI-RXN	0.76	[]	0.76	PDXH-CPLX[c]	['PYRIDOXINE-5P[c]']	[]	37	False
8394006	PPPGPPHYDRO-RXN	0.023	[]	0.023	PPPGPPHYDRO-CPLX[c]	['GDP-TP[c]']	[]	37	False
7727400	PRAISOM-RXN	40	[]	91.9	PRAI-IGPS[c]	['N-5-PHOSPHORIBOSYL-ANTHRANILATE[c]']	[]	25	False
10769128	PREPHENATEDEHYDRAT-RXN	32.2	[]	32.2	CHORISMUTPREPHENDEHYDRAT-CPLX[c]	['PREPHENATE[c]']	[]	37	False
6395895	PREPHENATEDEHYDROG-RXN	135	[130]	135	CHORISMUTPREPHENDEHYDROG-CPLX[c]	['PREPHENATE[c]', 'NAD[c]']	['NAD[c]']	37	False
12356303	PRIBFAICARPISOM-RXN	14.3	[]	14.3	PRIBFAICARPISOM-MONOMER[c]	['PHOSPHORIBOSYL-FORMIMINO-AICAR-P[c]']	[]	37	False
16864571	PROLINE–TRNA-LIGASE-RXN-PRO-tRNAs/PRO/ATP/PROTON//Charged-PRO-tRNAs/AMP/PPI.52.	70	[250]	70	PROS-CPLX[c]	['PRO[c]']	['PRO[c]']	37	True
19583219	PROTOPORGENOXI-RXN	0.292	[]	0.292	CPLX0-7811[c]	['PROTOPORPHYRINOGEN[c]']	[]	37	False
2651124	PRTRANS-RXN__ANTHRANSYN-CPLX	4.4	[0.28]	10.11	ANTHRANSYN-CPLX[c]	['ANTHRANILATE[c]']	['ANTHRANILATE[c]']	25	False
8706854	PSERTRANSAM-RXN (reverse)	0.39	[]	0.39	PSERTRANSAM-CPLX[c]	['3-P-SERINE[c]']	[]	37	False
8706854	PSERTRANSAM-RXN	1.75	[]	1.75	PSERTRANSAM-CPLX[c]	['3-P-HYDROXYPYRUVATE[c]']	[]	37	False
8706854	PSERTRANSAMPYR-RXN (reverse)	0.15	[]	0.15	PSERTRANSAM-CPLX[c]	['4-PHOSPHONOOXY-THREONINE[c]']	[]	37	False
15249053	PYRAMKIN-RXN	0.67	[]	0.67	PDXK-CPLX[c]	['PYRIDOXAMINE[c]']	[]	37	False
15249053	PYRIDOXKIN-RXN__PDXK-CPLX	2.33	[]	2.33	PDXK-CPLX[c]	['PYRIDOXAL[c]']	[]	37	False

Reddick, J.J.; Kinsland, C.; Nicewonger, R.; Christian, T.; Downs, D.M.; Winkler, M.E.; Begley, T.P. Overexpression, purification and characterization of two pyrimidine kinases involved in the biosynthesis of thiamin: 4-amino-5-hydroxymethyl-2-methylpyrimidine kinase and 4-amino-5-hydroxymethyl-2-methylpyrimidine phosphate kinase Tetrahedron 54 15983-15991 1998	PYRIMSYN3-RXN	0.07	[193.0]	0.1608	HMP-P-KIN-CPLX[c]	[’ATP[c]’]	[’ATP[c]’]	25	False
16829675	R601-RXN-FUM/REDUCED-MENAQUINONE//SUC/CPD-9728.38. (reverse)	15	[]	24.37	FUMARATE-REDUCTASE[i]	[’SUC[c]’]	[]	30	False
16829675	R601-RXN-FUM/REDUCED-MENAQUINONE//SUC/CPD-9728.38.	230	[4]	373.6	FUMARATE-REDUCTASE[i]	[’FUM[c]’]	[’FUM[c]’]	30	False
15271350	RFFTRANS-RXN	0.38	[]	0.873	CPLX0-7990[c]	[’DTDP-DEOH-DEOXY-GLUCOSE[c]’]	[]	25	False
Chiu, T.H.; Evans, K.L.; Feingold, D.S. L-Rhamnulose-1-phosphate aldolase Methods Enzymol. 42C 264-269 1975	RHAMNULPALDOL-RXN	38.3	[]	38.3	RHAMNULPALDOL-CPLX[c]	[’RHAMNULOSE-1P[c]’]	[]	37	False

18085797	RHAMNULPALDOL-RXN (reverse)	9.1	[]	9.1	RHAMNULPALDOL-CPLX[c]	['DIHYDROXY-ACETONE-PHOSPHATE[c]']	[]	37	False
12517338	RIB5PISOM-RXN__RIB5PISOMA-CPLX	2100	[]	2100	RIB5PISOMA-CPLX[c]	['RIBOSE-5P[c]']	[]	37	False
15843156	RIBOFLAVIN-SYN-RXN	0.267	[]	0.6134	CPLX0-3952[c]	['DIMETHYL-D-RIBITYL-LUMAZINE[c]']	[]	25	False
19099445	RIBOSYLHOMOCYSTEINASE-RXN	0.38	[]	0.873	EG12712-MONOMER[c]	['CPD-564[c]']	[]	25	False
9548961	RIBULPEPIM-RXN__RIBULPEPIM-CPLX	20.4	[]	20.4	RIBULPEPIM-CPLX[c]	['L-RIBULOSE-5-P[c]']	[]	37	False
17176045	RXN-10856	0.089	[]	0.2045	CPLX0-3936[c]	['CPD-11770[c]']	[]	25	False
10642176	RXN-11484-Pimeloyl-ACPs/L-ALPHA-ALANINE/PROTON//8-AMINO-7-OXONONANOATE/CARBON-DIOXIDE/ACP.80.	0.06	[500]	0.09747	7KAPSYN-CPLX[c]	['Pimeloyl-ACPs[c]', 'L-ALPHA-ALANINE[c]']	['L-ALPHA-ALANINE[c]']	30	False
13785427	RXN-11496-PYRUVATE/UBIQUINONE-8/WATER//CARBON-DIOXIDE/ACET/CPD-9956.58.	167	[]	383.7	PYRUVOXID-CPLX[i]	['PYRUVATE[c]']	[]		False
17542990	RXN-11832	103	[0.035]	167.3	CMPKI-MONOMER[c]	['CMP[c]']	['CMP[c]']	30	False
10529181	RXN-11839	0.12	[]	0.12	EG11177-MONOMER[c]	['tRNA-uridine55[c]']	[]	37	False
19664587	RXN-11841 (reverse)	0.00097	[]	0.00097	G7422-MONOMER[c]	['tRNA-uridine13[c]']	[]	37	False
20452364	RXN-12458	0.09	[]	0.09	CPLX0-3950[c]	['Guanine37-in-tRNA[c]']	[]	37	False
17635929	RXN-12583__E1P-CPLX (reverse)	37.9	[4.62]	61.57	E1P-CPLX[c]	['PYRUVATE[c]', 'THIAMINE-PYROPHOSPHATE[c]']	['THIAMINE-PYROPHOSPHATE[c]']	30	False
11342129	RXN-13990 (reverse)	0.0121	[]	0.0278	KDPGALDOL-4OH2OXOGLUTARALDOL-CPLX[c]	['PYRUVATE[c]']	[]		False
18083802	RXN-14073__GLYCPDIESTER-CYTO-MONOMER	3.3	[]	3.791	GLYCPDIESTER-CYTO-MONOMER[c]	['GLYCEROPHOSPHOGLYCEROL[c]']	[]	35	False
18083802	RXN-14136	2.8	[]	3.216	GLYCPDIESTER-CYTO-MONOMER[c]	['CPD0-2030[c]']	[]	35	False
24210219	RXN-14139__CPLX0-8106	15.2	[69.1]	15.2	CPLX0-8106[c]	['UTP[c]']	['UTP[c]']	37	False
15489502	RXN-14142[CCO-CYTOSOL]-DGMP/WATER//DEOXYGUANOSINE/Pi.43.__EG11817-MONOMER	8.04	[280]	8.04	EG11817-MONOMER[c]	['DGMP[c]']	['DGMP[c]']	37	False
16297670	RXN-14142[CCO-PERI-BAC]-DGMP/WATER//DEOXYGUANOSINE/Pi.44.__APHA-CPLX	15	[3.0]	15	APHA-CPLX[p]	['DGMP[p]']	['DGMP[c]']	37	False
15489502	RXN-14143__CPLX0-7625	0.23	[]	0.23	CPLX0-7625[c]	['DUMP[c]']	[]	37	False
18083802	RXN-14160	1.1	[]	1.264	GLYCPDIESTER-CYTO-MONOMER[c]	['L-1-GLYCEROPHOSPHORYLETHANOLAMINE[c]']	[]	35	False

