Cell Growth Simulator: a user-friendly web server for growth balance analysis

Sajjad Ghaffarinasab[1¶], Hugo Dourado[1¶*]

[1]Institute for Computer Science and Department of Biology, Heinrich Heine University, Universitätsstraße 1, D-40225 Düsseldorf, Germany

*Corresponding author

Email: hugo.dourado@hhu.de

The authors contributed equally to this work

Abstract

Computational models are essential for understanding the complex biochemical processes that govern cell growth. Growth Balance Analysis (GBA) offers a general mathematical framework for efficiently modeling balanced cell growth by integrating metabolism, protein synthesis requirements, dilution of all cellular components due to growth, and the nonlinear dependence of reaction rates on metabolite concentrations. The computational implementation of this framework has been however limited by demanding some level of programming expertise. Here we present Cell Growth Simulator, a web-based application that streamlines the construction and analysis of GBA models through an intuitive interface and interactive visualizations, including customizable plots and pathway maps for interpreting optimization results. The platform makes nonlinear modeling of resource allocation in coarse-grained models of growing cells accessible to a broader scientific audience, fostering interdisciplinary collaboration and advancing our understanding of cellular metabolism and growth. Cell Growth Simulator is freely available at: https://cellgrowthsim.com/.

Keywords:

Growth Balance Analysis, Optimal Resource allocation, Web application, Nonlinear Cell Model, Systems Biology

Introduction

Metabolic modeling has emerged as a crucial approach for understanding the complex biochemical processes that govern cell growth and metabolism. Computational models enable researchers to predict cellular behaviors, optimize metabolic pathways, and explore resource allocation strategies from simple principles. Linear computational approaches have been widely used to study resource allocation during balanced growth, a steady-state condition where concentrations of cellular components are constant in time. These methods typically approximate biochemical reaction kinetics by assuming linear relationships between fluxes and catalyst concentrations, neglecting the nonlinear effects of metabolite concentrations on reaction kinetics [1–3]. More sophisticated nonlinear models can explicitly account for these metabolite-dependent effects through kinetic rate laws [4]. Recently, Growth Balance Analysis (GBA) was introduced as a general framework to model and analyze such nonlinear systems, investigating how cells grow and allocate their resources under fundamental physical constraints including mass conservation, nonlinear reaction kinetics, and a limit on cellular density [5,6].

While the Growth Balance Analysis (GBA) framework simplifies and facilitates the mathematical modeling and analysis of cellular balanced growth under nonlinear constraints, its computational implementation requires mathematical and programming skills. To increase the accessibility of Growth Balance Analysis (GBA) to a wider audience, we developed Cell Growth Simulator, a user-friendly web-based platform built using R/Shiny. R is a widely used programming language and software environment for statistical computing and graphics, while Shiny is a web application framework for R that facilitates the creation of interactive and user-friendly web-based interfaces.

There are currently several web-based tools for linear metabolic modeling and visualization, including CNApy [7], Escher [8], ModelExplore [9], CAVE [10], and Fluxer [11]. These tools implement linear approaches such as Flux Balance Analysis (FBA) [12]. Other platforms such as COPASI [13] support linear enzyme kinetics with constant catalytic rates k_{cat} in a framework equivalent to Resource Balance Analysis models [1]. Here we introduce Cell Growth Simulator as the first web-based tool designed specifically for nonlinear cellular modeling. By focusing on coarse-grained systems of up to 10 reactions, Cell Growth Simulator enables efficient exploration of cellular resource allocation and proteome efficiency [4,14–18].

The GBA framework estimates the cell state at balanced growth by solving a nonlinear optimization problem that maximizes the growth rate (μ), subject to mass conservation, nonlinear kinetic rate laws of biochemical reactions, constant cell density, and non-negative concentrations. This optimization is greatly simplified by its formulation on the flux fraction vector defined as $f := v/\mu\rho$, where v denotes the vector of mass fluxes of reactions in units [g L^{-1} h^{-1}] and ρ is the cell mass density in [g L^{-1}] [5]. Fig 1 presents a general view of the framework. A detailed mathematical description of the GBA framework can be found in the original publications [5,6].

