UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

Multi-scale host-aware modeling for analysis and tuning of synthetic gene circuits for bioproduction

PhD dissertation

Author: Fernando Nóbel Santos-Navarro
Supervisors: Jesús Andrés Picó i Marco
Alejandro Vignoni

April 2022

Instituto Universitario de Automática e Informática Industrial
Departamento de Ingeniería de Sistemas y Automática
Universitat Politècnica de València

This research was funded by MCIN/AEI/10.13039/501100011033 grant number PID2020-117271RB-C21, and MINECO/AEI, EU grant number DPI2017-82896-C2-1-R.

The author was recipient of the grant "Programa para la Formación de Personal Investigador (FPI) de la Universitat Politècnica de València — Subprograma 1 (PAID-01-2017)".

The author was also a grantee of the pre-doctoral stay "Ayudas para Movilidad de Estudiantes de Doctorado de la Universitat Politècnica de València 2019".

The Control Theory and Systems Biology Lab of the ETH Zürich is acknowledged for accepting the author in their facilities as pre-doctoral stay and their valuable collaboration sharing knowledge.

A mi familia: Nóbel, Pilar, Eduardo y Anna.

Agradecimientos

Escribir estas palabras pone fin a una etapa de mi vida.

Apenas me identifico con ese estudiante de ingeniería electrónica que empezó a interesarse por los maravillosos problemas de la biología. Y si no me identifico con mi yo del pasado, es debido a que he crecido gracias al apoyo, vivencias y enseñanzas de todos los compañeros, amigos y familiares que me han acompañado en este viaje de descubrimiento y —más importante aún— de autodescubrimiento.

Primero quiero agradecer a Jesús Picó, por haberme acompañado durante todos estos años. Has sido uno de mi mayores referentes tanto a nivel profesional como a nivel personal. He disfrutado mucho aprendiendo de ti hábitos y conductas que considero realmente valiosas como: la paciencia, el trabajo duro, la pasión por el deporte, y, sobretodo, tu gran forma proactiva y calmada de superar los obstáculos. También quiero agradecer a Alejandro Vignoni y Yadira Boada. Vuestra ayuda y apoyo han sido vitales para mi doctorado, y todas las experiencias que hemos compartido juntos en el trabajo, y fuera del trabajo, han sido geniales, ojalá poder repetir juntos algún viaje de congreso o ir al «jamón del viernes». En especial, quiero agradecer a Enric Picó por reavivar en mí la pasión por las matemáticas, es una pena que no pudiésemos continuar con los análisis de contractividad: una verdadera pena. También quiero dar las gracias a Jose Luis Navarro por su plena disposición para ayudarme y para darme consejos realmente precisos y útiles. Por último, quiero agradecer a muchos otros investigadores y profesores que en algún momento confiaron en mí: Pablo Carbonell, Sebastián Nuñez, Hernan De Battista y Fabricio Garelli.

I have only good words for my stay in Basel at the Control Theory and Systems Biology Lab with Mustafa Kammash: I really felt like one of the team. I am amazed by the professionalism of your team and your personal kindness. Thanks to Morice Filo, I really enjoyed working with you, and I learned a lot by analyzing and extending the antithetic controller together. And I also want to thank the rest of the team: Joaquín, Timothy, Tomasz, Armin, Stephanie, Ankit, Armin, and Sara. And in particular, thanks to Sant, Moritz, and Anja: I miss going to the Indian restaurant, swimming in the river, and sharing moments with you.

I also want to thank two people—although I never met them in real person—that have been indispensable to my PhD. Thanks to Uri Alon, you have been one of the primary motivators to keep on with my PhD and not get lost in the «cloud». And thanks to Eric S. Lander, thanks for showing me and unraveling the secrets of life. Thank you both for your passion for teaching and your online videos and courses: I really feel I have been one of your students.

Gracias a todos los miembros de «La Sala». Con vosotros he vivido muchas experiencias geniales y compartir esa sala de trabajo con vosotros fue una de las mejores partes de todo mi doctorado. Gracias Vanesa, Temo, Carlos, Carles, Henry, Uriel, Juanfer, Michelle, Julio y Eslam. Atesoraré con mucho cariño esos momentos de cotidianidad para siempre.