15489502	RXN-14161[CCO-CYTOSOL]-DAMP/WATER//DEOXYADENOSINE/Pi.43.__CPLX0-7625	0.3	[12]	0.3	CPLX0-7625[c]	[’DAMP[c]’]	[’DAMP[c]’]	37	False
12383057	RXN-14325	9.4	[]	9.4	CTPSYN-CPLX[c]	[’AMMONIUM[c]’]	[]	37	False
15498577	RXN-15125__CPLX0-7622	544	[4800]	544	CPLX0-7622[c]	[’SER[c]’]	[’SER[c]’]	37	False
19370061	RXN-15131__CYSTATHIONINE-BETA-LYASE-CPLX	35	[]	80.41	CYSTATHIONINE-BETA-LYASE-CPLX[c]	[’L-CYSTATHIONINE[c]’]	[]	25	False
12711733	RXN-15313 (reverse)	7.7	[]	17.69	ACNEULY-CPLX[c]	[’CPD0-1123[c]’]	[]	25	False
21081107	RXN-15578	6.8	[227]	15.62	TRYPTOPHAN-CPLX[c]	[’TRP[c]’]	[’TRP[c]’]	25	False
16221850	RXN-15581	15.7	[]	36.07	DSERDEAM-MONOMER[c]	[’D-SERINE[c]’]	[]		False
24509771	RXN-15607	0.0517	[]	0.0517	KDUD-MONOMER[c]	[’11-DEOXYCORTICOSTERONE[c]’]	[]	37	False
24509771	RXN-15607	0.283	[37]	0.283	KDUD-MONOMER[c]	[’NADH[c]’]	[’NADH[c]’]	37	False
6792201	RXN-15740-GLYCEROL-3P/CPD-9728//DIHYDROXY-ACETONE-PHOSPHATE/REDUCED-MENAQUINONE.70.	33.3	[339]	76.5	ANGLYC3PDEHYDROG-CPLX[i]	[’GLYCEROL-3P[c]’]	[’GLYCEROL-3P[c]’]	25	False
25225400	RXN-15943	127.6	[]	293.1	CPLX0-7820[c]	[’CO-A[c]’]	[]		False
12535615	RXN-16000 (reverse)	5.76	[]	9.357	CPLX0-1962[c]	[’L-ALLO-THREONINE[c]’]	[]	30	False
Kiick, K.L.; Tirrell, D.A. Protein Engineering by In Vivo Incorporation of Non-Natural Amino Acids: Control of Incorporation of Methionine Analogues by Methionyl-tRNA Synthetase Tetrahedron 56 9487-9493 2000	RXN-16165-Initiation-tRNAmet/MET/ATP/PROTON//L-methionyl-tRNAfmet/AMP/PPI.64. (reverse)	13.3	[24.3]	30.56	METG-CPLX[c]	[’MET[c]’]	[’MET[c]’]	25	False
9607323	RXN-16165-Initiation-tRNAmet/MET/ATP/PROTON//L-methionyl-tRNAfmet/AMP/PPI.64. (reverse)	6.85	[]	6.85	METG-CPLX[c]	[’Initiation-tRNAmet[c]’]	[]	37	False
16039592	RXN-1961	0.0153	[220]	0.0153	G6096-MONOMER[c]	[’ATP[c]’]	[’ATP[c]’]	37	False
16039592	RXN-1961	0.0702	[380]	0.0702	G6096-MONOMER[c]	[’LYS[c]’]	[’LYS[c]’]	37	False
11515538	RXN-6622__CPLX0-8122	178.33333	[]	178.3	CPLX0-8122[c]	[’CYS-GLY[c]’]	[]	37	False

16990279	RXN-7607[CCO-CYTOSOL]-IMP/WATER//INOSINE/Pi.35.__G7742-MONOMER	2.6	[]	5.973	G7742-MONOMER[c]	[’IMP[c]’]	[]		False
16990279	RXN-7609[CCO-CYTOSOL]-GMP/WATER//GUANOSINE/Pi.37.__G7742-MONOMER	5.5	[1900]	12.64	G7742-MONOMER[c]	[’GMP[c]’]	[’GMP[c]’]		False
16297670	RXN-7609[CCO-PERI-BAC]-GMP/WATER//GUANOSINE/Pi.38.__APHA-CPLX	65	[]	65	APHA-CPLX[p]	[’GMP[p]’]	[]	37	False
16990279	RXN-7609[CCO-PERI-BAC]-GMP/WATER//GUANOSINE/Pi.38.__G7742-MONOMER	5.5	[]	12.64	G7742-MONOMER[c]	[’GMP[p]’]	[]		False
17542990	RXN-7913	108	[]	175.4	CMPKI-MONOMER[c]	[’DCMP[c]’]	[]	30	False
12637497	RXN-7967-QUINATE/NAD//DEHYDROQUINATE/NADH/PROTON.40. (reverse)	0.05	[116]	0.1625	CPLX0-7462[c]	[’QUINATE[c]’, ’NAD[c]’]	[’NAD[c]’]	20	False
12637497	RXN-7967-QUINATE/NADP//DEHYDROQUINATE/NADPH/PROTON.42. (reverse)	0.05	[500]	0.1625	CPLX0-7462[c]	[’QUINATE[c]’, ’NADP[c]’]	[’NADP[c]’]	20	False
23018273	RXN-8667	83	[]	219	HYDROPEROXIDI-CPLX[c]	[’HYDROGEN-PEROXIDE[c]’]	[]	23	False
22471615	RXN-8992	2	[]	4.595	CPLX0-7426[c]	[’FARNESYL-PP[c]’, ’DELTA3-ISOPENTENYL-PP[c]’]	[]	25	False
21704016	RXN-8999	2.5	[]	5.743	UPPSYN-CPLX[c]	[’FARNESYL-PP[c]’, ’DELTA3-ISOPENTENYL-PP[c]’]	[]	25	False
18284213	RXN-9310	167	[]	167	EG12438-MONOMER[c]	[’CPD-9924[c]’]	[]	37	False
23564174	RXN-9311	6.2	[]	14.24	CPLX0-8128[c]	[’CPD-9925[c]’]	[]	25	False
22017312	RXN-9535-Dodecanoyl-ACPs/MALONYL-ACP/PROTON//3-oxo-myristoyl-ACPs/CARBON-DIOXIDE/ACP.76.	0.0567	[]	0.1303	FABB-CPLX[c]	[’Dodecanoyl-ACPs[c]’]	[]	25	False
22017312	RXN-9535-Dodecanoyl-ACPs/MALONYL-ACP/PROTON//3-oxo-myristoyl-ACPs/CARBON-DIOXIDE/ACP.76.	0.11	[]	0.2527	FABB-CPLX[c]	[’MALONYL-ACP[c]’]	[]	25	False
8706750	RXN-9772	5.55	[2700, 2500]	12.75	L-ASPARTATE-OXID-MONOMER[c]	[’L-ASPARTATE[c]’, ’FUM[c]’]	[’L-ASPARTATE[c]’, ’FUM[c]’]	25	False
10542272	RXN0-1441	117.3	[]	117.3	CPLX0-3721[c]	[’ADENOSINE_DIPHOSPHATE_RIBOSE[c]’]	[]	37	False
12975365	RXN0-1461	0.003	[]	0.003	CPLX0-7808[c]	[’COPROPORPHYRINOGEN_III[c]’]	[]	37	False
11934293	RXN0-1941 (reverse)	2.6	[]	4.224	MONOMER0-148[c]	[’ETHYL-2-METHYLACETOACETATE[c]’]	[]	30	False
12535615	RXN0-2201 (reverse)	4.2	[40000, 540]	6.823	CPLX0-1962[c]	[’SER[c]’, ’NADP[c]’]	[’SER[c]’, ’NADP[c]’]	30	False
10769117	RXN0-268 (reverse)	0.72	[7.1]	1.654	G7517-MONOMER[c]	[’PROPIONYL-COA[c]’]	[’PROPIONYL-COA[c]’]	25	False
11237876	RXN0-300__CPLX0-235	72.5	[]	166.6	CPLX0-235[i]	[’OH-PYR[c]’]	[]	25	False
11237876	RXN0-300__G6539-MONOMER	11.8	[]	27.11	G6539-MONOMER[c]	[’OH-PYR[c]’]	[]	25	False