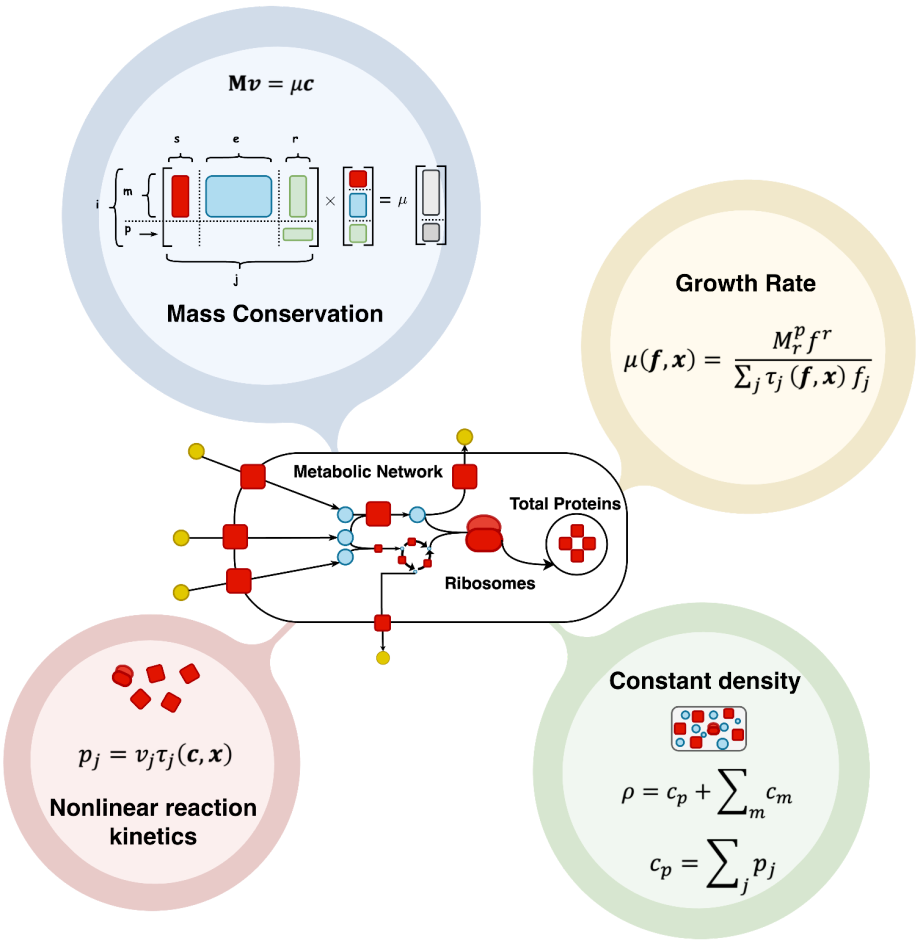


Fig 1. Schematic representation of the Growth Balance Analysis (GBA) framework as implemented in the Cell Growth Simulator. The diagram illustrates the key components and constraints governing the optimal balanced growth of a self-replicating cell model defined by the triple (M,τ,ρ), where M is the mass fraction matrix encoding the reaction network structure, τ are kinetic rate laws, and ρ is the cell density. The mass conservation constraint (blue) relates mass fluxes of reactions v with the growth rate μ and concentrations c of reactants in the model, including concentrations of metabolites c_m and the total protein c_p (equal to the sum of individual protein concentrations p). The reaction

kinetics (red) express the relationship between reaction fluxes v, protein concentrations p, and reaction turnover times τ that depend on internal concentrations c and external concentrations x. The density constraint (green) defines the total cellular density ρ as the sum of metabolite and protein concentrations. All these constraints can be encoded into a single equation (yellow) for the constrained growth rate μ in terms of the flux fractions [5].

Design and implementation

User interface and front-end development

Cell Growth Simulator is built using R (version 4.4.1) and the Shiny framework, creating an interactive web application that combines robust backend functionality with an intuitive user interface. The frontend architecture integrates HTML for content structure, CSS for visual styling, and JavaScript for dynamic interactions and enhanced client-side performance. Building upon the responsive Mazer dashboard template (https://github.com/zuramai/mazer), we extensively customized the interface through additional CSS and JavaScript implementations to create a specialized environment optimized for cell growth modeling and analysis. This layered architecture ensures both technical functionality and user accessibility while maintaining high performance across different devices and screen sizes.

Data sources and management

Cell Growth Simulator integrates comprehensive enzyme kinetic data from the BRENDA database [19], providing users access to curated turnover numbers (k_{cat}) and Michaelis constants

(K_m). The data processing pipeline filters the JSON-formatted database to include only wild-type enzyme parameters, which are then systematically organized by Enzyme Commission (EC) numbers. Through an interactive table powered by the "reactable" package [20], users can efficiently search and filter enzyme parameters based on multiple criteria including organism, EC number classification, and substrate specificity. This streamlined interface simplifies the often challenging task of identifying appropriate kinetic parameters for metabolic models.

Input data structure and model configuration

Kinetic models such as GBA describe reaction dynamics through reaction fluxes v_j that depend on protein concentrations p_j catalyzing the reaction (transport, enzymatic, or "ribosome" reaction producing proteins) and turnover times τ_j according to general kinetic rate laws:

$$p_j = v_j \tau_j(c, x) \tag{1}$$

The turnover times $\tau_j(c, x)$ depend on internal (c) and external (x) metabolite concentrations through several kinetic parameters: turnover numbers (k_{cat}) for both forward and backward directions, Michaelis constants (K_m), and where applicable, activation (KA) and inhibition constants (KI).

In Cell Growth Simulator, we implement convenience kinetics [21], a general framework that can describe reversible reactions and regulatory effects from activators and inhibitors. For an irreversible reaction j (where the backward turnover number $k^j_{catb} = 0$), the turnover time τ_j is given by:

$$\tau_j(c,x) = \frac{1}{k^j_{catf}} \Pi_m \left(\frac{c_m}{K^j_m + c_m}\right)^{-1} \left(\frac{c_m}{KA^j_m + c_m}\right)^{-1} \left(\frac{KI^j_m}{KI^j_m + c_m}\right)^{-1} \Pi_n \left(\frac{x_n}{K^j_n + x_n}\right)^{-1} . \quad (2)$$

The parameters in this equation represent:

1. Forward turnover number (k^j_{catf}): the maximum rate at which protein j (transporter, enzyme, or ribosome) can convert substrates to products.
2. Michaelis constants (K^j_m): the substrate concentration where the reaction rate reaches half its maximum value. Lower values indicate stronger protein-substrate binding. For metabolites not participating in reaction j, we set $K^j_m = 0$.
3. Regulatory constants:
    a. Activation constants (KA^j_m): quantify activator effects. Set to 0 for non-activating metabolites.
    b. Inhibition constants (KI^j_m): quantify inhibitor effects. Set to ∞ for non-inhibiting metabolites.