También quiero agradecer a amigos que me han ayudado de manera directa o indirecta para mantener la motivación para acabar el doctorado. Gracias Clara, Iván, Guille, Laura, Alba y Joan. Y en especial, gracias a Borja, que me ayudó con varias decisiones técnicas para sacar adelante *OneModel*.

Gracias a mi familia: a mi padre Nóbel, a mi madre Pilar y a mi hermano Edu, habéis hecho posible que haya llegado a ser quien soy hoy. He disfrutado enormemente convivir con vosotros en la última etapa del doctorado. Siempre habéis tomado las decisiones necesarias para que pudiese crecer y desarrollar mis propias «alas» para lograr las metas por mí mismo. Gracias.

Y por último, gracias Anna: por ser, estar y crecer junto a mí.

Gracias a todos. Doblemente gracias.

Abstract

This Thesis was devoted to the multi-scale host-aware analysis and tuning of synthetic gene circuits for bioproduction. The main objectives were:

- The development of a reduced-size host-aware model for simulation and analysis purposes.
- The development of a software toolbox for modeling and simulation, oriented to synthetic biology.
- The implementation of a multi-scale model that considers the scales relevant to bioproduction (bioreactor, cell, and synthetic circuit).
- The host-aware analysis of the antithetic controller, as an example of the application of the developed tools.
- The development and experimental validation of robust control laws for continuous bioreactors.

The work presented in this Thesis covers the three scales of the bioproduction process. The first scale is the bioreactor: this scale considers the macroscopic substrate and biomass dynamics and how these dynamics connect to the internal state of the cells. The second scale is the host cell: this scale considers the internal dynamics of the cell and the competition for limited shared resources for protein expression. The third scale is the synthetic genetic circuit: this scale considers the dynamics of expressing exogenous synthetic circuits and the burden they induce

on the host cell. Finally, as a «fourth» scale, part of the Thesis was devoted to developing software tools for modeling and simulation.

This document is divided into seven chapters. Chapter 1 is an overall introduction to the Thesis work and its justification; it also presents a visual map of the Thesis and lists the main contributions. Chapter 2 shows the development of the host-aware model (Chapters 4 and 5 make use of this model for their simulations). Chapter 3 presents *OneModel*: a software tool developed in the Thesis that facilitates modeling and simulation for synthetic biology—in particular, it facilitates the use of the host-aware model—. Chapter 4 uses the host-aware model to assemble the multi-scale model considering the bioreactor and analyzes the titer, productivity (rate), and yield in expressing an exogenous protein. Chapter 5 analyzes a more complex circuit, the recently proposed and highly cited antithetic biomolecular controller, using the host-aware model. Chapter 6 shows the design of nonlinear control strategies that allow controlling the concentration of biomass in a continuous bioreactor in a robust way. Chapter 7 summarizes and presents the main conclusions of the Thesis. Appendix A shows the theoretical development of the host-aware model.

This Thesis emphasizes the importance of studying cell burden in biological systems since these effects are very noticeable and generate interactions between seemingly unconnected circuits. The Thesis provides tools to model, simulate and design synthetic genetic circuits taking into account these burden effects and allowing the development of models that connect phenomena in synthetic genetic circuits, ranging from the intracelullar dynamics of gene expression to the macroscopic dynamics of the population of cells inside the bioreactor.

Resumen

Esta Tesis ha sido dedicada al modelado multiescala considerando al anfitrión celular para el análisis y ajuste de circuitos genéticos sintéticos para bioproducción. Los objetivos principales fueron:

- El desarrollo de un modelo que considere el anfitrión celular de tamaño reducido enfocado para simulación y análisis.
- El desarrollo de herramientas de programación para el modelado y la simulación, orientada a la biología sintética.
- La implementación de un modelo multiescala que considere las escalas relevantes para la bioproducción (biorreactor, célula y circuito sintético).
- El análisis del controlador antitético considerando las interacciones célula-circuito, como ejemplo de aplicación de las herramientas desarrolladas.
- El desarrollo y la validación experimental de leyes de control robusto para biorreactores continuos.

