Supplementary Material

Evolution Shapes Enzyme Turnover Numbers to Support Cellular Objectives

Samira L. van den Bogaard[a], Lorenzo Wormer[b,c], Nadja A. Henke[b], Lars M. Blank[a,d], and Tobias B. Alter[a,*]

[a]Institute of Applied Microbiology, Aachen Biology and Biotechnology;
RWTH Aachen University, Aachen, Germany

[b]Institute of Process Engineering in Life Sciences;
Karlsruhe Institute of Technology, Karlsruhe, Germany

[c]Institute of Functional Interfaces;
Karlsruhe Institute of Technology, Eggenstein-Leopoldshafen, Germany

[d]WSS Research Center Catalaix;
RWTH Aachen University, Aachen, Germany

[*]Corresponding author (tobias.alter@rwth-aachen.de)

April 20, 2026

Contents

1 Supplementary Methods

1.1 Updated version of the *E. coli* Protein Allocation Model

The original Protein Allocation Model (PAM), described by Alter et al. (2021) [1], incorporates three coarse-grained protein sectors which are related to the metabolic phenotype of the organism under study. These sectors include (1) the active enzyme sector, which adds enzyme kinetics to the reaction stoichiometry present in metabolic models, (2) the translational protein sector, which describes proteins required for translation, and (3) the unused enzymes sector, including the proteins which are not used at their full capacity. The unused enzymes sector compensates for enzymes that are not operating at their maximum *in vivo* efficiency. As all condition-dependent fluctuations in protein efficiency are compensated for by the unused enzymes sector, we can assume that all enzymes in the active enzyme sectors operate at their maximum *in vivo* turnover rate, and the k_{cat} values are *in vivo* maximal turnover numbers under biologically feasible substrate and product concentrations. Recent advancements in our understanding of protein allocation have led to significant updates, reflecting improved correlations with fundamental principles of research allocation and experimental findings.

Besides revision of the sectors, additional constraints were added to allow for more the complexity of protein-reaction associations present in real biological systems. The integration of isozymes, promiscuous enzymes and enzyme complexes increased to protein burden of biomass production, resulting in a lower

final growth rate and an earlier start of overflow metabolism compared to the original model structure. In the updated PAM structure, turnover numbers or k_{cat} values represent the maximum *'in vivo'* catalytic rate per enzyme taking kinetic effects, such as reversibility, into account. As such, the genome-scale set of k_{cat} values needs to be parametrized, i.e., adapted to the new model structure.

1.1.1 Configuration of protein sectors

In the previous version of the PAM, the protein fraction allocated of the translational sector was related to the growth rate. Using experimental data from Schmidt et al. (2016) [2], we find that translational protein abundance does not exhibit a strictly linear correlation with growth rate across the full range of physiologically plausible growth rates (Supplementary Figure S3), but there is a linear relation between substrate uptake rate and translational proteins. In addition, You and colleagues (2013) showed that the fraction of ribosomal proteins is indirectly related to the glucose uptake rate by a regulatory network mediated by CRP-cAMP [3]. Furthermore, substrate availability has been shown to be correlated to synthesis of ribosomes and rRNA through ppGpp [4, 5]. Indeed, relating the translational protein fraction seems to yield a better fit to quantitative proteomics data (Supplementary Figure S3).

To correct for un- and underused enzymes, the PAM includes an unused enzymes sector, based on the findings of O'Brien and colleagues (2015) [1, 6]. This sector enables us to assume all the enzymes in the metabolic sector are operating at their maximal *in vivo* catalytic efficiency under biologically plausible conditions. The previous version of the PAM assumed the unused protein fraction to be zero at the maximal growth rate. The unused proteins consists out of those proteins which do not result in a change of phenotype, e.g. exchange fluxes or growth rate, when the amount of that protein is decreased [7]. Based on this definition, we can assume that the amount of unused proteins at zero growth equals the maximal amount of overexpressed protein that leads to zero growth. Furthermore, there are strong suggestions that there is still a small fraction of unused proteins left at the maximum growth rate, which are related to bet-hedging strategies to ensure optimal growth under changing conditions [7, 8]. To correct for this bet-hedging behaviour, we make the following assumptions: (1) 37% of the metabolic proteins is unused at zero growth [9], and (2) if strains are subjected to adaptive laboratory evolution all metabolic proteins are performing at their full '*in vivo*' capacity at the maximum growth rate [8, 10]. Based on (1), a lower bound for the number of enzymes in the unused enzymes sector can be defined. However, the fraction of the underused metabolic enzymes at zero growth is unknown, condition dependent, and quantification requires absolute proteomics across conditions [6]. The total unused enzymes sector at zero growth can be optimized after consolidating the qualitative k_{cat} distribution, with (1) providing a close enough starting point for generating a PAM.

For *E. coli* the maximal growth rate was assumed to be 1.63 doublings per hour, corresponding to $1.63 \cdot \ln(2) = 1.13$ 1/h [10]. Although the we can

Table S1: Novel parametrization of the *Escherichia coli)* Protein Allocation Model, including the relation between the translational protein fraction and the substrate uptake rate (v_s), and between the unused protein fraction and the growth rate (μ).

Protein Sector	**Relation**	**Units**	**Reference**
Translational Proteins ($E_{translation}$)	$3.9 \cdot 10^{-2} + 3.7 \cdot 10^{-3} v_{glc}$	Intercept: $g_{protein}/mmol_{glc}$ Slope: $g_{protein}/mmol_{glc}/h$	[2, 3]
Unused Enzymes (E_{unused})	$8.8 \cdot 10^{-3} - 0.13 v_{glc}$	Intercept: $g_{protein}/mmol_{glc}$ Slope: $g_{protein}/mmol_{glc}/h$	[8–10]

use effect of ALE and recombinant protein expression on the growth rate to derive a relation between the unused enzyme sectors and the growth rate, the amount of active proteins is determined by the substrate uptake rate via cAMP regulation [1,6]. As a consequence, the unused enzyme sector is linearly related to the substrate uptake rate. The new parameter sets are integrated in the PAM as described by Alter, et al. (2021) [1]. The updated relations are summarized in Supplementary Table S1.

1.1.2 Complex protein-reaction associations

The previous version of the PAM makes use of the central dogma in its most modest form: each enzyme is associated to one single reaction and each reaction is associated to a single enzyme. This is not relevant in biological settings as most protein-reaction relationships are more complex [11]. The protein-reaction associations can take multiple forms: isozymes (a single reaction can be catalyzed by multiple proteins), promiscuous enzymes (a single enzyme can catalyze multiple reactions), and enzyme complexes (multiple enzymes combined catalyze a single reaction). The separate forms are defined by AND (additive), and OR (exclusive) relations, or a combination thereof. The updated version of the PAM includes a single variable for each enzyme-reaction relation, which together determine the flux through that reaction.

$$\begin{aligned} v_{rxn,i} &= \sum_{j \in \text{enz},i} v_{catalysis,i,j} \\ v_{catalysis,i,j} &= k_{\text{cat},i,j} E_j \end{aligned} \tag{1}$$

where reaction i is catalyzed by enzyme E_i enzymes, and $v_{catalysis,i,j}$ represents the flux through reaction i, which is established by catalysis of enzyme j. The protein-reaction relation (isozyme, promiscuous enzyme, or enzyme complex), determines the exact mathematical implementation of the general rule in Eq 1.

1. **Isozymes**. If multiple isozymes are associated with a single reaction, they can independently catalyze a reaction. As a consequence, the summed catalytic potential of the isozymes limits the upper bound of the associated reaction.

$$
\begin{aligned}
v_{rxn,i} &= v_{catalysis,i,j} + v_{catalysis,i,k} \\
v_{catalysis,i,j} &= k_{\mathrm{cat},i,j} E_j \\
v_{catalysis,i,k} &= k_{\mathrm{cat},i,j} E_k
\end{aligned}
\tag{2}
$$

where reaction i is catalyzed by two different enzymes j and k. Each reaction-enzyme relation has a separate catalytic efficiency defined by $k_{cat,i,j}$ and $k_{cat,i,k}$, respectively.

2. **Promiscuous enzymes**. In the current definition of the enzyme-reaction associations, a single enzyme can be associated with two constraints, related to two protein-reaction associations.

$$
\begin{aligned}
v_{catalysis,i,j} &= k_{\mathrm{cat},i,j} E_j \\
v_{catalysis,l,j} &= k_{\mathrm{cat},l,j} E_j
\end{aligned}
\tag{3}
$$

Here, both reaction i and l are catalyzed by the same enzyme, again with reaction-enzyme specific catalytic efficiencies.