This formulation captures how reaction fluxes are modulated by both enzyme concentrations and metabolite-dependent effects, with the turnover time τ_j representing the time required to process one unit of substrate under the given conditions.

Cell Growth Simulator accepts GBA models in open-source spreadsheet format (ODS), consisting of the following matrices and vectors:

- **Mass fraction matrix** *M*: quantifies the mass fraction of each reactant going through each reaction, with negative entries representing reactant consumption and positive entries indicating product formation. Rows correspond to reactants, columns to reactions. Due to mass conservation, the sum of positive entries in each column equals 1, and the sum of negative entries equals -1. External reactants are denoted by the prefix "x_". By default, the last row represents the total protein concentration in mass units, while the last column corresponds to the ribosomal reaction that produces all cellular proteins. For computational efficiency, Cell Growth Simulator limits models to a maximum of 10 reactions and reactants.
- **Michaelis constant matrix** *K*: Contains K^{j}_{m} values (in [$g \cdot L^{-1}$]) representing the Michaelis constant for metabolite *m* in reaction *j*, organized in the same order as in *M*. When unspecified, a low default value of 0.1 $g \cdot L^{-1}$ is applied.
- **Matrices of regulatory constants**: inhibition constants (*KI*) and activation constants (*KA*) for each metabolite *m* in each reaction *j*, expressed in [$g \cdot L^{-1}$] and arranged as in matrix *M*. Zero values for individual entries indicate that no inhibition or activation effects are considered.
- **Turnover number matrix** (k_{cat}): Comprises two rows: the first (k_{catf}) contains forward-direction turnover numbers, while the second (k_{catb}) contains backward-direction values. Column labels match those in matrix *M*. Values represent molecules of product produced per catalyst protein per unit time (in [h^{-1}]).
- **Condition matrix**: Defines simulation parameters for each growth condition. The first row ("rho") specifies cell density in [$g \cdot L^{-1}$]. Subsequent rows contain external reactant concentrations ($g \cdot L^{-1}$), with the number of rows corresponding to the number of external

reactants (with "x_" prefix) in *M*. Each column represents a distinct growth condition for the simulations.

Cell Growth Simulator provides a downloadable GBA model template as a reference for users developing their own models. Alternatively, users can build models directly within the "Create Model" section of Cell Growth Simulator. The application employs the "shinyMatrix" R package to generate and display matrices across model tabs, with custom JavaScript modifications that enhance usability through features like arrow key navigation and full-cell editing capabilities [22].

The model creation process begins by specifying the number of reactants and reactions, followed by entering their corresponding labels. The system automatically designates "Ribosome" for the final column and "Protein" for the final row. Users must input kinetic parameters (K_m and k_{cat}) on dedicated tabs. To facilitate this process, these tabs display collapsible cards containing relevant kinetic parameters sourced from the BRENDA database [19], with an additional option to query and filter parameters based on organism, EC numbers, and enzyme substrates.

In the "Condition" tab, users input the number of external reactants and "rho" for cell density, and specify the total number of growth conditions for optimization. The application then dynamically adjusts the "condition" matrix, enabling users to configure different external concentrations for each optimization scenario. To keep runtimes low, Cell Growth Simulator limits models to a maximum of 20 growth conditions.

After uploading or creating a GBA model, the application presents a comprehensive preview of the input data. This allows users to review their model thoroughly and make direct modifications

within the interface. Once finalized, users can export their refined model in ODS format for future use, supporting an iterative development approach that progressively enhances model performance and accuracy.

Model validation and numerical optimization process in Cell Growth Simulator

After importing their GBA model, users can validate its compliance with framework requirements through the "Check Model" function. This comprehensive validation process ensures proper parameter import by performing several critical checks:

1. **Data integrity:** verifies that no missing (NA) or non-numeric values exist throughout the model;
2. **Dimensional consistency:** confirms that all matrices maintain consistent dimensions, with matching numbers of reactions and reactants;
3. **Non-negative value verification:** validates that all external concentrations are non-negative and that cell density is positive;
4. **Michaelis constant validation:** identifies instances where substrates (negative entries in the matrix M) have Michaelis constants incorrectly set to zero (K_m=0).

If K_m=0 errors are detected, the system issues a warning and automatically applies the default low value of 0.1 $g \cdot L^{-1}$ to these parameters. Should any other validation criteria fail, the application displays a specific error message directing users to reset their session and revise the problematic aspects of their model.

Upon successful validation, the system notifies users that their model meets all requirements, enabling them to proceed confidently to subsequent analysis steps. This rigorous validation workflow ensures that only properly formatted models enter the computational pipeline, reducing the likelihood of simulation errors or biologically implausible results.