El trabajo presentado en esta Tesis cubre las tres escalas del proceso de bioproducción. La primera escala es el biorreactor: esta escala considera la dinámica macroscópica del sustrato y la biomasa, y como estas dinámica se conecta con el estado interno de las células. La segunda escala es la célula anfitriona: esta escala considera la dinámica interna de la célula y la competencia por los recursos limitados compartidos para la expresión de proteínas. La tercera escala es el circuito genético sintético: esta escala considera la dinámica de expresión de los circuitos

sintéticos exógenos y la carga que inducen en la célula anfitriona. Por último, como «cuarta» escala, parte de la Tesis se ha dedicado a desarrollar herramientas de software para el modelado y la simulación.

Este documento se divide en siete capítulos. El Capítulo 1 es una introducción general al trabajo de la Tesis y su justificación; también presenta un mapa visual de la Tesis y enumera las principales contribuciones. El Capítulo 2 muestra el desarrollo del modelo del anfitrión celular (los Capítulos 4 y 5 hacen uso de este modelo para sus simulaciones). El Capítulo 3 presenta *OneModel*: una herramienta de software desarrollada en la Tesis que facilita el modelado y la simulación en biología sintética, en particular, facilita el uso del modelo del anfitrión celular. El Capítulo 4 utiliza el modelo del anfitrión celular para montar el modelo multiescala que considera el biorreactor y analiza el título, la productividad y el rendimiento en la expresión de una proteína exógena. El Capítulo 5 analiza un circuito más complejo, el recientemente propuesto y muy citado controlador biomolecular antitético, utilizando el modelo del anfitrión celular. El Capítulo 6 muestra el diseño de estrategias de control no lineal que permiten controlar la concentración de biomasa en un biorreactor continuo de forma robusta. El Capítulo 7 resume y presenta las principales conclusiones de la Tesis. En el Apéndice A se muestra el desarrollo teórico del modelo del anfitrión celular.

Esta Tesis destaca la importancia de estudiar la carga celular en los sistemas biológicos, ya que estos efectos son muy notables y generan interacciones entre circuitos aparentemente independientes. La Tesis proporciona herramientas para modelar, simular y diseñar circuitos genéticos sintéticos teniendo en cuenta estos efectos de carga y permite el desarrollo de modelos que conecten estos fenómenos en los circuitos genéticos sintéticos, que van desde la dinámica intracelular de la expresión génica hasta la dinámica macroscópica de la población de células dentro del biorreactor.

Resum

Aquesta Tesi tracta del modelat multiescala considerant l'amfitrió cel·lular per a l'anàlisi i ajust de circuits genètics sintètics per a bioproducció. Els objectius principals van ser:

- El desenvolupament d'un model de grandària reduïda que considere l'amfitrió cel·lular, enfocat al seu ús en simulació i anàlisi.
- El desenvolupament d'eines de programari per al modelatge i la simulació, orientada a la biologia sintètica.
- La implementació d'un model multiescala que considere les escales rellevants per a la bioproducció (bioreactor, cèl·lula i circuit sintètic).
- L'anàlisi del controlador antitètic considerant les interacciones cèl·lula-circuit, com a exemple d'aplicació de les eines desenvolupades.
- El desenvolupament i la validació experimental de lleis de control robust per a bioreactors continus.

El treball presentat en aquesta Tesi cobreix les tres escales del procés de bioproducció. La primera escala és el bioreactor: aquesta escala considera la dinàmica macroscòpica del substrat i la biomassa, i com aquestes dinàmiques es connecten amb l'estat intern de les cèl·lules. La segona escala és la cèl·lula amfitriona: aquesta escala considera la dinàmica interna de la cèl·lula i la competència pels recursos limitats compartits per a l'expressió de proteïnes. La tercera escala és la del circuit genètic sintètic: aquesta escala considera la dinàmica d'expressió

de circuits sintètics exógens i la càrrega que indueixen en la cèl·lula amfitriona. Finalment, com a «quarta» escala, part de la Tesi s'ha dedicat a desenvolupar eines de programari per al modelatge i la simulació.