20480490	RXN0-302	2.7	[]	6.203	CPLX0-721[c]	[’2-PHOSPHO-4-CYTIDINE-5-DIPHOSPHO-2-C-MET[c]’]	[]		False
10769117	RXN0-310	1.6	[]	1.6	CPLX0-254[c]	[’METHYL-MALONYL-COA[c]’]	[]	37	False
1906883	RXN0-3281	2833	[]	6976	FORMATEDEHYDROGH-MONOMER[c]	[’FORMATE[c]’]	[]	24	False
18353368	RXN0-3741[CCO-CYTOSOL]-DAMP/WATER//DEOXYADENOSINE/Pi.43.__G7230-MONOMER	2.28	[118.9]	2.28	G7230-MONOMER[c]	[’DAMP[c]’]	[’DAMP[c]’]	37	False
18353368	RXN0-3741[CCO-CYTOSOL]-DCMP/WATER//DEOXYCYTIDINE/Pi.42.__G7230-MONOMER	0.84	[]	0.84	G7230-MONOMER[c]	[’DCMP[c]’]	[]	37	False
15489502	RXN0-3741[CCO-CYTOSOL]-DIMP/WATER//DEOXYINOSINE/Pi.41.__CPLX0-7625	0.24	[]	0.24	CPLX0-7625[c]	[’DIMP[c]’]	[]	37	False
18353368	RXN0-3741[CCO-CYTOSOL]-DIMP/WATER//DEOXYINOSINE/Pi.41.__G7230-MONOMER	0.61	[]	0.61	G7230-MONOMER[c]	[’DIMP[c]’]	[]	37	False
18353368	RXN0-3741[CCO-CYTOSOL]-DUMP/WATER//DEOXYURIDINE/Pi.41.__G7230-MONOMER	1.36	[]	1.36	G7230-MONOMER[c]	[’DUMP[c]’]	[]	37	False
15489502	RXN0-3741[CCO-CYTOSOL]-TMP/WATER//THYMIDINE/Pi.37.__CPLX0-7625	0.16	[]	0.16	CPLX0-7625[c]	[’TMP[c]’]	[]	37	False
18353368	RXN0-3741[CCO-CYTOSOL]-TMP/WATER//THYMIDINE/Pi.37.__G7230-MONOMER	1.19	[]	1.19	G7230-MONOMER[c]	[’TMP[c]’]	[]	37	False
17698004	RXN0-384__H2NEOPTERINP3PYROPHOSPHOHYDRO-MONOMER	4.59	[790]	4.59	H2NEOPTERINP3PYROPHOSPHOHYDRO-MONOMER[c]	[’DATP[c]’]	[’DATP[c]’]	37	False
16077126	RXN0-4281__G6967-MONOMER	2.85	[]	6.548	G6967-MONOMER[c]	[’METHYL-GLYOXAL[c]’]	[]	25	False
16077126	RXN0-4281__G7558-MONOMER	11	[]	25.27	G7558-MONOMER[c]	[’METHYL-GLYOXAL[c]’]	[]	25	False
16077126	RXN0-4281__MONOMER0-148	27.62	[]	63.45	MONOMER0-148[c]	[’METHYL-GLYOXAL[c]’]	[]	25	False
16077126	RXN0-4281__MONOMER0-149	29.15	[]	66.97	MONOMER0-149[c]	[’METHYL-GLYOXAL[c]’]	[]	25	False
18056714	RXN0-4301	0.44	[]	1.011	CPLX0-7660[c]	[’D-SEDOHEPTULOSE-7-P[c]’]	[]	25	False
20050615	RXN0-4361	35.7	[]	82.02	EG11736-MONOMER[c]	[’D-BETA-D-HEPTOSE-17-DIPHOSPHATE[c]’]	[]	25	False
18837509	RXN0-4641-CPD0-881/WATER//CPD-16168/D-LACTATE.36.	5.7	[]	9.26	CPLX0-7732[c]	[’CPD0-881[c]’]	[]	30	False
16464851	RXN0-5065 (reverse)	100	[]	229.7	CPLX0-3941[c]	[’34-DIHYDROXYPHENYLACETYL-COA[c]’]	[]	25	False
24210219	RXN0-5107__CPLX0-8106	15.4	[]	15.4	CPLX0-8106[c]	[’TTP[c]’]	[]	37	False

16766526	RXN0-5108	75	[]	75	CPLX0-3971[c]	[’GDP-MANNOSE[c]’]	[]	37	False
16990279	RXN0-5114[CCO-CYTOSOL]-3-P-SERINE/WATER//SER/Pi.38.	66	[]	151.6	PSERPHOSPHA-MONOMER[c]	[’3-P-SERINE[c]’]	[]		False
11747300	RXN0-5116	80	[20]	226.3	RIBULOKIN-CPLX[c]	[’L-RIBULOSE[c]’, ’ATP[c]’]	[’ATP[c]’]	22	False
22059588	RXN0-5118	4.7	[]	4.7	EG11189-MONOMER[c]	[’KDO2-LIPID-A[c]’]	[]	37	False
24479701	RXN0-5120	0.5	[162]	1.149	EG11339-MONOMER[c]	[’CPD-12575[c]’, ’CPD0-930[c]’]	[’CPD-12575[c]’]	25	False
16813561	RXN0-5141 (reverse)	50.8	[100]	116.7	MONOMER0-149[c]	[’NADPH[c]’]	[’NADPH[c]’]	25	False
16990279	RXN0-5187[CCO-CYTOSOL]-FMN/WATER//RIBOFLAVIN/Pi.38.__EG11202-MONOMER	9.4	[]	21.6	EG11202-MONOMER[c]	[’FMN[c]’]	[]		False
24123841	RXN0-5187[CCO-CYTOSOL]-FMN/WATER//RIBOFLAVIN/Pi.38.__G6442-MONOMER	0.6	[]	0.6	G6442-MONOMER[c]	[’FMN[c]’]	[]	37	False
16813561	RXN0-5213 (reverse)	5.6	[52]	12.87	G6958-MONOMER[c]	[’NADH[c]’]	[’NADH[c]’]	25	False
11747447	RXN0-5228 (reverse)	10	[]	16.25	G6661-MONOMER[c]	[’CPD0-2190[c]’]	[]	30	False
19883126	RXN0-5234 (reverse)	0.5	[]	0.8123	GLYOHMETRANS-CPLX[c]	[’ALLO-THR[c]’]	[]	30	False
21059411	RXN0-5240 (reverse)	0.000633	[]	0.000633	GLYOHMETRANS-CPLX[c]	[’D-ALANINE[c]’]	[]	37	False
15489502	RXN0-5292[CCO-CYTOSOL]-DCMP/WATER//DEOXYCYTIDINE/Pi.42.__CPLX0-7625	0.23	[]	0.23	CPLX0-7625[c]	[’DCMP[c]’]	[]	37	False
18390652	RXN0-5364	0.08	[]	0.1838	EG12387-MONOMER[c]	[’CPD-381[c]’]	[]		False
16672604	RXN0-5387	370	[]	370	G7868-MONOMER[c]	[’CPD0-1156[c]’]	[]	37	False
16672604	RXN0-5387	6	[170]	6	G7868-MONOMER[c]	[’NADPH[c]’]	[’NADPH[c]’]	37	False
18576672	RXN0-5390	23.7	[]	54.45	G6244-MONOMER[c]	[’CPD0-1158[c]’]	[]		False
18754693	RXN0-5433	0.4	[]	0.919	CPLX0-7723[c]	[’DEHYDRO-3-DEOXY-L-RHAMNONATE[c]’]	[]	25	False
10828971	RXN0-6274	1.26	[]	1.26	EG10595-MONOMER[c]	[’CPD-4211[c]’]	[]	37	False
21119630	RXN0-6562	5.7	[]	5.7	EG12693-MONOMER[c]	[’3-P-HYDROXYPYRUVATE[c]’]	[]	37	False
21119630	RXN0-6564	9.7	[]	22.28	HOMOSERKIN-CPLX[c]	[’CPD0-2189[c]’]	[]	25	False
21169495	RXN0-6565	0.28	[]	0.4549	CPLX0-7788[c]	[’DIHYDRO-THYMINE[c]’]	[]	30	False
21169495	RXN0-6565 (reverse)	0.39	[]	0.6336	CPLX0-7788[c]	[’THYMINE[c]’]	[]	30	False
19140736	RXN0-7008-PRO/UBIQUINONE-8//L-DELTA1-PYRROLINE_5-CARBOXYLATE/CPD-9956/PROTON.67.__PUTA-CPLXBND	29	[59000]	66.62	PUTA-CPLXBND[c]	[’PRO[c]’]	[’PRO[c]’]	25	False
19140736	RXN0-7008-PRO/UBIQUINONE-8//L-DELTA1-PYRROLINE_5-CARBOXYLATE/CPD-9956/PROTON.67.__PUTA-CPLX	29	[59000]	66.62	PUTA-CPLX[c]	[’PRO[c]’]	[’PRO[c]’]	25	False
11741871	RXN0-703	110	[]	252.7	CPLX0-2061[c]	[’CPD-334[c]’]	[]	25	False