Users have the option to bypass the model validation step and directly initiate growth balance analysis by selecting the "Run" button. The optimization process utilizes the "nloptr" R package, which implements non-linear optimization through the AUGLAG (augmented Lagrangian) method [23]. This framework allows users to choose from several specialized local solvers:

- **LBFGS** (Low-storage BFGS): the default solver, selected for its efficient memory usage and robust performance across most tested models;
- **MMA** (Method of Moving Asymptotes): particularly effective for highly non-linear models with complex constraints;
- **SLSQP** (Sequential Least Squares Quadratic Programming): well-suited for smaller models requiring precise constraint management.

While LBFGS serves as the default, users can select alternative solvers through the 'Advanced option' panel when facing performance challenges. Due to the nature of numerical optimization, it is not always possible to know beforehand which solver will be most appropriate for a given model. If the default solver exhibits slow convergence or fails to reach a solution that satisfies the optimization criteria, users are encouraged to evaluate alternative solvers to enhance performance outcomes.

The selection of solver can significantly impact both convergence rate and solution accuracy, with the optimal choice depending on model complexity and problem characteristics. During optimization, a dynamic progress bar tracks completion for each growth condition, providing users with real-time estimates of the total processing time. For reference, the included case study model (comprising 4 reactions and 4 internal reactants) typically completes optimization in under 1 second per condition on standard personal computing hardware.

Interactive metabolic pathway and data visualization

Cell Growth Simulator uses the "apexcharter" package [24] to generate comprehensive, interactive visualizations of numerical solutions across all simulation conditions. These visualizations include growth rates, protein concentrations, protein fractions, reaction fluxes, and metabolite concentrations. Each interactive plot supports zooming and panning to enable researchers to investigate specific trends. Users can export these visualizations in both SVG and JPEG formats for publication or presentation purposes.

Beyond conventional plotting capabilities, Cell Growth Simulator integrates d3flux (https://github.com/pstjohn/d3flux) to automatically generate interactive metabolic pathway diagrams directly within the web interface. These pathway visualizations dynamically represent the GBA analysis results from the first growth condition, with line thicknesses and node sizes proportionally reflecting flux magnitudes and concentration values, respectively. This feature enhances biological understanding, helping researchers to quickly identify key metabolic features or potential pathway errors. Users can interact with the pathway map by zooming, dragging nodes to refine the layout, and viewing detailed information on metabolites and

reactions. For collaboration and documentation purposes, users can save pathway configurations as JSON files for future editing or export them as publication-ready SVG files.

In addition to the visual outputs, users can download the numerical optimization results as a CSV file, supporting rigorous statistical analysis and custom visualization.

Results

Case study: example of a GBA model using Cell Growth Simulator

We next provide an example of a GBA model to showcase the practical application of Cell Growth Simulator, using a streamlined GBA model previously reported in the literature [5]. This model consists of 4 reactions and 4 internal reactants, with two redundant reactions, only one of which is active under optimal growth conditions.

The model assumes irreversible Michaelis-Menten kinetics, where the turnover times (τ) are determined by a matrix (K) of Michaelis constants and a vector (k_{cat}) of forward turnover numbers for each reaction. Fig 2a-c illustrates the schematic representation of the model, along with its parameters (M, K, k_{cat}, and ρ).

While most model parameters are arbitrary for demonstration purposes, several key values are based on empirical data: the ribosome reaction parameters (k_{cat} = 4.55 h^{-1} and K_m = 8.3 g L^{-1} for its primary substrate) reflect *E. coli* estimates [6], and the cell density (ρ = 340 g L^{-1}) reflects the

measured *E. coli* dry mass density [25]. This example deliberately omits regulatory mechanisms, with activation (*KA*) and inhibition (*KI*) matrices set to zero.

Users can examine the complete model in the "Model Preview" section (Fig 2d), where they can also modify parameters and save customized versions. The "Check Model" function validates parameter integrity before optimization, while the "Run" button initiates numerical optimization using the SLSQP solver. Upon completion, Cell Growth Simulator generates a comprehensive overview of key results, such as the number of simulated growth conditions, the number of converged optimizations, the maximum achieved growth rate, and the average total protein concentration (Fig 2e). In the "Interactive Plots" panel, Cell Growth Simulator displays customizable plots where the x-axis can represent external reactant concentrations or growth rate (μ), while the y-axis can display various metrics including growth rates, protein fractions, protein concentrations, reaction fluxes, and metabolite concentrations.