3. **Enzyme complexes**. Enzyme complexes represent an additive (AND) relation between two or more polypeptides. The combination of the polypeptides results in a functional enzyme. Thus, the proteins which are related by an AND relation are not independent of each other. In this case, the protein resources allocated required to catalyze the associated reaction, depends on concentrations of the of all peptides involved in the enzyme complex. Therefore, each enzyme complex is represented in the model as a single, lumped, enzyme variable, representing all enzyme instances included in the AND relationship, with the molmass being the sum of all individual peptide molmasses.

$$
\mathrm{molmass}_{enzymecomplex} = \sum_{j \in \mathrm{enz}} \mathrm{molmass}_j \tag{4}
$$

Apart from the biological relevant enzyme motifs discussed previously, the relation between reactions and proteins is complicated because of the mathematical definition of reaction directionality. In the PAM all reactions are irreversible, reversible reactions are split in a forward and reverse reactions. Similarly, each protein has a forward and reverse variable, which are related to the forward and reverse components of a reaction, respectively.

$$
v_{rxn,i} = v_{rxn,i,fwd} - v_{rxn,i,rev} \tag{5}
$$

$$
E_i = E_{i,fwd} + E_{i,rev} \tag{6}
$$

$$
\begin{aligned}
\frac{1}{k_{cat,fwd,i}} v_{rxn,i,fwd} &<= E_{i,fwd} \\
\frac{1}{k_{cat,rev,i}} v_{rxn,i,rev} &<= E_{i,rev}
\end{aligned}
\tag{7}
$$

This formulation allows for separation of the forward and reverse components of enzyme catalysis.

1.2 A pipeline for the parametrization of Protein Allocation Models

The entire parametrization framework is split in three steps: (1) preprocessing, (2) parametrization, and (3) analysis. During preprocessing, the sectors are initialized. The slope and the intercept for the translational protein sector and unused enzymes sectors are obtained by fitting to experimental data, or obtained from a closely related model organism. Also, the active enzyme sector is initialized using *in silico* k_{cat} values from machine learning algorithms, and possibly changing the median k_{cat}. After the protein sectors are initialized, a PAM is build and the k_{cat} values are optimized using the PAMparametrizer. After optimizing the qualitative relations between the k_{cat} values of the enzymes, the quantitative protein burden is optimized to match experimental data by optimizing the unused enzyme sector at zero growth (more details in 1.1.1). The results of one or multiple PAMparametrizer runs can be subsequently analyzed, and compared, and the most accurate models should be selected for further validation. An overview of the parametrization workflow is given in Figure S1.

1.2.1 Initializing the coarse-grained protein sectors

As described by Alter et al. (2021), the translational protein sector and unused enzymes sector are coarse-grained protein sectors representing proteins required for translation, and proteins which are underused, respectively [1]. The relation between the protein sectors and the phenotype can either be determined using experiments, or can be assumed to be similar to the relations in *E. coli* or another closely related model organism. Both relations are first defined as functions of the growth rate, which will be converted to functions of the substrate uptake rate of all provided substrates in the first step of the PAMparametrizer.

To obtain the relation between the translational protein sector and the growth rate rate, quantitative proteomics data at multiple growth rates, before overflow-like phenotypes, are required. For the unused enzyme sector, the fraction of heterologous protein can be used to assess the amount of unused proteins at zero growth, as described in the previous section 1.1.1. Examples of how to perform overexpression experiments are an eGFP overexpression assay in *E. coli* [12], or the analysis of the effect of mCherry overexpression on growth rate in *Saccharomyces cerevisiae* (*S. cerevisiae*) [13]. More details about the procedure can be found in the documentation and notebooks of the GitHub repository (`https://github.com/iAMB-RWTH-Aachen/PAMParametrizer`).

1.2.2 Initializing the active enzyme sector

To initiate the active enzyme sector, an estimation of the parameter values, and the gene-protein-reaction associations of the metabolic reactions are required.

The initial k_{cat} values in this study were obtained from the GotEnzymes web interface [14], and mapped to UniProt identifiers [15]. To enhance the starting position for optimization, the k_{cat}s can be multiplied by a factor selected after scanning a range of potential values.

1.2.3 The PAMparametrizer framework: optimization of k_{cat} values

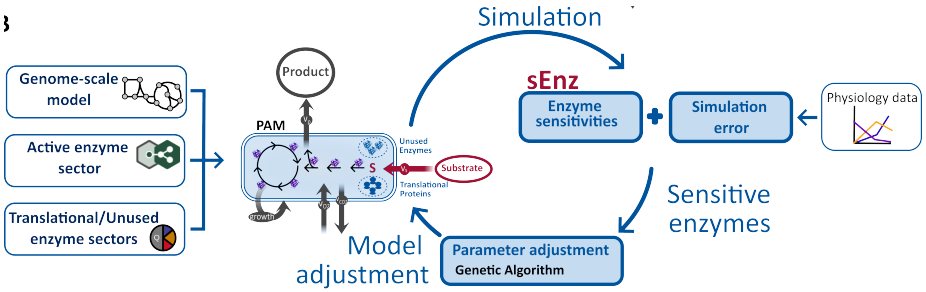

Figure S1: Overview of the PAMparamertizer workflow.

After a PAM is setup with coarse-grained enzyme sectors for each substrate and an initial set of turnover numbers, the turnover numbers have to be adjusted to correctly simulate experimentally observed behaviour. The PAMparametrizer aims to minimize the error between the simulated and experimentally measured fluxes by altering the kinetic parameters stored in the active enzyme sector of the protein allocation model. Based on the results of simulations over a range of substrate uptake rates, the enzymes which have the most control over the biomass production rate are selected for optimization. The k_{cat} values of these enzymes are subsequently optimized in a genetic algorithm. The best solution is used to reparametrize the model, and to determine the model performance. The parametrization workflow will repeat until a user-defined performance or maximal number of iterations is reached. After consolidating the k_{cat} distribution, the amount of unused enzyme sector at zero growth is fitted to match the experimental data, using 37% of the total protein amount as lower bound. In the following sections, each step in the parametrization workflow will be discussed. An overview of the parametrization workflow is given in S1.

1.2.4 Sector configuration

The coarse-grained protein sectors parameters are linearly related to the substrate uptake rate, and the sector parameters are thus dependent on the substrate. The PAMparametrizer automatically establishes the relation between the substrate uptake rate and the protein sector using the relation between the substrate uptake rate and growth rate as computed by a genome-scale model (without protein limitation). From the relationship between the growth rate and substrate uptake rate, the linear fit of the substrate uptake rate and respective protein sector can be obtained for all substrates.

1.2.5 Parameter optimization

After initialization of the model, the k_{cat} values are further optimized. The optimization of the parameters can be separated in four steps.

1. Selecting the most influential enzymes
2. Optimization of the selected k_{cat} values
3. Model reparametrization
4. Fitness evaluation

These steps are repeated until either the maximal number of iterations, or a predefined fitness is reached. If the error converges, all k_{cat} values in the model are optimized.

1.2.6 Selecting the most influential enzymes

To decrease the complexity of the optimization, a predefined number of enzymes are selected for parameter optimization. Selecting the enzymes is based enzyme sensitivity coefficients calculated using sEnz [16]. These sensitivity coefficients relate to the effect of changes in the enzyme concentration on the growth rate. As the enzyme sensitivity may change with changing conditions, simulations over a range of substrate uptake rates are used to find the most important enzymes. The enzymes with the highest mean enzyme sensitivity coefficient over all simulations are optimized.

1.2.7 Optimization of selected parameters

The k_{cat} values for each forward and reverse reaction associated with the selected enzymes are modified in a genetic algorithm for parameter optimization (see Supplementary Methods 1.3). This genetic algorithm uses random mutations, crossover, elitism and selection to change the parameter set, aiming to decrease the distance between simulation results and experimental data. The PAM is subsequently updated with the new set of parameters and the fidelity of the model is evaluated.

1.2.8 Determining model fitness

As well the genetic algorithm, as the PAMparametrizer aim to improve the prediction fidelity of the PAM. The difference between the model predictions and experimental measurements are determined by calculating the coefficient of determination (R^2) between the model simulations and experimental data. The R^2 value quantifies how much of the variance in the experimental data is explained by the model predictions, serving as a key metric for evaluating simulated fluxes with experimental measurements. Therefore, maximizing the mean R^2 of all data points is the objective in both algorithms.

$$R^2 = 1 - \frac{RSS}{TSS}$$

$$RSS = \sum (v_{sim} - v_{exp})^2 \tag{8}$$

$$TSS = \sum (\overline{v_{exp}} - v_{exp})^2$$

Where RSS is the residual sum of squared, that is the squared difference between simulated (v_{sim}) and experimentally measured (v_{exp}) fluxes. TSS represents the total sum of squares, representing the variability in the data, e.g. the squared difference between the mean of all measurements ($\overline{v_{exp}}$) and the individual datapoints. For large datasets, the experimental measurements used by the genetic algorithm to determine the R^2, are uniformly sampled to decrease the runtime. Furthermore, if simulation of the production of byproducts not observed experimentally can be penalized by setting the measured exchange flux for these byproducts to zero.

1.3 Genetic Algorithm for kinetic parameter optimization

A genetic algorithm is an optimization technique inspired by the principles of natural selection and genetics. It iteratively evolves a population of candidate solutions to a given problem through processes such as selection, crossover, and mutation, aiming to improve their fitness. Genetic algorithms are capable of efficiently searching large and complex solution spaces, often finding near-optimal solutions for problems where traditional methods may struggle [17]. Each genetic algorithm consists of (1) a set of variables to optimize, (2) populations of individuals, where an individual is an independent set of variables, (3) a fitness function, used as optimization objective, and (4) methods to create new individuals from the existing ones [17].

In short, the genetic algorithm creates populations of individuals. During one generation of a genetic algorithm, the fittest individuals are selected and modified using crossover and mutation, followed by calculation of the fitness of each individual. After a number of generations, the populations are combined and reshuffled, and the workflow is repeated. Each of these iterations is called a gene flow event. After a number of gene flow events, the final population is saved and the best individual is selected. An overview is given in Figure S2, and the associated hyperparameters are summarized in Table S2. The relevant packages and functionalities of the software implementation are shown in Table S3.