11741871	RXN0-704	48	[]	110.3	LYXK-CPLX[c]	[’3-KETO-L-GULONATE[c]’]	[]	25	False
11741871	RXN0-705__CPLX0-7744	51	[]	117.2	CPLX0-7744[c]	[’CPD-2343[c]’]	[]	25	False
11741871	RXN0-705__EG12285-MONOMER	64	[]	147	EG12285-MONOMER[c]	[’CPD-2343[c]’]	[]	25	False
23935849	RXN0-7075	11	[17]	25.27	CPLX0-8032[c]	[’ACETYL-COA[c]’]	[’ACETYL-COA[c]’]	25	False
23935849	RXN0-7075	15	[]	34.46	CPLX0-8032[c]	[’OXALATE[c]’]	[]	25	False
24509771	RXN0-7101__KDUD-MONOMER	0.21	[285]	0.21	KDUD-MONOMER[c]	[’NAD[c]’]	[’NAD[c]’]	37	False
24509771	RXN0-7101__KDUD-MONOMER (reverse)	0.5116667	[]	0.5117	KDUD-MONOMER[c]	[’5-DEHYDROGLUCONATE[c]’]	[]	37	False
24509771	RXN0-7101__KDUD-MONOMER	0.9716667	[544800]	0.9717	KDUD-MONOMER[c]	[’GLUCONATE[c]’]	[’GLUCONATE[c]’]	37	False
16464851	RXN0-7104	0.41	[]	0.9419	CPLX0-3941[c]	[’CPD-207[c]’]	[]	25	False
19384989	RXN0-742	52	[90]	119.5	PURK-CPLX[c]	[’ATP[c]’, ’5-PHOSPHORIBOSYL-5-AMINOIMIDAZOLE[c]’]	[’ATP[c]’]	25	False
22970852	RXN0-961	38.3	[]	38.3	CPLX0-7989[c]	[’L-ALA-GAMMA-D-GLU-DAP[c]’]	[]	37	False
21296885	RXNMETA-12671	39	[]	110.3	G6708-MONOMER[c]	[’CPD0-2363[c]’]	[]	22	False
12135357	S-ADENMETSYN-RXN	1.5	[92, 73]	4.243	S-ADENMETSYN-CPLX[c]	[’MET[c]’, ’ATP[c]’]	[’MET[c]’, ’ATP[c]’]	22	False
16567800	S-FORMYLGLUTATHIONE-HYDROLASE-RXN__CPLX0-3954	6.51	[]	6.51	CPLX0-3954[c]	[’CPD-548[c]’]	[]	37	False
16567800	S-FORMYLGLUTATHIONE-HYDROLASE-RXN__G6208-MONOMER	28.5	[]	28.5	G6208-MONOMER[c]	[’CPD-548[c]’]	[]	37	False
8065908	SERINE–TRNA-LIGASE-RXN-SER-tRNAs/SER/ATP/PROTON//Charged-SER-tRNAs/AMP/PPI.52.	0.34	[64]	0.7811	SERS-CPLX[c]	[’SER[c]’]	[’SER[c]’]	25	True
8065908	SERINE–TRNA-LIGASE-RXN-SER-tRNAs/SER/ATP/PROTON//Charged-SER-tRNAs/AMP/PPI.52.	0.35	[0.068]	0.8041	SERS-CPLX[c]	[’ATP[c]’]	[’ATP[c]’]	25	True
8065908	SERINE–TRNA-LIGASE-RXN-SER-tRNAs/SER/ATP/PROTON//Charged-SER-tRNAs/AMP/PPI.52.	2.6	[]	5.973	SERS-CPLX[c]	[’SER-tRNAs[c]’]	[]	25	True
12627979	SERINE-O-ACETTRAN-RXN__CPLX0-237	427	[200, 1170]	981	CPLX0-237[c]	[’ACETYL-COA[c]’, ’SER[c]’]	[’ACETYL-COA[c]’, ’SER[c]’]	25	False

Mino, K et al Effects of Bienzyme Complex Formation of Cysteine Synthetase from Escherichia coli on Some Properties and Kinetics. 2014. Bioscience, Biotechnology, and Biochemistry	SERINE-O-ACETTRAN-RXN__CYSSYNMULTI-CPLX	427	[200, 1170]	981	CYSSYNMULTI-CPLX[c]	['ACETYL-COA[c]', 'SER[c]']	['ACETYL-COA[c]', 'SER[c]']	25	False
12637497	SHIKIMATE-5-DEHYDROGENASE-RXN__AROE-MONOMER	236.67	[65, 56]	768.9	AROE-MONOMER[c]	['SHIKIMATE[c]', 'NADP[c]']	['SHIKIMATE[c]', 'NADP[c]']	20	False
23001854	SPERMIDINESYN-RXN	0.14	[]	0.14	SPERMIDINESYN-CPLX[c]	['S-ADENOSYLMETHIONINAMINE[c]']	[]	37	False
10353839	SUCCCOASYN-RXN (reverse)	44.7	[250, 70, 4]	102.7	SUCCCOASYN[c]	['SUC[c]', 'ATP[c]', 'CO-A[c]']	['SUC[c]', 'ATP[c]', 'CO-A[c]']	25	False
11803023	SUCCINATE-DEHYDROGENASE-UBIQUINONE-RXN-SUC/UBIQUINONE-8//FUM/CPD-9956.31.	85	[2]	138.1	CPLX0-8160[i]	['SUC[c]']	['SUC[c]']	30	True
10074354	SUCCINYLDIAMINOPIMTRANS-RXN__ACETYLORNTRANSAM-CPLX (reverse)	0.3	[]	0.6892	ACETYLORNTRANSAM-CPLX[c]	['N-SUCCINYLLL-2-6-DIAMINOPIMELATE[c]']	[]	25	False
23484010	SUCCORNTRANSAM-RXN	3	[]	6.892	SUCCORNTRANSAM-CPLX[c]	['N2-SUCCINYLORNITHINE[c]']	[]		False
4150390	SULFITE-REDUCT-RXN	30	[4.5]	79.17	SULFITE-REDUCT-CPLX[c]	['SO3[c]', 'NADPH[c]']	['NADPH[c]']	23	False
25160617	TARTRATE-DEHYDROGENASE-RXN	0.26	[600]	0.5973	G6986-MONOMER[c]	['TARTRATE[c]', 'NAD[c]']	['NAD[c]']	25	False
2187529	THIOREDOXIN-REDUCT-NADPH-RXN-Red-Thioredoxin/NADP//Ox-Thioredoxin/NADPH/PROTON.50.	22.8	[]	22.8	THIOREDOXIN-REDUCT-NADPH-CPLX[c]	['Ox-Thioredoxin[c]']	[]	37	False
22411989	THREODEHYD-RXN__THREODEHYD-CPLX	88	[6500, 270]	202.2	THREODEHYD-CPLX[c]	['THR[c]', 'NAD[c]']	['THR[c]', 'NAD[c]']	25	False
14690420	THREONINE–TRNA-LIGASE-RXN-THR-tRNAs/THR/ATP/PROTON//Charged-THR-tRNAs/AMP/PPI.52.	90	[0.201, 0.267]	90	THRS-CPLX[c]	['THR[c]', 'ATP[c]']	['THR[c]', 'ATP[c]']	37	True
21059411	THREONINE-ALDOLASE-RXN	1.033	[10000]	1.678	LTAA-CPLX[c]	['THR[c]']	['THR[c]']	30	False

7907888	THRESYN-RXN	7.33	[]	7.33	THRESYN-MONOMER[c]	['O-PHOSPHO-L-HOMOSERINE[c]']	[]	37	False
15658853	THYM-PHOSPH-RXN	1770	[]	4066	DEOA-CPLX[c]	['THYMIDINE[c]']	[]	25	False
10831428	TREHALOSE6PSYN-RXN	7.3	[8600]	11.86	TREHALOSE6PSYN-MONOMER[c]	['CPD-12575[c]']	['CPD-12575[c]']	30	False
10831428	TREHALOSEPHOSPHA-RXN	14.6	[]	23.72	TREHALOSEPHOSPHASYN-MONOMER[c]	['TREHALOSE-6P[c]']	[]	30	False
9442062	TRIOSEPISOMERIZATION-RXN (reverse)	9000	[]	2.068e+04	TPI[c]	['GAP[c]']	[]	25	False
22984449	TRIPHOSPHATASE-RXN[CCO-CYTOSOL]-P3I/WATER//Pi/PPI.31.__CPLX0-243	16.7	[]	16.7	CPLX0-243[c]	['P3I[c]']	[]	37	False
8555191	TRYPTOPHAN–TRNA-LIGASE-RXN-TRP/TRP-tRNAs/ATP/PROTON//Charged-TRP-tRNAs/AMP/PPI.52.	1.5	[190]	1.5	TRPS-CPLX[c]	['ATP[c]', 'TRP[c]', 'TRP-tRNAs[c]']	['ATP[c]']	37	True
8555191	TRYPTOPHAN–TRNA-LIGASE-RXN-TRP/TRP-tRNAs/ATP/PROTON//Charged-TRP-tRNAs/AMP/PPI.52.	2	[12.4]	2	TRPS-CPLX[c]	['ATP[c]', 'TRP[c]', 'TRP-tRNAs[c]']	['TRP[c]']	37	True
9772162	TSA-REDUCT-RXN-GLYCERATE/NAD//TARTRONATE-S-ALD/NADH/PROTON.44.	23	[]	65.05	TSA-REDUCT-MONOMER[c]	['TARTRONATE-S-ALD[c]']	[]	22	False
12097643	TYROSINE–TRNA-LIGASE-RXN-TYR-tRNAs/TYR/ATP/PROTON//Charged-TYR-tRNAs/AMP/PPI.52.	12	[5.3]	12	TYRS-CPLX[c]	['TYR[c]']	['TYR[c]']	37	True
8218300	TYROSINE-AMINOTRANSFERASE-RXN__TYRB-DIMER	210	[320, 1300]	482.5	TYRB-DIMER[c]	['TYR[c]', '2-KETOGLUTARATE[c]']	['TYR[c]', '2-KETOGLUTARATE[c]']	25	False
8976565	UDP-NACMUR-ALA-LIG-RXN	16.3	[48, 130]	16.3	CPLX0-8014[c]	['L-ALPHA-ALANINE[c]', 'UDP-N-ACETYLMURAMATE[c]', 'ATP[c]']	['L-ALPHA-ALANINE[c]', 'ATP[c]']	37	False
20136146	UDPACYLGLCNACDEACETYL-RXN	8.83	[]	14.34	UDPACYLGLCNACDEACETYL-MONOMER[c]	['UDP-OHMYR-ACETYLGLUCOSAMINE[c]']	[]	30	False