Aquest document es divideix en set capítols. El Capítol 1 és una introducció general al treball de la Tesi i la seua justificació; també presenta un mapa visual de la Tesi i enumera les principals contribucions. El Capítol 2 mostra el desenvolupament del model de l'amfitrió cel·lular (els Capítols 4 i 5 fan ús d'aquest model per a les seues simulacions). El Capítol 3 presenta *OneModel*: una eina de programari desenvolupada en la Tesi que facilita el modelatge i la simulació en biologia sintètica, en particular, facilita l'ús del model de l'amfitrió cel·lular. El Capítol 4 utilitza el model de l'amfitrió cel·lular per a muntar el model multiescala que considera el bioreactor i analitza el títol, la productivitat i el rendiment en l'expressió d'una proteïna exògena. El Capítol 5 analitza un circuit més complex, el recentment proposat i molt citat controlador biomolecular antitètic, utilitzant el model de l'amfitrió cel·lular. El Capítol 6 mostra el disseny d'estratègies de control no lineal que permeten controlar la concentració de biomassa en un bioreactor continu de manera robusta. El Capítol 7 resumeix i presenta les principals conclusions de la Tesi. En l'Apèndix A es mostra el desenvolupament teòric del model de l'amfitrió cel·lular.

Aquesta Tesi destaca la importància d'estudiar la càrrega cel·lular en els sistemes biològics, ja que aquests efectes són molt notables i generen interaccions entre circuits aparentment independents. La Tesi proporciona eines per a modelar, simular i dissenyar circuits genètics sintètics tenint en compte aquests efectes de càrrega i permet el desenvolupament de models que connecten aquests fenòmens en els circuits genètics sintètics, que van des de la dinàmica intracel·lular de l'expressió gènica fins a la dinàmica macroscòpica de la població de cèl·lules dins del bioreactor.

Contents

List of Acronyms

CLI	Command-Line Interface
CSS	Cascading Style Sheets
DAE	Differential-Algebraic Equation
dFBA	Dynamic Flux Balance Analysis
FBA	Flux Balance Analysis
GFP	Green Fluorescent Protein
GUI	Graphical User interface
HTML	HyperText Markup Language
mRNA	Messenger RNA
ODE	Ordinary Differential Equation
OD	Optical Density
PI	Proportional-Integral
PyPI	Python Package Index
QSS	Quasi-Steady State
RBS	Ribosome Binding Site
RNAP	RNA polymerase
RNA	Ribonucleic Acid
RRS	Resources Recruitment Strength
SBML	Systems Biology Markup Language
spMSR	Specific Protein Mass Synthesis Rate
TRY	Titer, Productivity (Rate), and Yield

Chapter 1

Introduction

"As a physicist, I was used to studying matter that obeys precise mathematical laws. But cells are matter that dances."
—Uri Alon, *An Introduction to Systems Biology*

1.1 Justification

At least as we understand it, a doctoral thesis is a living entity that evolves adapting to the new results and needs of its environment. What you have in your hands is an effort to arrange logically all those ideas that we have considered valuable and have withstood the weight of time.

Due to my background in electronic engineering, I initially started the Thesis working on the design, implementation, and control of bioreactors. However, the extraordinary problems of biology gradually took over the Thesis, leaving the work with bioreactors for just a single chapter.

I would not like to convey that this work merely started from problem A to arrive at solution B: the reality was much more chaotic. As Uri Alon would say, I went into the «cloud», not knowing where I would come out. I went through many paths and ideas that led me to dead ends (unfortunately, I don't have time to show them here). However, I want to exemplify the work process in an uncertain new territory with one of the most significant breakthroughs in the Thesis. We were developing a new host-aware model (the one we show in Chapter 2); the original idea was to create a minimal model that could capture the essence of cell-burden interactions. To simplify it to the minimum, we started with a vast model—more

than twenty states and too many equations—. I spent several months on this simplification process but reached a point where I could not simplify the model any further, but still felt it was not enough: there was a complexity that, if removed, would result in beautiful and simple-to-use equations.