The implementation of the genetic algorithm for kinetic parameter optimization is inspired by the genetic algorithm for metabolic engineering by Alter et al. (2018) [18]. In the following sections, we describe the details of the genetic algorithm for kinetic parameter optimization.

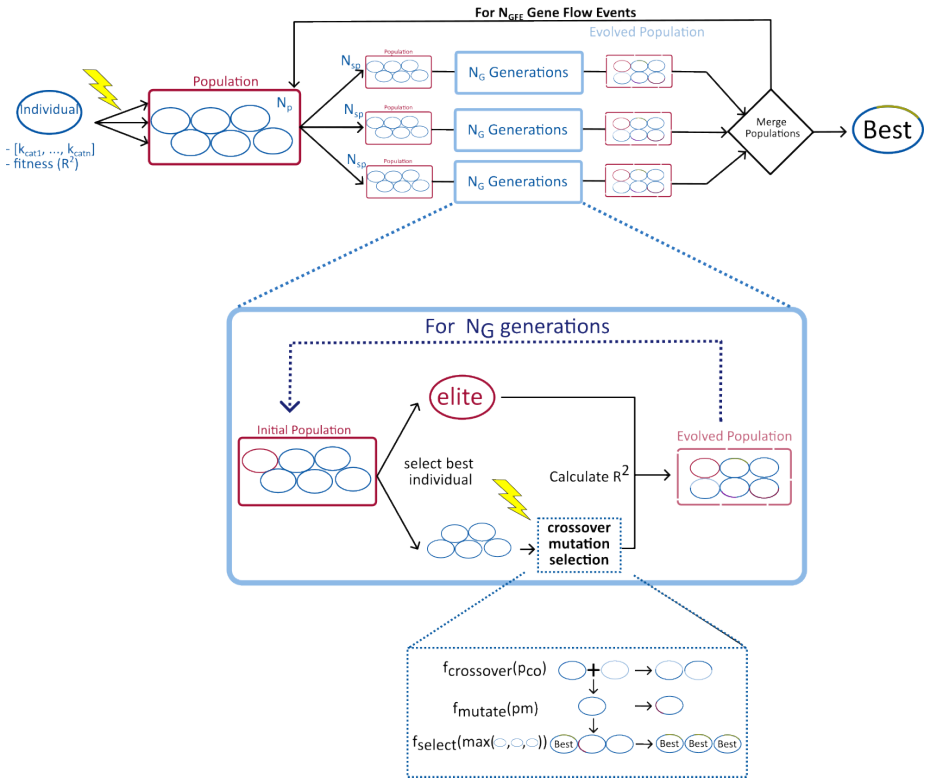

Figure S2: Overview of the Genetic Algorithm used for the optimization of k_{cat} values. Each individual is associated with a list of k_{cat} values. First, a population of N_P individuals is created by mutating the original list of k_{cat}s. This population is split into different subpopulations of N_{sp} individuals, marking the start of the first gene flow event. After the best solution ('elite') is saved, the individuals of each subpopulation undergo crossover, mutation, and selection, based on a crossover p_{co}, and mutation probability p_m, for N_G generations. After N_G generations, the subpopulations are merged, and will be splitted again to perform parallel generations. This will be repeated for N_{GFE} gene flow events. Afterwards, the best individual out of the entire population is selected, and returned.

1.3.1 A population of parameter sets

In the problem of optimizing the k_{cat} values for the selected enzymes, each individual in the population represents a list of k_{cat} values related specific enzyme-reaction associations. An individual thus represents a list of numeric values, and contains information about the minimal and maximal possible k_{cat} values, the associated enzymes, reactions, and directionality of the catalysis. The number of selected enzymes N_E to optimize, is user-defined and indirectly relates to the

size of the individual. As a single enzyme can catalyze multiple reactions as well in the forward, as reverse direction, the size of the individual depends on the number of k_{cat}s associated with the selected enzymes. A population is simply a list of individuals, with length N_p.

The initial state of an individual is based on the k_{cat} values in the model. When the population is initiated, the list of k_{cat} values are mutated with a user determined mutation probability ($\mathrm{P_m}$) and the fitness of the mutated individuals is reassessed.

1.3.2 Description of a generation: elitism, selection, crossover and mutation

During a single generation of the genetic algorithm, the individuals in the population are splitted in subpopulations with size N_{sp} and subjected to elitism, selection, crossover, and mutation, aiming to sample the solution space in an effective manner.

In order to only maintain the best solutions, the fittest individuals of the subpopulation are selected during elitism and selection. First, the best individual (elite) is extracted from the subpopulation and protected from further modification. The remaining $N_{sp} - 1$ individuals are subjected to selection for breeding the subsequent population, referred to as offspring. During selection, three individuals are randomly drawn from the population. Each of the drawn individuals is replaced by the best of the three. This procedure is repeated until $N_{sp} - 1$ individuals have been drawn from the population.

The offspring is then altered by crossover and mutation. During crossover, information from the k_{cat} lists of two subsequent individuals in a population is exchanged, based on a user-defined crossover probability P_{co}. The crossover is performed by randomly selecting two points on the k_{cat} list, to mark the beginning and end of the list to be exchanged. After crossover, the individuals in the offspring are mutated based on the mutation probability $\mathrm{P_m}$. For each k_{cat} value to mutate, a new value is drawn from an uniform distribution from zero to twice the k_{cat} value, bounded by the diffusion limit of $10^6 \mathrm{s}^{-1}$ or user-determined upper and lower bounds on the k_{cat} value.

After all evolutionary operations, a new population is created from the offspring and the elite, and the fitness of all modified individuals is determined. The new population is passed on to the next generation until the maximal number of generations is reached.

1.3.3 Gene flow events and parallel computing

Each population undergoes several generations, after which the populations are combined and reshuffled. Each subpopulation has equal population size N_{sp}, which together sum up to the total population size N_p. Similar to the exchange genetics between independently evolved populations in classical genetics, we refer to each round of reshuffling and propagation with new populations as a 'gene flow event'. To make optimally use of the computational resources

available, each flow of generations within a gene flow event can be run on parallel threads. After N_{GFE} gene flow events, the final population is determined and the genetic algorithm is terminated.

Table S2: Overview of the main hyperparameters in the genetic algorithm. All hyperparameters can be found in the Github repository in the *HyperParameters* dataclass under Modules PAM_parametrizer pam_data_classes.py.

Parameter	**Default value**	**Description**
N_p	10	Population size
N_{sp}	5	Subpopulation size
N_E	5	Number of enzymes to adapt
N_G	10	Number of generations
N_{GFE}	2	Number of gene flow events
P_m	0.9	Mutation probability
P_{co}	0.8	Crossover probability

1.3.4 Optimizing the unused enzyme sector at zero growth

After parametrizing the active enzyme sector, the slope of the unused enzyme sector is optimized by changing the amount of un- and underused enzymes at zero growth. The amount of unused enzymes at zero growth can be determined using protein overexpression experiments (1.1.1), but the amount of underused enzymes is unknown [19]. This parameter is associated with the quantitative effect the metabolic enzymes have on the phenotype by determining the increase of the fraction for active enzymes with increasing growth rate, but does not affect the qualitative relation between the turnover numbers of the individual enzymes. Thus after optimizing the k_{cat} distribution, the slope of the unused enzyme sector is optimized by minimizing the simulation error to experimental data points. The optimization is performed using the local minimization of scalar function of one variable as implemented in SciPy (acceptable error in solution, *xatol*, set to 10^{-3}) [20], using the original intercept as a lower bound and the fraction of un- and underused enzymes determined by O'Brien et al. (2016) [19] ($0.67 \mathrm{g_{UE}}/\mathrm{g_p}$) as upper bound (Table S3).

1.3.5 Software implementation

The entire parametrization framework, including the genetic algorithm, are available in Python at `https://github.com/iAMB-RWTH-Aachen/PAMParametrizer`. The algorithm is based on the DEAP toolbox [21], amongst others. Table S3 gives an overview of the dependencies and specific functions. The framework was run in Python 3.9 with DEAP v.1.4.1y and SciPy v.....

Table S3: Software dependencies of the PAMparametrizer toolbox

Functionality	Package	Function	Settings	Reference
Genetic Algorithm				
Creating a population of individuals	DEAP.tools	initRepeat	3 individuals k=$N_{sp}-1$	[21]
Selection	DEAP.tools	selBest		[21]
Crossover	DEAP.tools	cxTwoPoint		[21]
Cloning	DEAP.base	clone		[21]
PAMParametrizer				
Optimization of UES at zero growth	SciPy.optimize	minimize_scalar	bounds=(0.37,0.67) $g_{UES}/g_p \cdot g_p/g_{CDW}$ method='bounded' options={'xatol': 1e-3}	[19, 20]

1.4 Comparing quantitative proteomics measurements to simulated enzyme concentrations

Comparing simulated enzyme concentrations to quantitative proteomics data requires harmonizing units and resolving gene-protein-reaction associations. Experimental measurements by Schmidt et al. (2016) include quantitative proteomics measurements at the peptide level in fg/cell, growth rates in 1/h, and cell volume in fl/cell [2]. The PAM predicts protein concentrations in mmol/g_{CDW}/h. These protein concentrations are valid for a single catalytically active protein, which can catalyze multiple reactions and can be associated to multiple peptides. Furthermore, the protein concentration allocated by the PAM is considered to be fully utilized for metabolic conversions, all protein mass not utilized is summarized in the unused enzymes sector. To compare the predicted and measured enzyme concentration, the units of the measurements and simulations were converted to $g_{protein,i}/\sum g_{protein}$ for each reaction i, where $g_{protein,i}$ denotes the sum of concentrations of all proteins and peptides associated to reaction i. $\sum g_{protein}$ also includes the unused protein fraction.