Chen, X.; Kowal, P.; Hamad, S.; Fan, H.; Wang, P.G. Cloning, expression and characterization of a UDP-galactose 4-epimerase from Escherichia coli Biotechnol. Lett. 21 1131-1135 1999	UDPGLUCEPIM-RXN	18	[1200]	44.32	UDPGLUCEPIM-CPLX[c]	['CPD-12575[c]']	['CPD-12575[c]']	24	False
20717852	UDPGLUCEPIM-RXN (reverse)	750	[]	861.5	UDPGLUCEPIM-CPLX[c]	['CPD-14553[c]']	[]	35	False
10480918	UDPNACETYLGLUCOSAMACYLTRANS-RXN-R-3-hydroxymyristoyl-ACPs/UDP-N-ACETYL-D-GLUCOSAMINE//UDP-OHMYR-ACETYLGLUCOSAMINE/ACP.86.	7.17	[820]	11.65	UDPNACETYLGLUCOSAMACYLTRANS-CPLX[c]	['UDP-N-ACETYL-D-GLUCOSAMINE[c]']	['UDP-N-ACETYL-D-GLUCOSAMINE[c]']	30	False
Dube, S.; Nanda, K.; Rani, R.; Kaur, N.; Nagpal, J.; Upadhyay, D.; Cliffe, I.; Saini, K.; Purnapatre, K. UDP-N-acetylglucosamine enolpyruvyl transferase from Pseudomonas aeruginosa World J. Microbiol. Biotechnol. 26 1623-1629 2010	UDPNACETYLGLUCOSAMENOLPYRTRANS-RXN	2.4	[36, 0.84]	2.4	UDPNACETYLGLUCOSAMENOLPYRTRANS-MONOMER[c]	['UDP-N-ACETYL-D-GLUCOSAMINE[c]', 'PHOSPHO-ENOL-PYRUVATE[c]']	['UDP-N-ACETYL-D-GLUCOSAMINE[c]', 'PHOSPHO-ENOL-PYRUVATE[c]']	37	False
16436710	UDPNACETYLMURAMATEDEHYDROG-RXN	15	[11]	34.46	UDPNACETYLMURAMATEDEHYDROG-MONOMER[c]	['NADPH[c]']	['NADPH[c]']		False
8689233	UGD-RXN	11.3	[]	25.96	CPLX0-8098[c]	['CPD-12575[c]']	[]	25	False
18411271	UNDECAPRENYL-DIPHOSPHATASE-RXN__PGPPHOSPHAB-MONOMER	50	[360]	50	PGPPHOSPHAB-MONOMER[i]	['UNDECAPRENYL-DIPHOSPHATE[c]']	['UNDECAPRENYL-DIPHOSPHATE[c]']	37	False
17309433	URA-PHOSPH-RXN	35	[]	80.41	DEOA-CPLX[c]	['DEOXYURIDINE[c]']	[]	25	False

2557837	UROGENIIISYN-RXN	500	[]	1149	UROGENIIISYN-MONOMER[c]	[’HYDROXYMETHYLBILANE[c]’]	[]		False
Panova, N.G.; Shcheveleva, E.V.; Alexeev, C.S.; Mukhortov, V.G.; Zuev, A.N.; Mikhailov, S.N.; Esipov, R.S.; Chuvikovsky, D.V.; Miroshnikov, A.I. Use of 4-thiouridine and 4-thiothymidine in studies on pyrimidine nucleoside phosphorylases Mol. Biol. 38 770-776 2004	URPHOS-RXN (reverse)	98	[80]	225.1	URPHOS-CPLX[c]	[’URIDINE[c]’]	[’URIDINE[c]’]	25	False
20498377	VALINE–TRNA-LIGASE-RXN-VAL-tRNAs/VAL/ATP/PROTON//Charged-VAL-tRNAs/AMP/PPI.52.	12.9	[9400]	12.9	VALS-MONOMER[c]	[’ATP[c]’]	[’ATP[c]’]	37	True
14970394	VALINE–TRNA-LIGASE-RXN-VAL-tRNAs/VAL/ATP/PROTON//Charged-VAL-tRNAs/AMP/PPI.52.	13.9	[4.3]	13.9	VALS-MONOMER[c]	[’VAL[c]’]	[’VAL[c]’]	37	True
16843487	VALINE–TRNA-LIGASE-RXN-VAL-tRNAs/VAL/ATP/PROTON//Charged-VAL-tRNAs/AMP/PPI.52.	4	[]	9.19	VALS-MONOMER[c]	[’VAL-tRNAs[c]’]	[]		True
9100006	XANPRIBOSYLTRAN-RXN	150	[]	344.6	GPT-CPLX[i]	[’XANTHINE[c]’]	[]	25	False
17123542	XYLULOKIN-RXN	255	[150]	585.8	CPLX0-7466[c]	[’D-XYLULOSE[c]’, ’ATP[c]’]	[’ATP[c]’]	25	False

Table S9. Data sources used in the model and presented in Movie S1. Citation or PMID show reference to original data source. Data used represents the number of data points included in the model, represented by the size of dots in the movie. Process is the model related process that uses the data. Equation is the grouping the animated dot moves to in the movie. The table is also included in the GitHub repository: wcEcoli/paper/movie_data_sources.tsv.

Citation	PMID	Year Published	Data Used	Process	Equation
	7918446	1994	6	translation	Protein
	9748544	1998	4	translation	Protein
	2009959	1991	4	translation	Protein
	9171418	1997	4	translation	Protein
	12662918	2003	6	translation	Protein
	368055	1979	4	translation	Protein
	6989402	1980	4	translation	Protein
	4594761	1974	4	translation	Protein
	15751955	2005	6	translation	Protein
	8672449	1996	4	translation	Protein
	12739900	2003	6	translation	Protein
	15362869	2004	4	translation	Protein
	9607323	1998	2	translation	Protein
	21175197	2011	6	translation	Protein
	16866361	2006	4	translation	Protein
	11408489	2001	2	translation	Protein
	8065908	1994	6	translation	Protein
	14690420	2003	4	translation	Protein
	8555191	1996	5	translation	Protein
	12097643	2002	2	translation	Protein
	12475234	2002	6	translation	Protein
https://doi.org/10.1006/jmbi.1996.0428		1996	220	transcription	RNA
GSE11183		2009	4353	transcription regulation	RNA
GSE1121		2004	4353	transcription regulation	RNA
GSE12006		2008	4353	transcription regulation	RNA
GSE15050		2009	4353	transcription regulation	RNA
GSE17519		2009	4353	transcription regulation	RNA
GSE17584		2009	4353	transcription regulation	RNA
GSE21857		2012	4353	transcription regulation	RNA
GSE24524		2012	4353	transcription regulation	RNA
GSE25318		2011	4353	transcription regulation	RNA
GSE26591		2012	4353	transcription regulation	RNA
GSE29076		2005	4353	transcription regulation	RNA
GSE31029		2011	4353	transcription regulation	RNA
GSE33147		2011	4353	transcription regulation	RNA
GSE34023		2012	4353	transcription regulation	RNA
GSE34275		2011	4353	transcription regulation	RNA
GSE40313		2012	4353	transcription regulation	RNA
GSE4724		2006	4353	transcription regulation	RNA
GSE50529		2014	4353	transcription regulation	RNA
GSE54901		2014	4353	transcription regulation	RNA
GSE55365		2014	4353	transcription regulation	RNA
GSE55662		2014	4353	transcription regulation	RNA
GSE65712		2015	4353	transcription regulation	RNA
GSE66482		2015	4353	transcription regulation	RNA

Table S9 continued.

Citation	PMID	Year Published	Data Used	Process	Equation
GSE6836		2007	4353	transcription regulation	RNA
GSE6992		2007	4353	transcription regulation	RNA
GSE72525		2015	4353	transcription regulation	RNA
GSE7398		2007	4353	transcription regulation	RNA
GSE9755		2007	4353	transcription regulation	RNA
PMC219383		2003	4353	transcription regulation	RNA
PMC400601		2004	4353	transcription regulation	RNA
PMC4299492		2014	4353	transcription regulation	RNA
PMC545700		2005	4353	transcription regulation	RNA
	17803904	2007	4353	transcription regulation	RNA
PMC2797255		2009	1	rna degradation	RNA
PMC438955		2004	2	rna degradation	RNA
PMID: 7535857		1995	1	rna degradation	RNA
PMID: 7544280		1995	1	rna degradation	RNA
PMC2223728		2008	1	rna degradation	RNA
PMC1935014		2007	1	rna degradation	RNA
PMC140075		2002	9	rna degradation	RNA
PMC1360286		2006	8	rna degradation	RNA
https://www.researchgate.net/publication/237130769		1996	2	dna replication	RNA
https://www.ncbi.nlm.nih.gov/nuccore/U00096.3		2013	1	dna replication	RNA
10.1093/nar/gkv1156		2016	902	transcription regulation	RNA
https://www.researchgate.net/publication/237130769		1996	10	transcription	RNA
This study		2017	4353	transcription	RNA
doi: 10.1073/pnas.112318199		2002	3835	rna degradation	RNA
DOI: 10.1126/science.3018930		1986	20	protein degradation	Protein
https://www.researchgate.net/publication/237130769		1996	6	translation	Protein
10.1016/j.cell.2014.02.033		2014	4032	translation	Protein
doi:10.1186/1752-0509-8-79		2014	2638	complexation	Protein
doi:10.1186/1752-0509-8-79		2014	10321	metabolism	Metabolite
doi:10.1038/nchembio.186		2009	91	metabolism	Metabolite
	9387	1976	2	metabolism	Metabolite
	346589	1978	1	metabolism	Metabolite
	357906	1978	2	metabolism	Metabolite
	374409	1979	1	metabolism	Metabolite
	1092682	1975	2	metabolism	Metabolite
	1099093	1975	1	metabolism	Metabolite
	1329945	1992	2	metabolism	Metabolite
	1544480	1992	4	metabolism	Metabolite
	1644758	1992	2	metabolism	Metabolite
	1906883	1991	1	metabolism	Metabolite
	1924337	1991	1	metabolism	Metabolite
	1993184	1991	2	metabolism	Metabolite
	2007545	1991	1	metabolism	Metabolite
	2111176	1990	4	metabolism	Metabolite
	2126467	1991	1	metabolism	Metabolite
	2139795	1990	2	metabolism	Metabolite
	2144454	1990	1	metabolism	Metabolite
	2187529	1990	1	metabolism	Metabolite
	2494074	1989	1	metabolism	Metabolite
	2514789	1990	1	metabolism	Metabolite
	2531000	1990	4	metabolism	Metabolite
	2549047	1989	2	metabolism	Metabolite
	2557837	1990	1	metabolism	Metabolite
	2651124	1989	2	metabolism	Metabolite
	2659070	1989	3	metabolism	Metabolite

Table S9 continued.