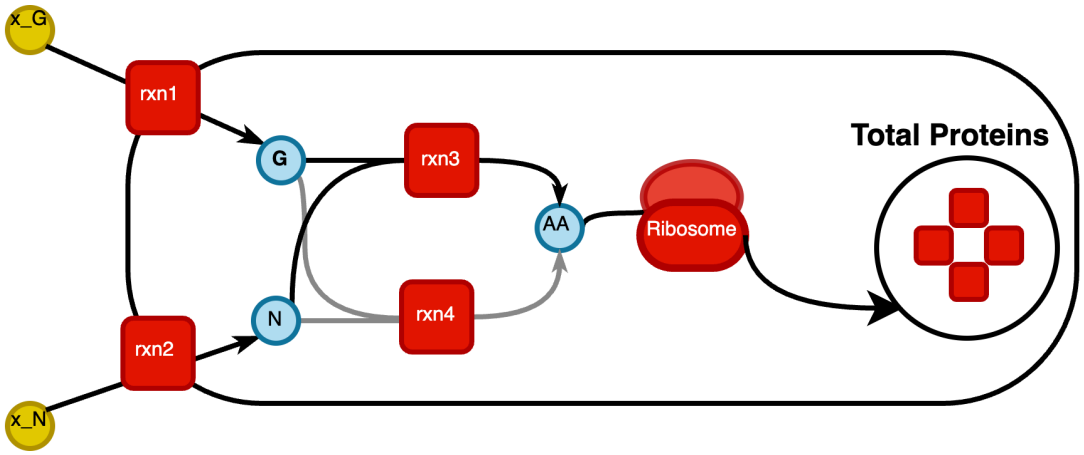
Analysis of the example model shows that growth rates at different external concentrations are consistent with the Monod equation (Fig 2g). The results show an approximately linear relationship between ribosomal mass fraction ϕ_r and growth rate μ. The protein fraction allocated to the transporter G decreases with the increasing external concentration x_G, corroborating previous theoretical findings regarding the efficiency of carbon source utilization [4,18,26,27]. These relationships reflect fundamental "bacterial growth laws," supporting the notion that optimal resource allocation drives maximal growth across diverse environmental conditions [17,28,29].

Analysis of reaction activities reveals a dynamic metabolic shift between rxn_3 and rxn_4: rxn_3 is active at lower growth rates (up to 1.58 h^{-1}), while rxn_4 is utilized at higher growth rates (Fig

2f). Both reactions share a similar structural role in the matrix *M* (linearly dependent columns when considering only the 4 internal reactants) but have different kinetic parameters, resulting in different resource allocation costs; this favors the sole use of rxn_3 at lower growth rates and the sole use of rxn_4 at higher growth rates.

The "Pathway Visualization" panel offers an intuitive graphical representation of the reaction network (Fig 2h), with visual elements scaled proportionally to simulation results from the first growth condition (representing maximal external concentrations in this example). Node sizes of metabolites reflect metabolite concentrations, while line thicknesses indicate protein concentrations. The redundant pathway, rxn_3 in this case, is highlighted using a dashed line. Users can export the complete optimization results as a ".csv" file through the "Results" button in the main panel.

a)



b)

	rxn1	rxn2	rxn3	rxn4	Rib
x_G	-1	0	0	0	0
x_N	0	-1	0	0	0
G	1	0	-0.7	-0.8	0
N	0	1	-0.3	-0.2	-0.2
AA	0	0	1	1	-0.8
Prot	0	0	0	0	1

c)

K_m	rxn1	rxn2	rxn3	rxn4	Rib
x_G	10	0	0	0	0
x_N	0	10	0	0	0
G	0	0	10	5	0
N	0	0	10	5	10
AA	0	0	0	0	8.3
Prot	0	0	0	0	0

k_{cat}	rxn1	rxn2	rxn3	rxn4	Rib
kcat_f	30	20	20	18	4.55
kcat_b	0	0	0	0	0

d)

Your GBA Model

Mass fraction | Km | KI | KA | kcat | Condition

Your Mass Fraction (M) matrix

	rxn1	rxn2	rxn3	rxn4	Ribosome
x_G	-1	0	0	0	0
x_N	0	-1	0	0	0
G	1	0	-0.7	-0.8	0
N	0	1	-0.3	-0.2	-0.2
AA	0	0	1	1	-0.8
Protein	0	0	0	0	1

e)

Number of Conditions
15

Maximum Growth rate
2.2488 h^{-1}

Average of Total Protein Concentration
203.28 [g L^{-1}]