1.4.1 Mapping peptide concentrations to catalytic units

The experimental measurements report concentrations for individual peptides, but the PAM returns concentrations of enzyme units, which can consist out of one or multiple enzymes and/or enzyme complexes. Therefore, we calculate the enzyme concentrations and molar masses as follows:

- **Homomeric enzymes** retain their original concentrations and mass
- **Enzyme complexes** take the minimum concentration and the sum of molar masses of all participating peptides

- **Isozymes** take the sum concentration of all isozymes and retain their original mass

As the PAM already includes enzyme complexes, the only modification to simulated enzyme concentration was summation of the enzyme concentrations associated to each reaction (isozyme).

1.4.2 Unit conversions

The quantitative proteomics measurements were adjusted using information about the cell volume of the cultured cells and an estimate on the amount of $\mathrm{g_{CDW}}$ per volume of cells from Bakken and Olsen (1983) [22].

$$\rho_{\mathrm{wet}} = 1.09 \qquad (\mathrm{g_{wet}/cm^3}) \tag{9}$$

$$f_{\mathrm{dry}} = 0.3 \qquad (\mathrm{g_{wet}/g_{dry}}) \tag{10}$$

$$\rho_{\mathrm{dry}} = \rho_{\mathrm{wet}} \times 10^3 \times f_{\mathrm{dry}} \qquad (\mathrm{g_{dry}/L}) \tag{11}$$

$$p_i = \frac{\mathrm{m}_{p,i}}{V_{\mathrm{cell}} \cdot \rho_{\mathrm{dry}}} (\mathrm{g_{protein},}_i/\mathrm{g_{dry}}) \tag{12}$$

$$\mathrm{f_{protein,i}} = \frac{p_i}{\sum \mathrm{g_{protein}}} (\mathrm{g_{protein},}_i/\mathrm{g_{protein}}) \tag{13}$$

$$\tag{14}$$

Where ρ_{wet} is amount of wet weight of cells per $\mathrm{cm^3}$, f_{dry} the fraction of dry weight per wet weight of cells, ρ_{dry} the amount of dry weight per liter of cells, p_i is the amount of protein i in $\mathrm{g_{protein},}_i/\mathrm{g_{CDW}}$, $\mathrm{m}_{p,i}$ is the measured protein mass in fg/cell, and V_{cell} is the volume in fl of a single cell. The protein fractions $\mathrm{f_{protein,i}}$ in $\mathrm{g_{protein},}_i/\mathrm{g_{protein}}$ are subsequently calculated by dividing each protein fraction p_i by the sum of all protein fractions.

The PAM predicts enzyme concentrations in $\mathrm{mmol/g_{CDW}}$, and only covers the enzyme mass which is fully metabolically active. The units thus need to be converted to the mass of enzymes per total mass of all metabolically active and unused metabolic enzymes.

$$p_i = \epsilon_i \cdot \mathrm{molmass}_i \cdot 10^{-3} (\mathrm{g_{protein},}_i/\mathrm{g_{dry}}) \tag{15}$$

$$\mathrm{f_{protein,i}} = \frac{p_i}{\sum \mathrm{g_{protein}} + w_{UE} \cdot \mu + \phi_{\mathrm{UE,0}}} (\mathrm{g_{protein},}_i/\mathrm{g_{protein}}) \tag{16}$$

$$\tag{17}$$

where ϵ_i is the simulated enzyme concentration of enzyme i in $\mathrm{mmol/g_{CDW}}$, $\mathrm{molmass}_i$ is the molar mass of enzyme i in g/mol, 10^{-3} converts mmol to mol, and w_{UE} and $\phi_{\mathrm{UE,0}}$ are the slope and intercept, respectively, of the relationship between unused enzymes and the growth rate μ in 1/h.

2 Supplementary Discussion

2.1 Description of the updated Protein Allocation Model structure

2.1.1 Translational protein sector

Using experimental data from Schmidt et al. (2016) [2], we find that translational protein abundance does not exhibit a strictly linear correlation with growth rate across the full range of physiologically plausible growth rates (Supplementary Figure S3), despite previous studies suggesting such a relationship [2, 9, 23–26]. The linear correlation seems to cease when *E. coli* enters the overflow metabolic regime. This indicates that when the microorganism needs to consume more substrate to establish an increase in growth rate, also the requirement of translational proteins rises. Furthermore, there exists an indirect correlation between the glucose abundance in *E. coli* to the translational protein fraction by CRP-cAMP [3, 27, 28] and ppGpp [4, 5, 29]. Increased carbon uptake activates CRP-cAMP signaling, leading to a downregulation of catabolic protein expression while promoting ribosome and anabolic protein synthesis, driven by the accumulation of biomass precursors [3]. Under nutrient limitation, synthesis of ppGpp is triggered, which is an inhibitor of rRNA and tRNA synthesis [5, 29, 30]. Both observations suggest that the translational protein sector is not related to the growth rate, but to the substrate uptake rate. Indeed, when we establish a relationship between glucose uptake rate and translational sector, the model captures the experimentally observed fraction of translational proteins (Supplementary Figure S3).

2.1.2 Unused protein sector

To correct for un- and underused enzymes, the PAM includes an unused enzymes sector, based on the findings of O'Brien and colleagues (2015) [1, 6]. This sector enables us to assume all the enzymes in the metabolic sector are operating at their maximal *in vivo* catalytic efficiency under biologically plausible conditions. In the original PAM, the fraction of unused proteins is assumed to be zero at the maximal growth rate. However, emerging evidence suggests that even at maximum growth rates, a fraction of unused proteins persists, likely due to bet-hedging strategies to adapt to fluctuating conditions in ecological niches [7, 31, 32]. Even if all the protein capacity is optimally used at the maximal growth rate, no higher growth rates can be achieved after adaptive laboratory evolution, unless the mutations enzymes improve the catalytic efficiency of these enzymes. Analysis of adapted strains showed that part of the proteome fraction for housekeeping can be allocated to metabolic or translational proteins after multiple generations [8, 10]. This might be the result of mutations in proteins such as RNA polymerase and *PyFk*, which direct the transcriptional regulation towards condition-specific proteins [8, 10]. Furthermore, there are suggestions that after ALE experiments, all proteins operate near their *in vivo* maximum [10]. In our definition, this part of the proteome falls into the unused protein

sector, as it can be allocated to growth. Assuming that after ALE, the protein allocation is optimal, e.g. there is no more protein space left to invest in growth, we can assume that at the maximal growth rate of the evolved strains, the unused protein sector is zero.

3 Supplementary Tables

Table S4: Statistics for the k_{cat} (Figure 2B of the main text) and flux (Supplementary Figure S5) distributions for protein allocation models parametrized with different parameter sets, as obtained from GotEzymes [33], after preprocessing, or from different alternative parametrizations. The flux distribution was obtained from simulations with the different models with upper bound of glucose uptake set to 11 mmol/g_{CDW}/h. AUC: Area under the curve.

Model	**k_{cat} distributions**			**Flux distribution**			
	Median	Mean	AUC	Median	Mean	AUC	Growth rate
	1/s			mmol/g_{CDW}/h			1/h
iML1515			0.09	2.03	$3.12 \cdot 10^4$	0.97	
GotEnzymes	13.7	296	$2.6 \cdot 10^5$	0.01	0.85	$7.44 \cdot 10^3$	0.11
After preprocessing	95.9	$2.07 \cdot 10^3$	$1.9 \cdot 10^6$	0.10	2.10	$2.67 \cdot 10^4$	0.77
Alternative 1	95.9	$2.07 \cdot 10^3$	$1.8 \cdot 10^6$	0.07	1.56	$1.65 \cdot 10^4$ 2	0.56
Alternative 2	95.9	$2.07 \cdot 10^3$	$1.9 \cdot 10^6$	0.08	1.86	$1.98 \cdot 10^4$	0.64
Alternative 3	95.9	$2.07 \cdot 10^3$	$1.9 \cdot 10^6$	0.08	1.78	$2.03 \cdot 10^4$	0.63
Alternative 4	95.9	$2.07 \cdot 10^3$	$1.9 \cdot 10^6$	0.08	2.02	$2.09 \cdot 10^4$	0.62
Alternative 5	95.9	$2.07 \cdot 10^3$	$1.9 \cdot 10^6$	0.07	1.73	$2.22 \cdot 10^4$	0.64
Alternative 6	95.9	$2.07 \cdot 10^3$	$1.9 \cdot 10^6$	0.078	1.54	$3.56 \cdot 10^4$	0.61
Alternative 7	95.9	$2.07 \cdot 10^3$	$1.9 \cdot 10^6$	0.08	1.80	$1.87 \cdot 10^4$	0.61
Alternative 8	95.9	$2.07 \cdot 10^3$	$1.9 \cdot 10^6$	0.08	2.10	$2.38 \cdot 10^4$	0.66
Alternative 9	95.9	$2.07 \cdot 10^3$	$1.9 \cdot 10^6$	0.08	2.15	$2.35 \cdot 10^4$	0.65
Alternative 10	95.9	$2.07 \cdot 10^3$	$1.9 \cdot 10^6$	0.08	1.63	$1.83 \cdot 10^4$	0.66

Table S5: Summary statistics for the absolute difference between simulated fluxes and measured fluxes for different parameter sets of the *Escherichia coli* Protein Allocation Model (PAM). Flux comparison was performed for five different dilution rates for growth on glucose (experimental data from Nanchen et al. (2006) and Rijsewijk et al. (2011) [34,35]) and eight different carbon sources (experimental measurements from Gerosa et al. (2015) [36]). All differences are given in mmol/g_{CDW}/h. Stdev: standard deviation to the mean difference.