Citation	PMID	Year Published	Data Used	Process	Equation
	2689171	1990	1	metabolism	Metabolite
	2826481	1988	2	metabolism	Metabolite
	2986688	1985	1	metabolism	Metabolite
	3003503	1986	3	metabolism	Metabolite
	4150390	1974	2	metabolism	Metabolite
	6142052	1984	4	metabolism	Metabolite
	6300054	1983	6	metabolism	Metabolite
	6378903	1984	1	metabolism	Metabolite
	6395895	1985	4	metabolism	Metabolite
	6792201	1981	2	metabolism	Metabolite
	7696318	1995	1	metabolism	Metabolite
	7727400	1995	1	metabolism	Metabolite
	7751290	1995	2	metabolism	Metabolite
	7775463	1995	3	metabolism	Metabolite
	7860596	1995	1	metabolism	Metabolite
	7873670	1995	2	metabolism	Metabolite
	7907888	1994	1	metabolism	Metabolite
	7918446	1994	3	metabolism	Metabolite
	8065908	1994	5	metabolism	Metabolite
	8083170	1994	3	metabolism	Metabolite
	8203917	1994	2	metabolism	Metabolite
	8218300	1993	3	metabolism	Metabolite
	8394006	1993	1	metabolism	Metabolite
	8422384	1993	3	metabolism	Metabolite
	8550422	1996	1	metabolism	Metabolite
	8555191	1996	4	metabolism	Metabolite
	8672449	1996	3	metabolism	Metabolite
	8689233	1996	1	metabolism	Metabolite
	8706750	1996	5	metabolism	Metabolite
	8706854	1996	3	metabolism	Metabolite
	8831972	1997	2	metabolism	Metabolite
	8969520	1997	1	metabolism	Metabolite
	8973190	1997	2	metabolism	Metabolite
	8976565	1997	3	metabolism	Metabolite
	9003442	1997	2	metabolism	Metabolite
	9015391	1997	1	metabolism	Metabolite
	9056491	1997	2	metabolism	Metabolite
	9100006	1997	4	metabolism	Metabolite
	9211277	1997	2	metabolism	Metabolite
	9211333	1997	2	metabolism	Metabolite
	9266856	1997	4	metabolism	Metabolite
	9391059	1998	1	metabolism	Metabolite
	9398312	1998	1	metabolism	Metabolite
	9442062	1998	1	metabolism	Metabolite
	9492273	1998	1	metabolism	Metabolite
	9548961	1998	1	metabolism	Metabolite
	9581571	1998	1	metabolism	Metabolite
	9607323	1998	1	metabolism	Metabolite
	9610360	1998	1	metabolism	Metabolite
	9657697	1998	4	metabolism	Metabolite
	9748544	1998	3	metabolism	Metabolite
	9772162	1998	5	metabolism	Metabolite
	9878448	1999	2	metabolism	Metabolite
	9927652	1999	2	metabolism	Metabolite
	10026151	1999	2	metabolism	Metabolite

Table S9 continued.

Citation	PMID	Year Published	Data Used	Process	Equation
	10074354	1999	1	metabolism	Metabolite
	10225425	1999	1	metabolism	Metabolite
	10231382	1999	1	metabolism	Metabolite
	10353839	1999	4	metabolism	Metabolite
	10387054	1999	2	metabolism	Metabolite
	10406936	1999	2	metabolism	Metabolite
	10480918	1999	2	metabolism	Metabolite
	10529181	1999	1	metabolism	Metabolite
	10542272	1999	1	metabolism	Metabolite
	10642176	2000	2	metabolism	Metabolite
	10757968	2000	1	metabolism	Metabolite
	10769117	2000	5	metabolism	Metabolite
	10769128	2000	1	metabolism	Metabolite
	10820011	2000	1	metabolism	Metabolite
	10821675	2000	1	metabolism	Metabolite
	10828971	2000	1	metabolism	Metabolite
	10831428	2000	3	metabolism	Metabolite
	10989422	2000	12	metabolism	Metabolite
	11084021	2001	1	metabolism	Metabolite
	11094340	2000	1	metabolism	Metabolite
	11106175	2001	2	metabolism	Metabolite
	11237876	2001	4	metabolism	Metabolite
	11342129	2001	1	metabolism	Metabolite
	11380254	2001	1	metabolism	Metabolite
	11488932	2001	4	metabolism	Metabolite
	11515538	2001	1	metabolism	Metabolite
	11527960	2001	1	metabolism	Metabolite
	11604533	2001	1	metabolism	Metabolite
	11686925	2001	1	metabolism	Metabolite
	11724562	2001	2	metabolism	Metabolite
	11736651	2001	2	metabolism	Metabolite
	11741871	2001	5	metabolism	Metabolite
	11747300	2001	2	metabolism	Metabolite
	11747447	2001	1	metabolism	Metabolite
	11803023	2002	2	metabolism	Metabolite
	11814333	2002	1	metabolism	Metabolite
	11934293	2002	1	metabolism	Metabolite
	11967363	2002	3	metabolism	Metabolite
	12021453	2002	2	metabolism	Metabolite
	12097643	2002	2	metabolism	Metabolite
	12135357	2002	3	metabolism	Metabolite
	12356303	2002	1	metabolism	Metabolite
	12370189	2002	2	metabolism	Metabolite
	12383057	2003	4	metabolism	Metabolite
	12517338	2003	1	metabolism	Metabolite
	12535615	2003	4	metabolism	Metabolite
	12627979	2003	3	metabolism	Metabolite
	12637497	2003	7	metabolism	Metabolite
	12706338	2003	6	metabolism	Metabolite
	12711733	2003	1	metabolism	Metabolite
	12805358	2003	2	metabolism	Metabolite
	12824188	2003	2	metabolism	Metabolite
	12937174	2003	2	metabolism	Metabolite
	12975365	2003	1	metabolism	Metabolite
	13785427	1961	1	metabolism	Metabolite

Table S9 continued.

Citation	PMID	Year Published	Data Used	Process	Equation
	14556652	2004	1	metabolism	Metabolite
	14597191	2003	1	metabolism	Metabolite
	14640961	2003	2	metabolism	Metabolite
	14690420	2003	3	metabolism	Metabolite
	14767072	2004	3	metabolism	Metabolite
	14970394	2004	2	metabolism	Metabolite
	14982443	2004	2	metabolism	Metabolite
	15049687	2004	2	metabolism	Metabolite
	15066435	2004	1	metabolism	Metabolite
	15110746	2004	4	metabolism	Metabolite
	15157072	2004	6	metabolism	Metabolite
	15248753	2004	2	metabolism	Metabolite
	15249053	2004	2	metabolism	Metabolite
	15271350	2004	1	metabolism	Metabolite
	15289581	2004	4	metabolism	Metabolite
	15308670	2004	2	metabolism	Metabolite
	15362869	2004	2	metabolism	Metabolite
	15379557	2004	1	metabolism	Metabolite
	15489502	2004	14	metabolism	Metabolite
	15489861	2004	1	metabolism	Metabolite
	15498577	2004	2	metabolism	Metabolite
	15531627	2004	2	metabolism	Metabolite
	15654896	2005	3	metabolism	Metabolite
	15658853	2005	1	metabolism	Metabolite
	15680231	2005	1	metabolism	Metabolite
	15723541	2005	1	metabolism	Metabolite
	15796715	2005	3	metabolism	Metabolite
	15797252	2005	3	metabolism	Metabolite
	15843156	2005	1	metabolism	Metabolite
	15938625	2005	2	metabolism	Metabolite
	15967977	2005	1	metabolism	Metabolite
	16023116	2005	2	metabolism	Metabolite
	16039592	2005	4	metabolism	Metabolite
	16077126	2005	4	metabolism	Metabolite
	16101288	2005	1	metabolism	Metabolite
	16199563	2005	2	metabolism	Metabolite
	16201833	2005	2	metabolism	Metabolite
	16221850	2005	1	metabolism	Metabolite
	16297670	2006	3	metabolism	Metabolite
	16427816	2006	2	metabolism	Metabolite
	16436710	2006	2	metabolism	Metabolite
	16464851	2006	2	metabolism	Metabolite
	16480719	2006	1	metabolism	Metabolite
	16567800	2006	3	metabolism	Metabolite
	16672604	2006	3	metabolism	Metabolite
	16731973	2006	2	metabolism	Metabolite
	16766526	2006	4	metabolism	Metabolite
	16793549	2006	2	metabolism	Metabolite
	16813561	2006	4	metabolism	Metabolite
	16829675	2006	3	metabolism	Metabolite
	16843487	2006	1	metabolism	Metabolite
	16864571	2006	2	metabolism	Metabolite
	16966333	2006	1	metabolism	Metabolite
	16981730	2006	3	metabolism	Metabolite
	16990279	2006	15	metabolism	Metabolite

Table S9 continued.