Number of Converged Optimization
15

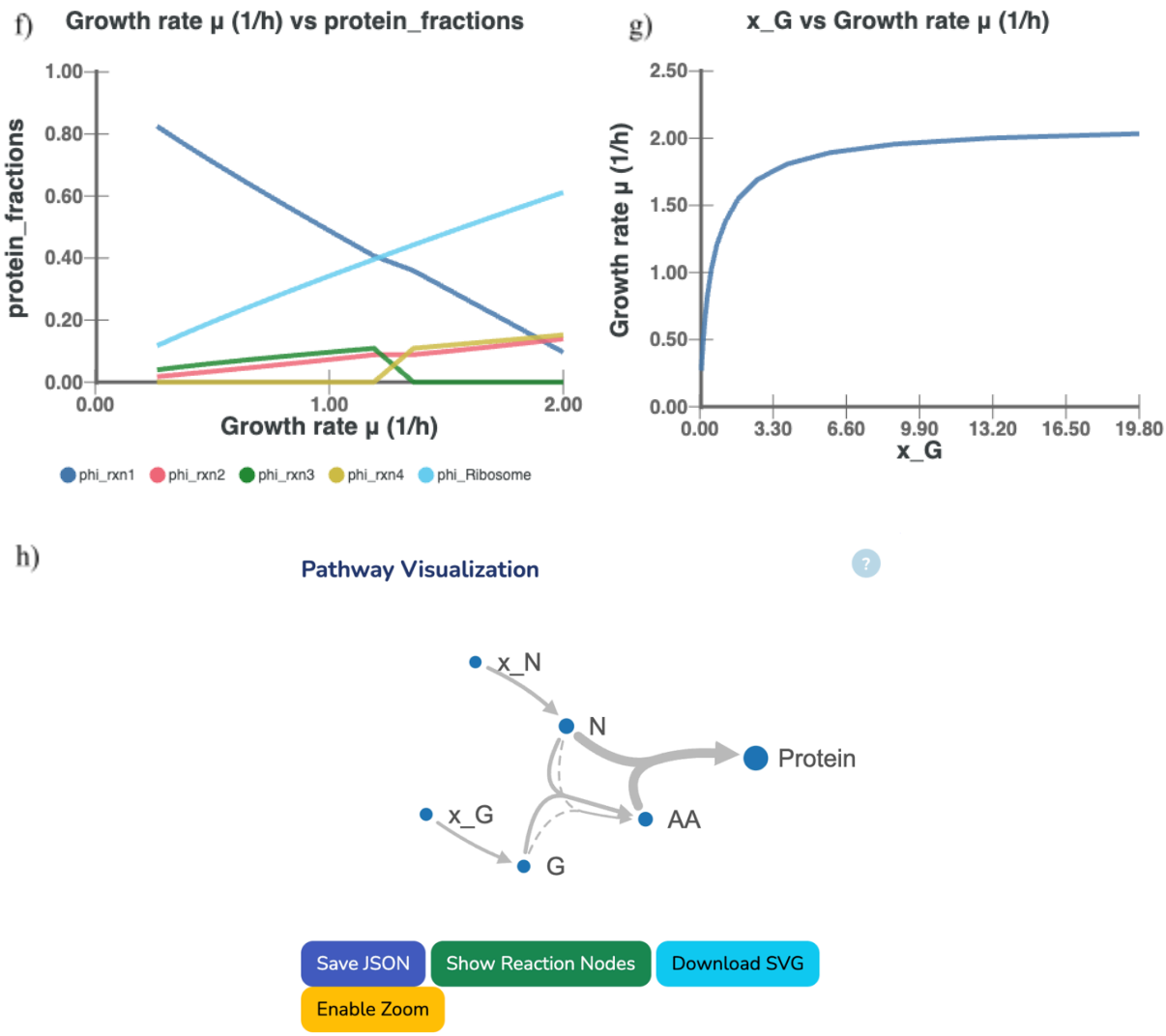


Fig 2. Analysis of an example GBA model using Cell Growth Simulator. (a) Schematic representation of the simplified GBA model, illustrating reaction pathways, metabolite interactions, and redundant routes. (b) Mass fraction matrix (M), detailing the stoichiometric relationships between reactants (rows) and reactions (columns), with negative values indicating consumption and positive values indicating production. Note that in panel b-c, "Ribosome" is shortened to "Rib" and "Protein" to "Prot" for clarity. (c) Kinetic parameter set comprising the Michaelis constant matrix (K) and turnover number vector (k_{cat}). (d) Cell Growth Simulator

model preview interface displaying imported parameters and allowing for real-time modification before analysis. (e) Results summary dashboard presenting key optimization outcomes: total growth conditions analyzed, number of successfully converged optimizations, maximum achieved growth rate, and average total protein concentration. (f) Optimal protein allocation across varying growth rates. (g) Monod-like growth curve in response to external carbon source availability. (h) Interactive metabolic pathway visualization with node sizes proportional to metabolite concentrations and edge thicknesses reflecting protein allocation, with dashed lines indicating redundant pathways.

Discussion

We introduce Cell Growth Simulator, a user-friendly web application for Growth Balance Analysis (GBA) that enhances accessibility for researchers interested in exploring cell growth and resource allocation under nonlinear kinetic rate laws. Cell Growth Simulator provides an intuitive interface built with R/Shiny, supplemented by HTML, CSS, and JavaScript to enhance the user experience. The application allows users to construct and modify models through interactive matrices that resemble spreadsheet environments (here, .ods files), a familiar open-source format for many researchers. This design choice simplifies the process of inputting and managing model parameters—such as the mass fraction matrix (M), kinetic parameters (K_m and k_{cat}), cell density, and external growth conditions—without requiring direct coding. Cell Growth Simulator significantly streamlines the incorporation of kinetic parameters into models by integrating data tables retrieved from the BRENDA enzyme database [19].

Cell Growth Simulator has an emphasis on customizable and interactive visualization to aid in the interpretation of optimization results. By integrating "d3flux", a package specifically designed for metabolic pathway visualization, and the “Apexcharter” package for generating interactive plots, the application provides a dynamic and informative representation of the metabolic network and the simulation results.