	Glucose			**Multiple carbon sources**		
Model	**Median**	**Mean**	**Stdev**	**Median**	**Mean**	**Stdev**
iML1515	0.41	0.32	0.99	0.00	0.15	6.69
GotEnzymes	0.46	0.21	2.20	0.00	0.83	5.01
Curated	0.41	0.33	1.02	0.47	1.73	4.26
Alternative 1	0.39	0.23	1.68	0.18	1.51	4.18
Alternative 2	0.39	0.24	0.93	0.13	1.45	4.25
Alternative 3	0.44	0.74	1.42	0.18	1.52	4.20
Alternative 4	0.42	0.50	0.95	0.18	1.51	4.21
Alternative 5	0.42	0.50	0.94	0.18	1.48	4.22
Alternative 6	0.39	0.27	0.98	0.18	1.43	4.24
Alternative 7	0.43	0.73	1.47	0.18	1.50	4.19
Alternative 8	0.42	0.73	1.42	0.18	1.55	4.18
Alternative 9	0.42	0.65	1.33	0.18	1.52	4.22
Alternative 10	0.41	0.66	1.36	0.18	1.51	4.19

Table S6: Reaction-protein mapping used to map the most sensitive enzymes in Figure 2B of the main text to reaction identifiers. Each enzyme is represented by a single reaction (in bold), although multiple reactions can be associated with the same enzyme. Each enzyme-reaction association group is separated by lines. Reaction identifiers are from the BiGG model namespace [37].

Model enzyme	**BiGG reaction identifier**	**Reaction description**
P0A6E6_P0AB98_P0ABA0_P0ABA4 _P0ABA6_P0ABB0_P0ABB4_P0ABC0 _P68699	**ATPS4rpp**	ATP synthase
P0AFC3_P0AFC7_P0AFD1_P0AFD4 _P0AFD6_P0AFE0_P0AFE4_P0AFE8 _P0AFF0_P31979_P33599_P33602 _P33607	**NADH16pp-18pp**	NADH dehydrogenase
P0ABI8_P0ABJ1_P0ABJ3_P0ABJ6	**CYTBO3_4pp**	Cytochrome oxidase bo3
P0A6P9	**ENO**	Enolase
P06959_P0A9P0_P0AFG8	**PDH**	Pyruvate dehydrogenase
P0AB80	**ILETA**	Isoleucine transaminase
P0AB80	LEUTAi	Leucine transaminase (irreversible)
P0AB80	PHETA1	Phenylalanine transaminase
P0AB80	TYRTA	Tyrosine transaminase
P0AB80	VALTA	Valine transaminase
P0AGB0	PSP_L	Phosphoserine phosphatase (L-serine)
P15640	PRAGSr	Phosphoribosylglycinamide synthase
P00509	ASPTA	Aspartate transaminase
P00509	CYSTA	Cysteine transaminase
P00509	PHETA1	Phenylalanine transaminas
P00509	TYRTA	Tyrosine transaminase

Table S7: Description of abbreviations used to indicate reaction fluxes in central carbon metabolism in Figure 5A and 5B of the main text and Supplementary Figure S8. Reaction identifiers are from the BiGG model namespace [37].

BiGG reaction identifier	**Reaction description**
EX_gly_e	Glycerol exchange
EX_tre_e	Trehalose exchange
GLCt2pp	D-glucose transport in via proton symport (periplasm)
GLCDpp	Glucose dehydrogenase (ubiquinone-8 as acceptor) (periplasm)
GAD2ktpp	Gluconate 2 dehydrogenase periplasm
GNK	Gluconokinase
2DHGLCK	Dehydrogluconokinase
PGLCNDH	Phosphogluconate 2 dehydrogenase
HEX1	Hexokinase (D-glucose:ATP)
PGI	Glucose-6-phosphate isomerase
PFK	Phosphofructokinase
FBA	Fructose-bisphosphate aldolase
FBP	Fructose-bisphosphatase
TPI	Triose-phosphate isomerase
GAPD	Glyceraldehyde-3-phosphate dehydrogenase
PGK	Phosphoglycerate kinase
PGM	Phosphoglycerate mutase
ENO	Enolase
PYK	Pyruvate kinase
EDD	6-phosphogluconate dehydratase
EDA	2-dehydro-3-deoxy-phosphogluconate aldolase
TKT1	Transketolase
PGL	6-phosphogluconolactonase
GND	Phosphogluconate dehydrogenase
RPI	Ribose-5-phosphate isomerase
RPE	Ribulose 5-phosphate 3-epimerase
TALA	Transaldolase
PDH	Pyruvate dehydrogenase
CS	Citrate synthase
PC	Pyruvate carboxylase
OAADC	Oxaloacetate decarboxylase
ACONTa/ACONTb	Aconitase
ICDHyr	Isocitrate dehydrogenase (NADP)
SUCOAS	Succinyl-CoA synthetase (ADP-forming)
SUCDi	Succinate dehydrogenase (irreversible)
FUM	Fumarase
MDH	Malate dehydrogenase
ME2	Malic enzyme (NADP)
PPC	Phosphoenolpyruvate carboxylase
PPS	Phosphoenolpyruvate synthase
PPCK	Phosphoenolpyruvate carboxykinase
F6PA	Fructose 6-phosphate aldolase
DHAPT	Dihydroxyacetone phosphotransferase
PFK_3	Phosphofructokinase (s7p)
FBA3	Sedoheptulose 1,7-bisphosphate D-glyceraldehyde-3-phosphate-lyase

Table S8: Parametrization of the coarse-grained protein sectors for the *Pseudomonas putida* and *Corynebacterium glutamicum* Protein Allocation Models.

Protein Sector	**substrate**	**Slope**	**Intercept**	**Reference**
Pseudomonas putida				
Translational Proteins	Glucose	$-4.73 \cdot 10^{-3}$	$4.72 \cdot 10^{-2}$	
	Glycerol	$-2.91 \cdot 10^{-17}$	$4.72 \cdot 10^{-2}$	
	Succinate	$-2.56 \cdot 10^{-3}$	$4.721 \cdot 10^{-2}$	This study
	Benzoate	$-4.20 \cdot 10^{-3}$	$4.72 \cdot 10^{-2}$	
Unused Enzymes	Glucose	$1.60 \cdot 10^{-2}$	0.20	
	Glycerol	$0.98 \cdot 10^{-2}$	0.20	This study
	Succinate	$0.87 \cdot 10^{-2}$	0.20	
	Benzoate	$1.42 \cdot 10^{-3}$	$4.61 \cdot 10^{-2}$	
Corynebacterium glutamicum				
Translational Proteins	Glucose	$-2.09 \cdot 10^{-3}$	$7.39 \cdot 10^{-2}$	[38]
	Gluconate	$-6.26 \cdot 10^{-3}$	$5.56 \cdot 10^{-2}$	
	Succinate	$-3.66 \cdot 10^{-3}$	$5.56 \cdot 10^{-2}$	This study
	Fructose	$-6.92 \cdot 10^{-3}$	$5.56 \cdot 10^{-2}$	
Unused Enzymes	Glucose	$2.12 \cdot 10^{-2}$	0.20	[38, 39]
	Gluconate	$1.91 \cdot 10^{-2}$	0.20	
	Succinate	$1.11 \cdot 10^{-2}$	0.20	This study
	Fructose	$2.11 \cdot 10^{-2}$	0.20	

4 Supplementary Figures

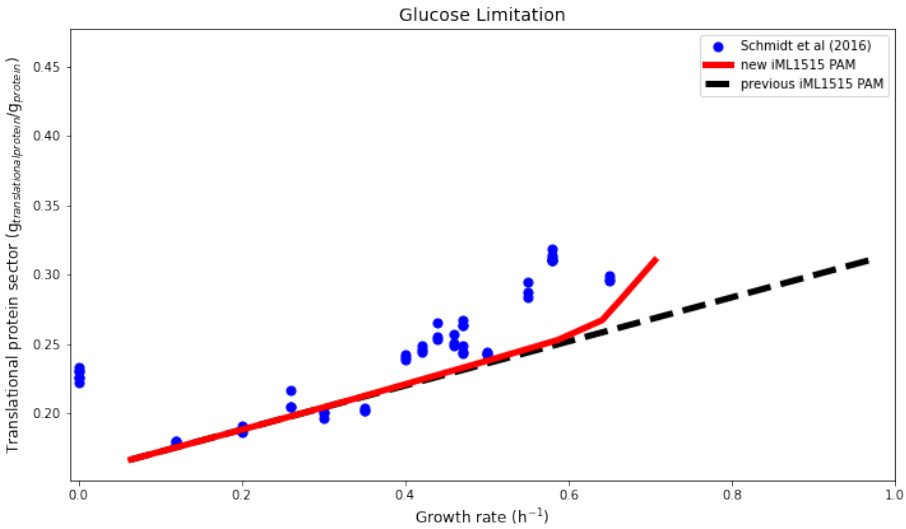

Figure S3: Translational proteins as function of the biomass production rate. Quantitative proteomics measurements from Schmidt et al (2016) [2] (points) do not seem to be linearly related to the growth rate. The previous iML1515 PAM (the protein allocation model (PAM) as published by Alter et al. (2021) [1]) includes a linear relation between the translational sector and the biomass production rate. In the new iML1515 PAM, the translational sector is linearly related to the substrate uptake rate, which results in a non-linear relation to biomass formation.