Citation	PMID	Year Published	Data Used	Process	Equation
	16990935	2006	2	metabolism	Metabolite
	17059210	2006	1	metabolism	Metabolite
	17079236	2006	2	metabolism	Metabolite
	17123542	2006	2	metabolism	Metabolite
	17176045	2007	2	metabolism	Metabolite
	17240978	2007	1	metabolism	Metabolite
	17309433	2007	1	metabolism	Metabolite
	17401542	2007	3	metabolism	Metabolite
	17442255	2007	2	metabolism	Metabolite
	17535911	2007	1	metabolism	Metabolite
	17542990	2007	3	metabolism	Metabolite
	17557829	2007	6	metabolism	Metabolite
	17559838	2007	1	metabolism	Metabolite
	17568739	2007	3	metabolism	Metabolite
	17616624	2007	1	metabolism	Metabolite
	17635929	2007	2	metabolism	Metabolite
	17698004	2007	3	metabolism	Metabolite
	17947381	2007	1	metabolism	Metabolite
	17996716	2008	1	metabolism	Metabolite
	18056714	2008	1	metabolism	Metabolite
	18081839	2008	2	metabolism	Metabolite
	18083802	2008	3	metabolism	Metabolite
	18085797	2008	1	metabolism	Metabolite
	18284213	2008	1	metabolism	Metabolite
	18353368	2008	6	metabolism	Metabolite
	18390652	2008	1	metabolism	Metabolite
	18411271	2008	2	metabolism	Metabolite
	18458150	2008	4	metabolism	Metabolite
	18459799	2008	4	metabolism	Metabolite
	18567546	2008	2	metabolism	Metabolite
	18576672	2008	1	metabolism	Metabolite
	18754693	2008	2	metabolism	Metabolite
	18837509	2008	1	metabolism	Metabolite
	18990827	2008	2	metabolism	Metabolite
	19073594	2009	8	metabolism	Metabolite
	19099445	2009	1	metabolism	Metabolite
	19140736	2009	4	metabolism	Metabolite
	19307712	2009	2	metabolism	Metabolite
	19370061	2009	4	metabolism	Metabolite
	19384989	2009	4	metabolism	Metabolite
	19583219	2009	1	metabolism	Metabolite
	19664587	2009	1	metabolism	Metabolite
	19678710	2009	1	metabolism	Metabolite
	19797049	2009	2	metabolism	Metabolite
	19883126	2009	1	metabolism	Metabolite
	20050615	2010	1	metabolism	Metabolite
	20136146	2010	1	metabolism	Metabolite
	20160120	2010	2	metabolism	Metabolite
	20170126	2010	4	metabolism	Metabolite
	20364820	2010	1	metabolism	Metabolite
	20416269	2010	2	metabolism	Metabolite
	20452364	2010	1	metabolism	Metabolite
	20480490	2010	1	metabolism	Metabolite
	20498377	2010	2	metabolism	Metabolite
	20553497	2010	1	metabolism	Metabolite

Table S9 continued.

Citation	PMID	Year Published	Data Used	Process	Equation
	20717852	2011	1	metabolism	Metabolite
	20853825	2010	6	metabolism	Metabolite
	20887711	2010	2	metabolism	Metabolite
	21046341	2010	10	metabolism	Metabolite
	21059411	2011	7	metabolism	Metabolite
	21081107	2011	2	metabolism	Metabolite
	21119630	2010	3	metabolism	Metabolite
	21169495	2011	2	metabolism	Metabolite
	21278708	2011	2	metabolism	Metabolite
	21279421	2011	1	metabolism	Metabolite
	21296885	2011	1	metabolism	Metabolite
	21298178	2011	3	metabolism	Metabolite
	21669179	2011	2	metabolism	Metabolite
	21696459	2011	1	metabolism	Metabolite
	21704016	2011	1	metabolism	Metabolite
	21832062	2011	2	metabolism	Metabolite
	21928762	2011	2	metabolism	Metabolite
	21998098	2011	1	metabolism	Metabolite
	22017312	2011	2	metabolism	Metabolite
	22059588	2011	1	metabolism	Metabolite
	22138634	2012	2	metabolism	Metabolite
	22173092	2012	1	metabolism	Metabolite
	22411989	2012	3	metabolism	Metabolite
	22443469	2012	1	metabolism	Metabolite
	22471615	2013	1	metabolism	Metabolite
	22970852	2012	1	metabolism	Metabolite
	22984449	2012	1	metabolism	Metabolite
	23001854	2013	1	metabolism	Metabolite
	23018273	2012	1	metabolism	Metabolite
	23237860	2013	3	metabolism	Metabolite
	23484010	2013	1	metabolism	Metabolite
	23564174	2013	1	metabolism	Metabolite
	23879525	2013	2	metabolism	Metabolite
	23935849	2013	3	metabolism	Metabolite
	24123841	2013	1	metabolism	Metabolite
	24210219	2013	3	metabolism	Metabolite
	24479701	2014	2	metabolism	Metabolite
	24509771	2014	8	metabolism	Metabolite
	24530526	2014	1	metabolism	Metabolite
	25160617	2014	6	metabolism	Metabolite
	25225400	2014	1	metabolism	Metabolite
	25484615	2014	1	metabolism	Metabolite
Barbas, C.F. et al. J. Am. Chem. Soc. 112 2013-2014 1990		1990	1	metabolism	Metabolite
Chiu, T.H. et al. Methods Enzymol. 42C 264-269 1975		1975	1	metabolism	Metabolite
Kiick, K.L. et al. Tetrahedron 56 9487-9493 2000		2000	2	metabolism	Metabolite
Kruger, N.J. Biochem. Soc. Trans. 17 760-761 1989 Escherichia coli ; 0 0		1989	1	metabolism	Metabolite
Mino, K et al. 2014. Bioscience, Biotechnology, and Biochemistry		2014	3	metabolism	Metabolite
Sheperdson, M.; Pardee, A.B. J. Biol. Chem. 235 3233-3237 1960		1960	1	metabolism	Metabolite
Zhang, Q.; Liu, H.w. J. Am. Chem. Soc. 122 9065-9070 2000		2000	1	metabolism	Metabolite
http://www.ncbi.nlm.nih.gov/pubmed/2994567		1985	1	signal transduction	RNA
http://gene.bio.jhu.edu/Ourspdf/89.pdf	9367758	1997	1	signal transduction	RNA
http://jb.asm.org/content/177/10/2798.full.pdf+html		1995	1	signal transduction	RNA
http://jb.asm.org/content/179/1/228.full.pdf		1997	1	signal transduction	RNA
http://jb.asm.org/content/194/18/5110.full.pdf		2012	1	signal transduction	RNA
http://journals.plos.org/plosone/article?id=10.1371/journal.pone.0057712		2013	1	signal transduction	RNA

Table S9 continued.