While several web-based tools for metabolic modeling exist—such as Escher, ModelExplorer, CAVE, and Fluxer—they only focus on linear approaches like Flux Balance Analysis (FBA) and its extensions [8–12]. These platforms excel at providing user-friendly interfaces for constructing and visualizing metabolic networks, but their modeling frameworks are not designed to handle kinetic rate laws. CNApy [7] offers a graphical user interface for constraint-based modeling using COBRA [30] methods, again focusing on linear optimization techniques like FBA. COPASI [13], on the other hand, is a comprehensive tool for modeling biochemical networks including nonlinear kinetic rate laws, supporting both deterministic and stochastic simulations. It provides extensive features like parameter estimation and steady-state analysis for detailed kinetic models. However, COPASI can be complex and may present a steep learning curve, especially for users without extensive computational experience. Additionally, while COPASI handles detailed kinetic models of specific pathways, it may not be optimized for modeling resource allocation and proteome constraints in whole-cell models with non-linear kinetics, as it only considers linear enzyme kinetics with a constant catalytic rate (k_{cat}) and ignores the dilution of metabolites, comparable to RBA-type models [1–3]. In contrast, Cell Growth Simulator allows the study of coarse-grained models with nonlinear kinetics, enabling users to explicitly consider metabolite concentrations and their effects on reaction fluxes through kinetic rate laws.

By focusing on simplified models that capture essential features of cellular metabolism [14], Cell Growth Simulator allows researchers to investigate critical aspects of cellular systems—such as resource allocation and growth optimization—with minimal computational overhead.

The current Cell Growth Simulator implementation is primarily tailored for researchers interested in exploring principles of growth laws and resource allocation under nonlinear kinetic constraints, particularly those with limited computational expertise. The application's main constraint is its limitation to a maximum of 10 reactions and 10 internal reactants, implemented to ensure tractable numerical optimization and reasonable computation times. Future development plans include enhancing computational efficiency to accommodate larger-scale models through the implementation of more efficient numerical solvers and the incorporation of parallelization strategies for multi-condition analyses.

Cell Growth Simulator addresses key challenges in analyzing self-replicating cell models with nonlinear kinetics, making such fundamental analyses accessible to a wide range of researchers and fostering interdisciplinary collaboration by lowering technical barriers. Furthermore, Cell Growth Simulator's approach aligns with the growing emphasis on open science and reproducibility. By offering a platform that is both user-friendly and transparent, it supports the sharing of models and results, facilitating peer review and collaborative contributions to advancing our understanding of cellular metabolism and growth.

Availability and Future Directions

Cell Growth Simulator is freely available as a web server at https://cellgrowthsim.com/, where researchers can analyze and share their models without additional registration or login

requirements. Looking ahead, we plan to expand Cell Growth Simulator's capabilities in two key areas. First, we aim to expand the current model size limitation (currently up to 10 reactions and 10 internal reactants) by implementing more efficient numerical solvers and optimizing parallelization strategies. This will allow the exploration of larger, more detailed models that capture additional levels of metabolic complexity. Second, we intend to incorporate dynamical simulations so that users can study time-dependent behavior and transient responses on reaction networks in growing cells. We welcome community-driven enhancements and collaborative projects to further improve Cell Growth Simulator's functionality, usability, and applications in diverse biological contexts.

Data availability

Cell Growth Simulator is a web server freely accessible without login requirement at https://cellgrowthsim.com/.

The source code for Cell Growth Simulator is available at: https://github.com/Sijr73/CellGrowthSimulator.

Acknowledgement

We thank Martin J. Lercher for many valuable discussions and suggestions. We are also grateful to Thomas Spitzlei for his support in deploying the application online.

Funding

This work was supported by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) [CRC 1310 and, under Germany's Excellence Strategy, EXC 2048/1–390686111] and by the European Union [ERC AdG "MechSys"–Project ID 101055141].

Conflict of interest

The authors declare no conflict of interest.

References

1. Goelzer A, Fromion V. Bacterial growth rate reflects a bottleneck in resource allocation. Biochimica et Biophysica Acta (BBA) - General Subjects. 2011;1810: 978–988. doi:10.1016/j.bbagen.2011.05.014

2. O'Brien EJ, Lerman JA, Chang RL, Hyduke DR, Palsson BØ. Genome‐scale models of metabolism and gene expression extend and refine growth phenotype prediction. Molecular Systems Biology. 2013;9: 693. doi:10.1038/msb.2013.52

3. Sánchez BJ, Zhang C, Nilsson A, Lahtvee P, Kerkhoven EJ, Nielsen J. Improving the phenotype predictions of a yeast genome‐scale metabolic model by incorporating enzymatic constraints. Molecular Systems Biology. 2017;13: 935. doi:10.15252/msb.20167411

4. Molenaar D, Van Berlo R, De Ridder D, Teusink B. Shifts in growth strategies reflect tradeoffs in cellular economics. Molecular Systems Biology. 2009;5: 323. doi:10.1038/msb.2009.82

5. Dourado H, Liebermeister W, Ebenhöh O, Lercher MJ. Mathematical properties of optimal fluxes in cellular reaction networks at balanced growth. Mendes P, editor. PLoS Comput Biol. 2023;19: e1011156. doi:10.1371/journal.pcbi.1011156

6. Dourado H, Lercher MJ. An analytical theory of balanced cellular growth. Nat Commun. 2020;11: 1226. doi:10.1038/s41467-020-14751-w

7. Thiele S, Von Kamp A, Bekiaris PS, Schneider P, Klamt S. CNApy: a CellNetAnalyzer

GUI in Python for analyzing and designing metabolic networks. Martelli PL, editor. Bioinformatics. 2022;38: 1467–1469. doi:10.1093/bioinformatics/btab828