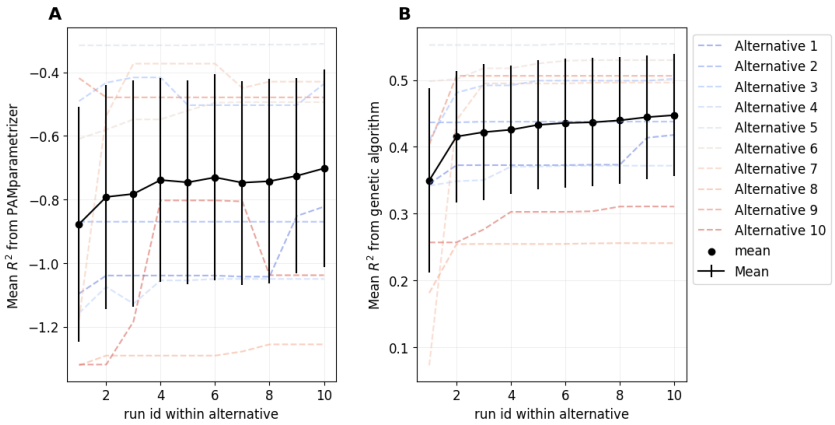

Figure S4: R^2 value as function of different iterations of the PAMparametrizer for ten different parametrizations of the *Escherichia coli* Protein Allocation Model. Error bars indicate the standard deviation from the mean error of the different models. (A) R^2 calculated by the PAMparametrizer without weights, (B) R^2 calculated by the genetic algorithm, with a weight of 7 on the growth rate and a weight of 3 on the acetate secretion rate (see methods),

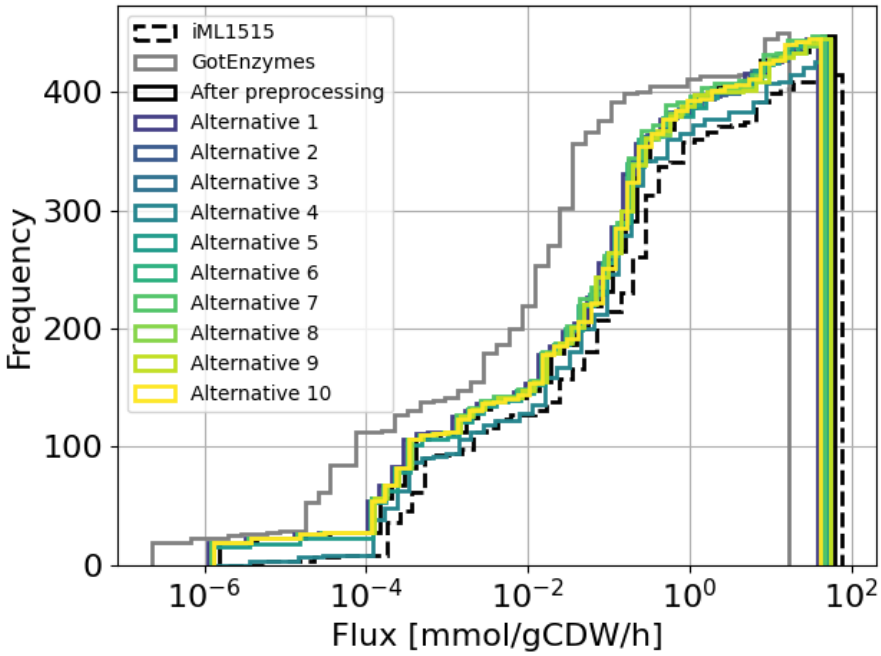

Figure S5: Cumulative distribution of the fluxes simulated by Protein Allocation Models for *Escherichia coli* with different parameter sets: one from the GotEnzymes database [14], one after preprocessing, and ten alternative parameter sets resulting from the PAMParametrizer.

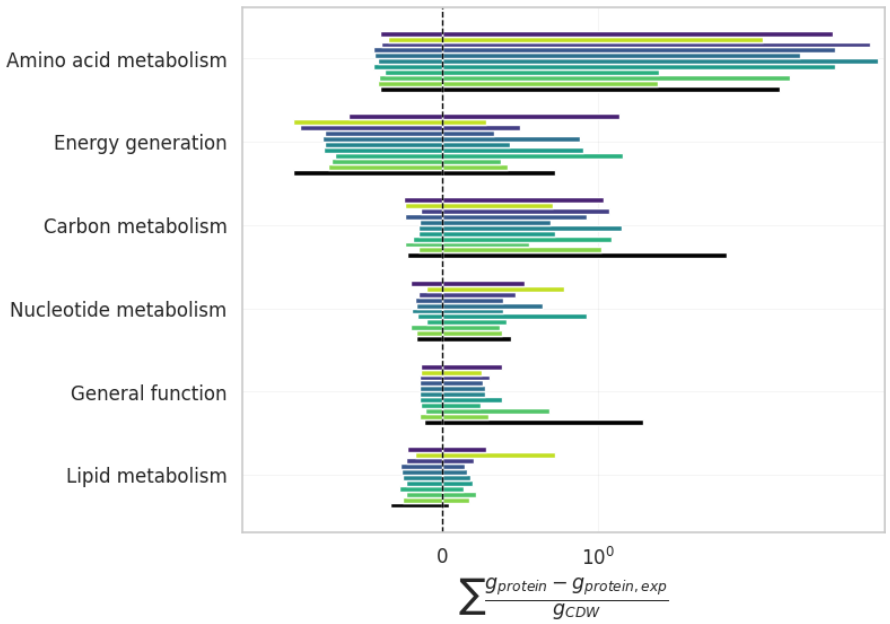

Figure S6: Difference between simulated protein concentrations to experimental measurements per clusters of orthologous genes (COGs). Only the COGs with a total difference of 10 $g_{protein}/g_{CDW}$ are shown. The annotation 'metabolism', also includes transport functionality. Experimental measurements were taken from Schmidt et al. (2016) at the following growth rates: 0.35, 0.5 and 0.67 h^{-1} [2].

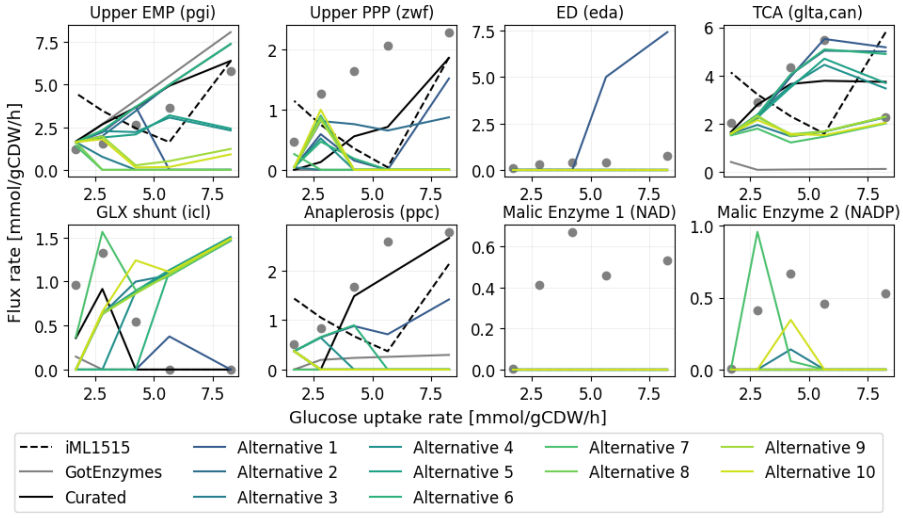

Figure S7: Simulated intracellular fluxes for as a function of the glucose uptake rate for ten different parametrizations of the *Escherichia coli* Protein Allocation Model. Points indicate experimental measurements from Nanchen et al. (2006) and Rijsewijk et al. (2011) [34, 35]. All flux rates are given in mmol/g_{CDW}/h