Citation	PMID	Year Published	Data Used	Process	Equation
http://link.springer.com/chapter/10.1007%2F7171_2006_088		2007	1	signal transduction	RNA
http://mmbr.asm.org/content/57/2/320.full.pdf		1993	2	signal transduction	RNA
http://nar.oxfordjournals.org/content/early/2012/12/14/nar.gks1207.full.pdf		2012	1	signal transduction	RNA
http://nar.oxfordjournals.org/content/early/2015/01/05/nar.gku1374.full		2015	1	signal transduction	RNA
http://pubs.acs.org/doi/pdf/10.1021/bi901158h		2009	1	signal transduction	RNA
http://pubs.rsc.org/En/content/articlepdf/2015/mt/c4mt00180j		2014	2	signal transduction	RNA
http://www.horizonpress.com/cimb/v/v8/04.pdf		2006	9	signal transduction	RNA
http://www.jbc.org/content/261/1/238.full.pdf		1986	1	signal transduction	RNA
http://www.jbc.org/content/261/23/10936.full.pdf		1986	1	signal transduction	RNA
http://www.jbc.org/content/272/20/13026.full.pdf		1997	2	signal transduction	RNA
http://www.jbc.org/content/272/22/14257.long		1997	2	signal transduction	RNA
http://www.jbc.org/content/272/27/16962.full.pdf		1997	1	signal transduction	RNA
http://www.jbc.org/content/272/3/1440.long		1997	1	signal transduction	RNA
http://www.jbc.org/content/273/50/33652.full.pdf		1998	1	signal transduction	RNA
http://www.jbc.org/content/281/50/38189.full.pdf		2006	1	signal transduction	RNA
http://www.jbc.org/content/282/22/16476.full.pdf		2007	4	signal transduction	RNA
http://www.jbc.org/content/284/32/21218.full.pdf		2009	2	signal transduction	RNA
http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2373468/		2002	1	signal transduction	RNA
http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2435081/		2008	2	signal transduction	RNA
http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2561245/		2008	2	signal transduction	RNA
http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2655528/		2009	1	signal transduction	RNA
http://www.ncbi.nlm.nih.gov/pmc/articles/PMC400900/pdf/emboj00127-0331.pdf		1989	1	signal transduction	RNA
http://www.ncbi.nlm.nih.gov/pmc/articles/PMC94037/		1999	4	signal transduction	RNA
http://www.ncbi.nlm.nih.gov/pubmed/15380549		2004	1	signal transduction	RNA
http://www.ncbi.nlm.nih.gov/pubmed/17644067		2007	3	signal transduction	RNA
http://www.ncbi.nlm.nih.gov/pubmed/18532985		2008	1	signal transduction	RNA
http://www.ncbi.nlm.nih.gov/pubmed/22148640		2012	1	signal transduction	RNA
http://www.ncbi.nlm.nih.gov/pubmed/23075318		2013	1	signal transduction	RNA
http://www.ncbi.nlm.nih.gov/pubmed/23941567		2013	2	signal transduction	RNA
http://www.ncbi.nlm.nih.gov/pubmed/24948475		2014	3	signal transduction	RNA
http://www.ncbi.nlm.nih.gov/pubmed/25369000		2014	1	signal transduction	RNA
http://www.ncbi.nlm.nih.gov/pubmed/8559067		1995	3	signal transduction	RNA
http://www.ncbi.nlm.nih.gov/pubmed/9278422		1997	1	signal transduction	RNA
http://www.pnas.org/content/101/33/12148.full.pdf		2004	1	signal transduction	RNA
http://www.pnas.org/content/94/7/2957.full.pdf		1997	1	signal transduction	RNA
http://www.pnas.org/content/96/17/9833.full.pdf		1999	1	signal transduction	RNA
http://www.pnas.org/content/98/2/435.full.pdf		2001	1	signal transduction	RNA
http://www.sciencedirect.com/science/article/pii/S0022283603002626		2003	1	signal transduction	RNA
http://www.sciencedirect.com/science/article/pii/S2211546314000369		2014	2	signal transduction	RNA
http://www.sciencedirect.com/science/bookseries/00702161		1990	2	signal transduction	RNA
http://www.sciencemag.org/content/292/5526/2488.long		2001	1	signal transduction	RNA
http://www.sciencemag.org/content/301/5638/1383.long		2003	1	signal transduction	RNA

Movie S1. Heterogeneous data integration via a large-scale model. This video shows the production, over several decades, of a subset of the data used as an input to our model. Each paper is depicted as a circle; once all of the circles have been generated, the size of the circles is briefly altered (on a log scale) to reflect the difference in data content between studies (e.g., scientists using early biochemical techniques might report a handful of data points in one manuscript, while those using RNA-Seq might report thousands or millions). The circles are then grouped and integrated into mathematical equations; for simplicity, only those related to RNA, proteins and small molecules are shown here. These equations are then themselves integrated into a unified mathematical model. More details about data sources shown in the movie are listed in Table S9.

Movie S2. A simulation of the shift from minimal to amino acid-supplemented media conditions. First, the growth and replication of the cell is depicted under minimal media + glucose conditions, as simulated by our model. Once amino acids are added to the media, a number of signaling systems are activated, which leads to a change in transcription factor activity, gene expression and metabolic flux. These changes further lead to an increase in ribosome production, as well as a larger and faster-growing cell that utilizes multiple replication forks to achieve its higher growth rate.

References

1. J. Gunawardena, *Nature biotechnology* **30**, 838 (2012).

2. A. Jain, Fireworks 1.6.2 documentation, https://materialsproject.github.io/fireworks/.

3. J. R. Karr, *et al.*, *Cell* **150**, 389 (2012).

4. Y. Tanouchi, *et al.*, *Nature* **523**, 357 (2015).

5. J. T. Sauls, D. Li, S. Jun, *Current opinion in cell biology* **38**, 38 (2016).

6. H. Bremer, G. Churchward, *Journal of Theoretical Biology* **69**, 645 (1977).

7. H. Bremer, P. P. Dennis, *EcoSal Plus* **3** (2008).

8. J. A. Bernstein, A. B. Khodursky, P.-H. Lin, S. Lin-Chao, S. N. Cohen, *Proceedings of the National Academy of Sciences* **99**, 9697 (2002).

9. J. B. Russell, G. M. Cook, *Microbiological reviews* **59**, 48 (1995).

10. G.-W. Li, D. Burkhardt, C. Gross, J. S. Weissman, *Cell* **157**, 624 (2014).

11. A. Schmidt, *et al.*, *Nature biotechnology* **34**, 104 (2016).

12. J. W. Tobias, T. E. Shrader, G. Rocap, A. Varshavsky, *Science (New York, N.Y.)* **254**, 1374 (1991).

13. I. M. Keseler, *et al.*, *Nucleic acids research* **41**, D605 (2013).

14. E. W. Birch, M. Udell, M. W. Covert, *Journal of Theoretical Biology* **345**, 12 (2014).

15. D. Weaver, I. Keseler, A. Machkie, I. Paulsen, P. Karp, *BMC Systems Biology* **8** (2014).

16. J. D. Orth, *et al.*, *Molecular systems biology* **7**, 535 (2011).

17. B. D. Bennett, *et al.*, *Nature chemical biology* **5**, 593 (2009).

18. D. S. Weaver, I. M. Keseler, A. Mackie, I. T. Paulsen, P. D. Karp, *BMC Systems Biology* **8**, 79 (2014).

19. I. Schomburg, *et al.*, *Nucleic acids research* **41**, D764 (2013).

20. A. Bar-Even, *et al.*, *Biochemistry* **50**, 4402 (2011).

21. P. D. Karp, *et al.*, *EcoSal Plus* (2014).

22. Y. Toya, *et al.*, *Biotechnology progress* **26**, 975 (2010).

23. A. M. Feist, *et al.*, *Molecular systems biology* **3**, 121 (2007).

24. F. R. Blattner, *et al.*, *Science (New York, N.Y.)* **277**, 1453 (1997).

25. H. Bremer, P. Dennis, *Escherichia coli and Salmonella: cellular and molecular biology,* **2**, 1553 (1996).

26. W. D. Donachie, *The American Journal of Gastroenterology* **219**, 1077 (1968).

27. F. C. Neidhardt, J. L. Ingraham, M. Schaechter, *Physiology of the bacterial cell: a molecular approach* (Sinauer Associates Sunderland, MA, 1990).

28. M. Wallden, D. Fange, E. G. Lundius, Ö. Baltekin, J. Elf, *Cell* **166**, 729 (2016).

29. H. Brunschede, T. Dove, H. Bremer, *Journal of Bacteriology* **129**, 1020 (1977).

30. G. Reshes, S. Vanounou, I. Fishov, M. Feingold, *Physical Biology* **5** (2008).

31. H. Bremer, P. P. Dennis, *et al.*, *Escherichia coli and Salmonella: cellular and molecular biology* **2**, 1553 (1996).

32. M. Scott, C. W. Gunderson, E. M. Mateescu, Z. Zhang, T. Hwa, *Science* **330**, 1099 (2010).

33. M. Wallden, D. Fange, E. G. Lundius, Ö. Baltekin, J. Elf, *Cell* **166**, 729 (2016).

34. J. R. Karr, *et al.*, *Cell* **150**, 389 (2012).

35. M. Campos, *et al.*, *Cell* **159**, 1433 (2014).

36. H. M. T. Y. O. Y. B. M. D. K. T. M. . W. B. Baba T, Ara T, M. H, *Molecular Systems Biology* **2**, 2006.0008 (2006).

37. A. R. Joyce, *et al.*, *Journal of bacteriology* **188**, 8259 (2006).

38. F. C. Neidhardt, P. L. Bloch, D. F. Smith, *Journal of bacteriology* **119**, 736 (1974).

39. B. Bushnell, Bbmap short read aligner, and other bioinformatics tools, http://sourceforge.net/projects/bbmap (2016).

40. B. Li, C. N. Dewey, *BMC bioinformatics* **12**, 323 (2011).

41. M. Kitagawa, *et al.*, *DNA research* **12**, 291 (2005).

42. M. MR, *Experientia* **48**, 178 (1992).

43. K. Nath, A. L. Koch, *Journal of Biological Chemistry* **245**, 2889 (1970).

44. H. Pang, H. Winkler, *Molecular Microbiology* **12**, 115 (1994).

45. D. Kennel, *Journal of Molecular Biology* **34**, 85 (1968).

46. M. W. Covert, E. M. Knight, J. L. Reed, M. J. Herrgard, B. O. Palsson, *Nature Letter* **429**, 92 (2004).

47. F. Mohammad, R. Green, A. Buskirk, *eLIFE* **8**, 1 (2019).

48. A. Beck, K. Hunt, R. Carlson, *Processes* **6**, 38 (2018).

49. J. Moffitt, S. Pandey, A. Boettiger, S. Wang, X. Zhuang, *eLIFE* **5**, 1 (2016).

50. S. Pedersen, *EMBO Journal* **3**, 2895 (1984).

51. X. Dai, *et al.*, *Nature Microbiology* **2** (2016).

52. F. Neidhardt, H. Umbarger, *Escherichia coli and Salmonella: Cellular and Molecular Biology*, F. Neidhardt, ed. (American Society of Microbiology, 1996), chap. 3, p. 2.