8. King ZA, Dräger A, Ebrahim A, Sonnenschein N, Lewis NE, Palsson BO. Escher: A Web Application for Building, Sharing, and Embedding Data-Rich Visualizations of Biological Pathways. Gardner PP, editor. PLoS Comput Biol. 2015;11: e1004321. doi:10.1371/journal.pcbi.1004321

9. Martyushenko N, Almaas E. ModelExplorer - software for visual inspection and inconsistency correction of genome-scale metabolic reconstructions. BMC Bioinformatics. 2019;20: 56. doi:10.1186/s12859-019-2615-x

10. Mao Z, Yuan Q, Li H, Zhang Y, Huang Y, Yang C, et al. CAVE: a cloud-based platform for analysis and visualization of metabolic pathways. Nucleic Acids Research. 2023;51: W70–W77. doi:10.1093/nar/gkad360

11. Hari A, Lobo D. Fluxer: a web application to compute, analyze and visualize genome-scale metabolic flux networks. Nucleic Acids Research. 2020;48: W427–W435. doi:10.1093/nar/gkaa409

12. Orth JD, Thiele I, Palsson BØ. What is flux balance analysis? Nat Biotechnol. 2010;28: 245–248. doi:10.1038/nbt.1614

13. Hoops S, Sahle S, Gauges R, Lee C, Pahle J, Simus N, et al. COPASI—a COmplex PAthway SImulator. Bioinformatics. 2006;22: 3067–3074. doi:10.1093/bioinformatics/btl485

14. Doan DT, Hoang MD, Heins A-L, Kremling A. Applications of Coarse-Grained Models in Metabolic Engineering. Front Mol Biosci. 2022;9: 806213. doi:10.3389/fmolb.2022.806213

15. Ghaffarinasab S, Lercher MJ, Dourado H. Growth balance analysis models of cyanobacteria for understanding resource allocation strategies. 2023. doi:10.1101/2023.04.18.537335

16. Hui S, Silverman JM, Chen SS, Erickson DW, Basan M, Wang J, et al. Quantitative proteomic analysis reveals a simple strategy of global resource allocation in bacteria. Molecular Systems Biology. 2015;11: 784. doi:10.15252/msb.20145697

17. Scott M, Gunderson CW, Mateescu EM, Zhang Z, Hwa T. Interdependence of Cell Growth and Gene Expression: Origins and Consequences. Science. 2010;330: 1099–1102. doi:10.1126/science.1192588

18. Weiße AY, Oyarzún DA, Danos V, Swain PS. Mechanistic links between cellular trade-offs, gene expression, and growth. Proc Natl Acad Sci USA. 2015;112. doi:10.1073/pnas.1416533112

19. Chang A, Jeske L, Ulbrich S, Hofmann J, Koblitz J, Schomburg I, et al. BRENDA, the ELIXIR core data resource in 2021: new developments and updates. Nucleic Acids Research. 2021;49: D498–D508. doi:10.1093/nar/gkaa1025

20. Lin G. reactable: Interactive Data Tables for R. 2019. p. 0.4.4. doi:10.32614/CRAN.package.reactable

21. Liebermeister W, Klipp E. Bringing metabolic networks to life: convenience rate law and

thermodynamic constraints. Theoretical Biology and Medical Modelling. 2006;3: 41. doi:10.1186/1742-4682-3-41

22. Andreas Neudecker. shinyMatrix: Shiny Matrix Input Field. 2019. p. 0.8.0. doi:10.32614/CRAN.package.shinyMatrix

23. Ypma J, Johnson SG. nloptr: R Interface to NLopt. 2011. p. 2.1.1. doi:10.32614/CRAN.package.nloptr

24. Perrier V, Meyer F. apexcharter: Create Interactive Chart with the JavaScript "ApexCharts" Library. 2019. p. 0.4.4. doi:10.32614/CRAN.package.apexcharter

25. Zimmerman SB, Trach SO. Estimation of macromolecule concentrations and excluded volume effects for the cytoplasm of Escherichia coli. Journal of Molecular Biology. 1991;222: 599–620. doi:10.1016/0022-2836(91)90499-V

26. Burnap RL. Systems and Photosystems: Cellular Limits of Autotrophic Productivity in Cyanobacteria. Front Bioeng Biotechnol. 2015;3. doi:10.3389/fbioe.2015.00001

27. Faizi M, Zavřel T, Loureiro C, Červený J, Steuer R. A model of optimal protein allocation during phototrophic growth. Biosystems. 2018;166: 26–36. doi:10.1016/j.biosystems.2018.02.004

28. Scott M, Hwa T. Bacterial growth laws and their applications. Current Opinion in Biotechnology. 2011;22: 559–565. doi:10.1016/j.copbio.2011.04.014

29. You C, Okano H, Hui S, Zhang Z, Kim M, Gunderson CW, et al. Coordination of bacterial

proteome with metabolism by cyclic AMP signalling. Nature. 2013;500: 301–306. doi:10.1038/nature12446

30. Ebrahim A, Lerman JA, Palsson BO, Hyduke DR. COBRApy: COnstraints-Based Reconstruction and Analysis for Python. BMC Syst Biol. 2013;7: 74. doi:10.1186/1752-0509-7-74