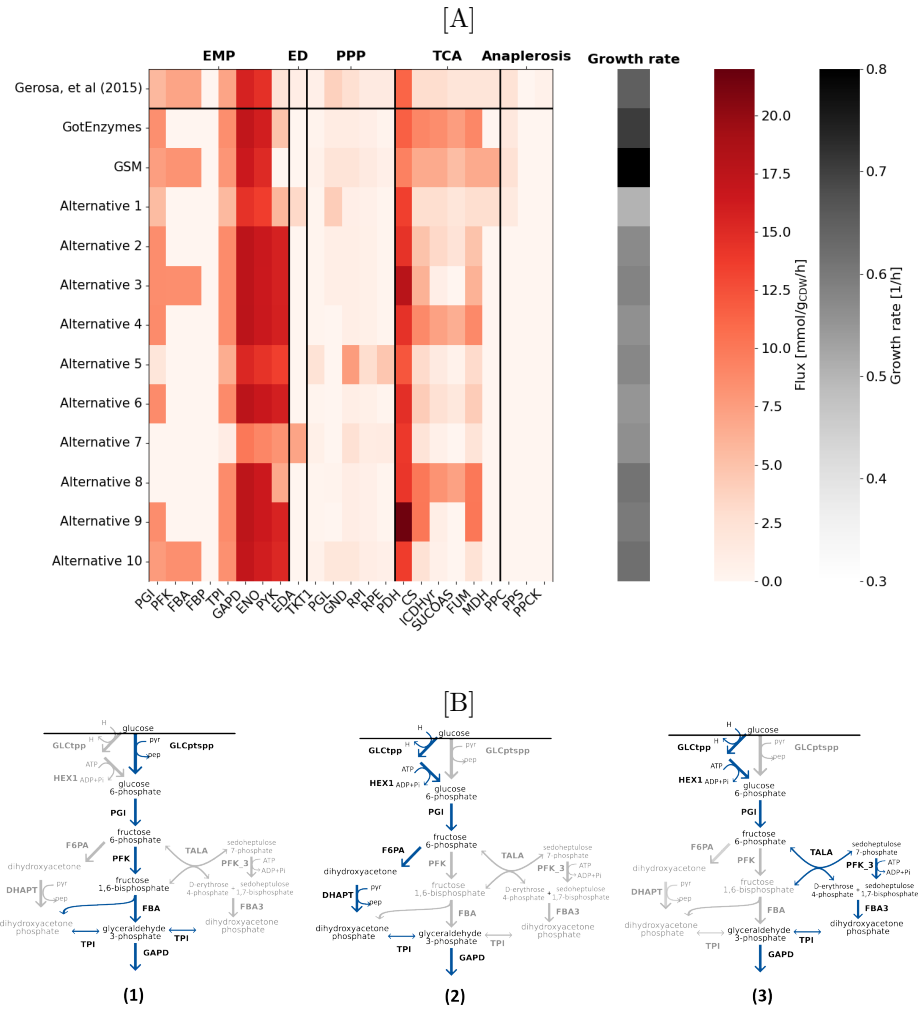

Figure S8: Glucose catabolism in *Escherichia coli* (*E. coli*) as predicted by genome-scale models (GEMs) and Protein Allocation Models (PAMs). (A) Simulated intracellular fluxes by *E. coli* PAMs created using parameters form GotEnzymes and ten alternatives resulting from parametrization at a glucose uptake rate of 9.65 mmol/g_{CDW}/h compared to experimental 13C flux measurement from Gerosa. (2015) [36]. (B) Stoichiometrically equivalent routes for glucose catabolization in *E. coli*. Abbreviations of the reaction identifiers are specified in Supplementary Table S7

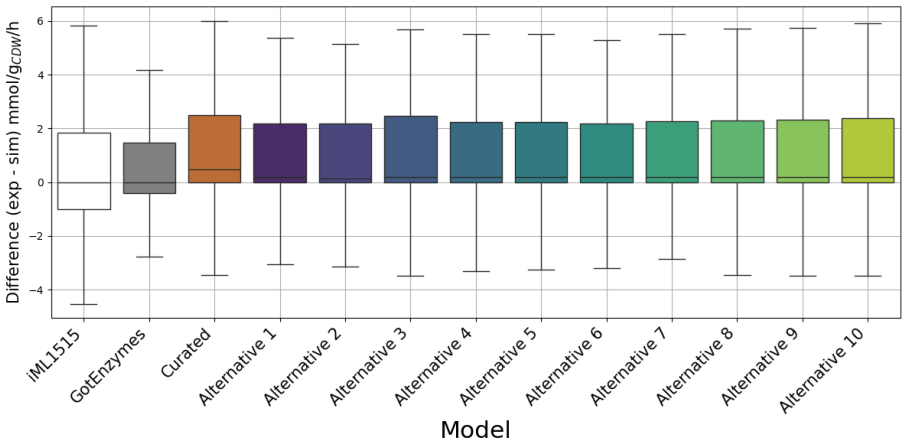

Figure S9: Difference between the measured intracellular fluxesfor growth on seven different carbon sources from Gerosa et al. (2015) [36] and the corresponding simulated fluxes of different *Escherichia coli* Protein Allocation Models, including a manually curated PAM from Alter et al. (2021) [1].

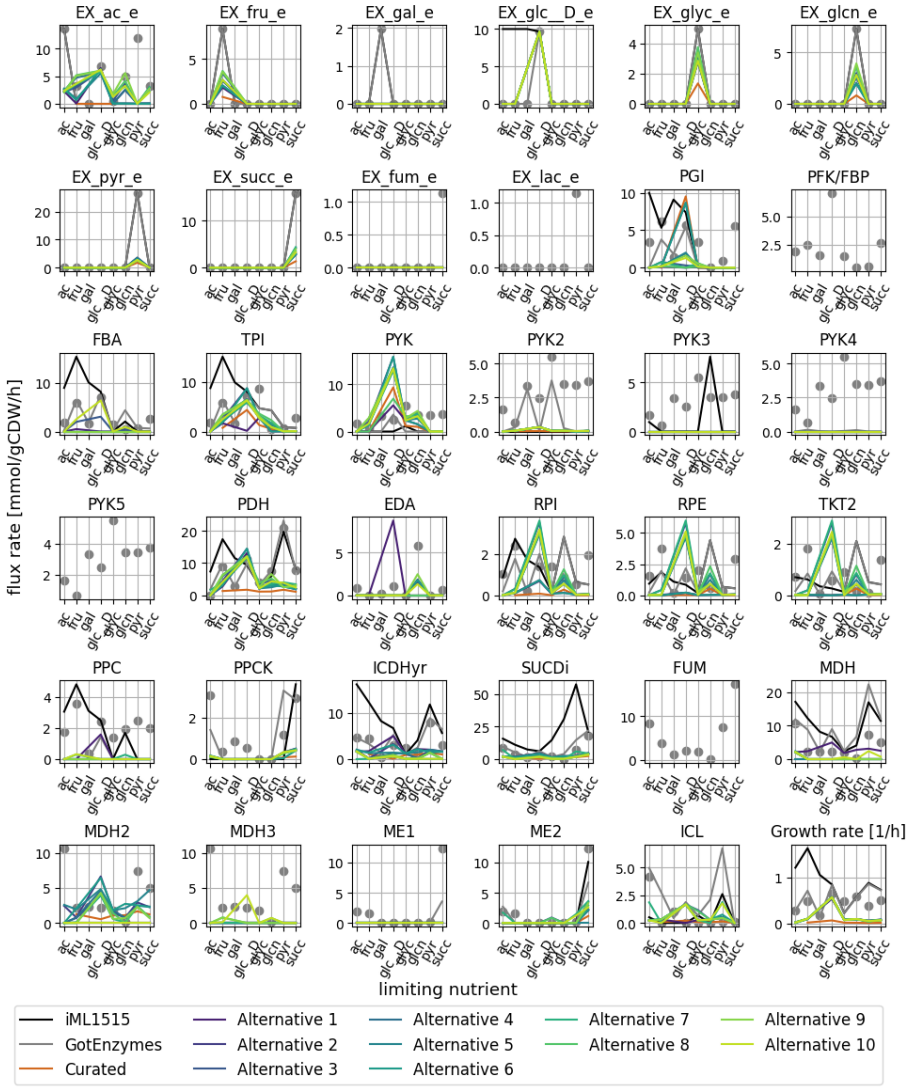

Figure S10: Simulated intracellular fluxes for growth on seven different carbon sources for ten different parametrizations of the *E. coli* Protein Allocation Model. Points indicate experimental measurements from Gerosa et al. (2015) [36]. All flux rates are given in mmol/g_{CDW}/h.

References

[1] T. B. Alter, L. M. Blank, and B. E. Ebert, "Proteome regulation patterns determine *Escherichia coli* wild-type and mutant phenotypes," *mSystems*, vol. 6, pp. e00625–20, Apr. 2021.

[2] A. Schmidt, K. Kochanowski, S. Vedelaar, E. Ahrné, B. Volkmer, L. Callipo, K. Knoops, M. Bauer, R. Aebersold, and M. Heinemann, "The quantitative and condition-dependent *Escherichia coli* proteome," *Nature Biotechnology*, vol. 34, pp. 104–110, Jan. 2016.

[3] C. You, H. Okano, S. Hui, Z. Zhang, M. Kim, C. W. Gunderson, Y.-P. Wang, P. Lenz, D. Yan, and T. Hwa, "Coordination of bacterial proteome with metabolism by cyclic AMP signalling," *Nature*, vol. 500, pp. 301–306, Aug. 2013.

[4] B. W. Anderson, D. K. Fung, and J. D. Wang, "Regulatory themes and variations by the stress-signaling nucleotide alarmones (p)ppGpp in bacteria," *Annual Review of Genetics*, vol. 55, pp. 115–133, Nov. 2021.

[5] R. Droghetti, P. Fuchs, I. Iuliani, V. Firmano, G. Tallarico, L. Calabrese, J. Grilli, B. Sclavi, L. Ciandrini, and M. Cosentino Lagomarsino, "Incoherent feedback from coupled amino acids and ribosome pools generates damped oscillations in growing *E. coli*," *Nature Communications*, vol. 16, p. 3063, Mar. 2025.

[6] E. J. O'Brien, J. Utrilla, and B. O. Palsson, "Quantification and classification of *E. coli* proteome utilization and unused protein costs across environments," *PLOS Computational Biology*, vol. 12, p. e1004998, June 2016.

[7] F. J. Bruggeman, B. Teusink, and R. Steuer, "Trade-offs between the instantaneous growth rate and long-term fitness: Consequences for microbial physiology and predictive computational models," *BioEssays*, vol. 45, no. 10, p. 2300015, 2023.

[8] J. Utrilla, E. O'Brien, K. Chen, D. McCloskey, J. Cheung, H. Wang, D. Armenta-Medina, A. Feist, and B. Palsson, "Global rebalancing of cellular resources by pleiotropic point mutations illustrates a multi-scale mechanism of adaptive evolution," *Cell Systems*, vol. 2, pp. 260–271, Apr. 2016.

[9] F. J. Bruggeman, R. Planqué, D. Molenaar, and B. Teusink, "Searching for principles of microbial physiology," *FEMS Microbiology Reviews*, vol. 44, pp. 821–844, Nov. 2020.

[10] M. Mori, V. Patsalo, C. Euler, J. R. Williamson, and M. Scott, "Proteome partitioning constraints in long-term laboratory evolution," *Nature Communications*, vol. 15, p. 4087, May 2024.

[11] D. Machado, M. J. Herrgård, and I. Rocha, "Stoichiometric representation of gene–protein–reaction associations leverages constraint-based analysis from reaction to gene-level phenotype prediction," *PLoS Computational Biology*, vol. 12, p. e1005140, Oct. 2016.

[12] M. S. Bienick, K. W. Young, J. R. Klesmith, E. E. Detwiler, K. J. Tomek, and T. A. Whitehead, "The interrelationship between promoter strength, gene expression, and growth rate," *PLOS ONE*, vol. 9, p. e109105, Oct. 2014.

[13] M. Kafri, E. Metzl-Raz, G. Jona, and N. Barkai, "The cost of protein production," *Cell Reports*, vol. 14, pp. 22–31, Dec. 2015.

[14] F. Li, Y. Chen, M. Anton, and J. Nielsen, "GotEnzymes: an extensive database of enzyme parameter predictions," *Nucleic Acids Research*, vol. 51, pp. D583–D586, Jan. 2023.

[15] The UniProt Consortium, "UniProt: the Universal Protein Knowledgebase in 2023," *Nucleic Acids Research*, vol. 51, pp. D523–D531, Jan. 2023.

[16] S. van den Bogaard, P. A. Saa, and T. B. Alter, "Sensitivities in protein allocation models reveal distribution of metabolic capacity and flux control," *Bioinformatics*, p. btae691, Nov. 2024.

[17] M. Srinivas and L. Patnaik, "Adaptive probabilities of crossover and mutation in genetic algorithms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 24, pp. 656–667, Apr. 1994.

[18] T. B. Alter, L. M. Blank, and B. E. Ebert, "Genetic optimization algorithm for metabolic engineering revisited," *Metabolites*, vol. 8, p. 33, May 2018.

[19] E. J. O'Brien, J. M. Monk, and B. O. Palsson, "Using Genome-scale Models to Predict Biological Capabilities," *Cell*, vol. 161, pp. 971–987, May 2015.

[20] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, İ. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and SciPy 1.0 Contributors, "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python," *Nature Methods*, vol. 17, pp. 261–272, 2020.

[21] F.-A. Fortin, "DEAP: Evolutionary Algorithms Made Easy," *Journal of Machine Learning Research*, pp. 2171–2175, July 2012.

[22] L. R. Bakken and R. A. Olsen, "Buoyant densities and dry-matter contents of microorganisms: Conversion of a measured biovolume into biomass," *Applied and Environmental Microbiology*, vol. 45, pp. 1188–1195, Apr. 1983.

[23] M. Scott, C. W. Gunderson, E. M. Mateescu, Z. Zhang, and T. Hwa, "Interdependence of cell growth and gene expression: Origins and consequences," *Science*, vol. 330, pp. 1099–1102, Nov. 2010.

[24] K. Peebo, K. Valgepea, A. Maser, R. Nahku, K. Adamberg, and R. Vilu, "Proteome reallocation in *Escherichia coli* with increasing specific growth rate," *Molecular BioSystems*, vol. 11, pp. 1184–1193, Mar. 2015.

[25] S. Hui, J. M. Silverman, S. S. Chen, D. W. Erickson, M. Basan, J. Wang, T. Hwa, and J. R. Williamson, "Quantitative proteomic analysis reveals a simple strategy of global resource allocation in bacteria," *Molecular Systems Biology*, vol. 11, p. 784, Feb. 2015.

[26] X. Dai, M. Zhu, M. Warren, R. Balakrishnan, V. Patsalo, H. Okano, J. R. Williamson, K. Fredrick, Y.-P. Wang, and T. Hwa, "Reduction of translating ribosomes enables *Escherichia coli* to maintain elongation rates during slow growth," *Nature Microbiology*, vol. 2, pp. 1–9, Dec. 2016.

[27] B. Görke and J. Stülke, "Carbon catabolite repression in bacteria: many ways to make the most out of nutrients," *Nature Reviews Microbiology*, vol. 6, pp. 613–624, Aug. 2008.

[28] A. Kolb, S. Busby, H. Buc, S. Garges, and S. Adhya, "Transcriptional regulation by cAMP and its receptor protein," *Annual Review of Biochemistry*, vol. 62, pp. 749–797, July 1993.

[29] K. Potrykus, H. Murphy, N. Philippe, and M. Cashel, "ppGpp is the major source of growth rate control in *E. coli*," *Environmental Microbiology*, vol. 13, pp. 563–575, Mar. 2011.

[30] E. Bosdriesz, D. Molenaar, B. Teusink, and F. J. Bruggeman, "How fast-growing bacteria robustly tune their ribosome concentration to approximate growth-rate maximization," *The FEBS Journal*, vol. 282, pp. 2029–2044, May 2015.

[31] M. Basan, T. Honda, D. Christodoulou, M. Hörl, Y.-F. Chang, E. Leoncini, A. Mukherjee, H. Okano, B. R. Taylor, J. M. Silverman, C. Sanchez, J. R. Williamson, J. Paulsson, T. Hwa, and U. Sauer, "A universal trade-off between growth and lag in fluctuating environments," *Nature*, vol. 584, pp. 470–474, Aug. 2020.

[32] M. Mori, S. Schink, D. W. Erickson, U. Gerland, and T. Hwa, "Quantifying the benefit of a proteome reserve in fluctuating environments," *Nature Communications*, vol. 8, p. 1225, Oct. 2017.

[33] F. Li, L. Yuan, H. Lu, G. Li, Y. Chen, M. K. M. Engqvist, E. J. Kerkhoven, and J. Nielsen, "Deep learning-based kcat prediction enables improved enzyme-constrained model reconstruction," *Nature Catalysis*, vol. 5, pp. 662–672, June 2022.

[34] A. Nanchen, A. Schicker, and U. Sauer, "Nonlinear dependency of intracellular fluxes on growth rate in miniaturized continuous cultures of *Escherichia coli*," *Applied and Environmental Microbiology*, vol. 72, pp. 1164–1172, Feb. 2006.

[35] B. R. B. Haverkorn van Rijsewijk, A. Nanchen, S. Nallet, R. J. Kleijn, and U. Sauer, "Large-scale 13C-flux analysis reveals distinct transcriptional control of respiratory and fermentative metabolism in *Escherichia coli*," *Molecular Systems Biology*, vol. 7, p. 477, Jan. 2011.

[36] L. Gerosa, B. R. B. Haverkorn van Rijsewijk, D. Christodoulou, K. Kochanowski, T. S. B. Schmidt, E. Noor, and U. Sauer, "Pseudo-transition analysis identifies the key regulators of dynamic metabolic adaptations from steady-state data," *Cell Systems*, vol. 1, pp. 270–282, Oct. 2015.

[37] Z. A. King, J. Lu, A. Dräger, P. Miller, S. Federowicz, J. A. Lerman, A. Ebrahim, B. O. Palsson, and N. E. Lewis, "BiGG Models: A platform for integrating, standardizing and sharing genome-scale models," *Nucleic Acids Research*, vol. 44, pp. D515–D522, Jan. 2016.

[38] S. Matamouros, T. Gensch, M. Cerff, C. C. Sachs, I. Abdollahzadeh, J. Hendriks, L. Horst, N. Tenhaef, J. Tenhaef, S. Noack, M. Graf, R. Takors, K. Nöh, and M. Bott, "Growth-rate dependency of ribosome abundance and translation elongation rate in *Corynebacterium glutamicum* differs from that in *Escherichia coli*," *Nature Communications*, vol. 14, p. 5611, Sept. 2023.

[39] M. Graf, T. Haas, A. Teleki, A. Feith, M. Cerff, W. Wiechert, K. Nöh, T. Busche, J. Kalinowski, and R. Takors, "Revisiting the growth modulon of *Corynebacterium glutamicum* under glucose-limited chemostat conditions," *Frontiers in Bioengineering and Biotechnology*, vol. 8, Oct. 2020.