Finally, I had to accept that it was impossible to do so. However, the model acquired its powerful predictive capacities thanks to embracing that complexity which I was trying hard to eliminate. This freedom—to explore and get lost in the «cloud»—led us to many unexpected and exciting results.

As explained in the Abstract, this Thesis's ultimate goal is to develop methods that are at the base of designing, analyzing, and implementing synthetic gene systems for bioproduction. This goal is inherently multi-scale; the work presented here offers solutions from different fields of knowledge at each scale.

- **Bioreactor scale**

  This scale is related to the macroscopic dynamics of the substrate and biomass in the bioreactor. The goal is to develop robust control laws that optimize bioproduction by dynamically modifying bioreactor inflows and outflows as a function of substrate, growth rate, or biomass measurements. Small-scale turbidostats, a class of continuous bioreactors, are gaining interest for characterization and scaling-up of biotechnological processes. Different control goals can be considered in these bioreactors, including the regulation of the cells' specific growth rate or the cell density at a prescribed value. These goals can be embedded within strategies for optimal bioproduction. The problem of regulating the specific growth rate was already solved in previous works of the SB2CL research group. In this Thesis, the problem considered is regulating the cell density. This problem is of particular interest to characterize genetic circuits at pre-industrial scales, where keeping a constant number of cells eases the analysis. It is also of interest at an industrial scale, at which an increasing number of bioprocesses are being migrated to continuous bioreactors for financial reasons.

- **Host scale**

  This scale is related to the internal dynamics of the host cell. The goal is to understand the effects of cell burden on the availability of limited cell resources and how this influences protein expression, ultimately determining the bioproduction observed at the bioreactor scale. The interrelations among the cell environment, its metabolism, and the engagement of cell resources needed for gene expression result in host-circuit interactions between gene circuits and their cell host. These interactions induce competition for com-

mon shared cell resources affecting gene expression and cell growth. Models of gene expression considering host-circuit interactions are relevant for understanding the strategies and trade-offs associated with the efficient design of heterologous protein expression systems and synthetic genetic circuits. In this Thesis, a small-size host-aware model is developed for the bacterial host *E. coli*. The model has enough granularity to provide good predictions of the host dynamics, the expression of the genes of interest, and their interactions while having a small number of parameters to be estimated.

- **Synthetic circuit scale**

  This scale is related to the design of synthetic genetic circuits. The goal is to design synthetic genetic circuits with interesting properties for optimal bioproduction, such as perfect regulation at steady-state and robustness to cell burden perturbations while also avoiding excess overburdening of their host. This Thesis addresses the analysis of the antithetic biomolecular controller [16] under realistic conditions considering the interaction with host dynamics.

1.2 Thesis outline

This Thesis is not a compendium of articles. However, it has been written following the structure of a compendium. Each chapter is based on the work that appeared in one or several conference and journal publications and is independent of the others. There is no global state of the art in the Thesis: each chapter has its introduction with the state of the art and conclusions. Nevertheless, indeed, all chapters are interrelated, as shown in Figure 1.1 and described below.

Chapter 2 is devoted to the central topic of the Thesis: the effect of expression of exogenous proteins on the cell host. The Chapter considers a host-aware model and seeks to understand the relationships between protein expression and cell growth caused by the competition for limited cell resources.

Simulating complex synthetic gene circuits integrated with the host-aware model is not easy with the available software tools. Chapter 3 focuses on developing two software tools, *OneModel* and *sbml2dae*, that simplify modeling and simulation tasks for synthetic biology.

Once we have a host-aware model and efficient software tools that allow us to design and analyze exogenous gene circuits, Chapter 4 implements a multi-scale (gene circuit, host, and bioreactor) model. The multi-scale model is then used

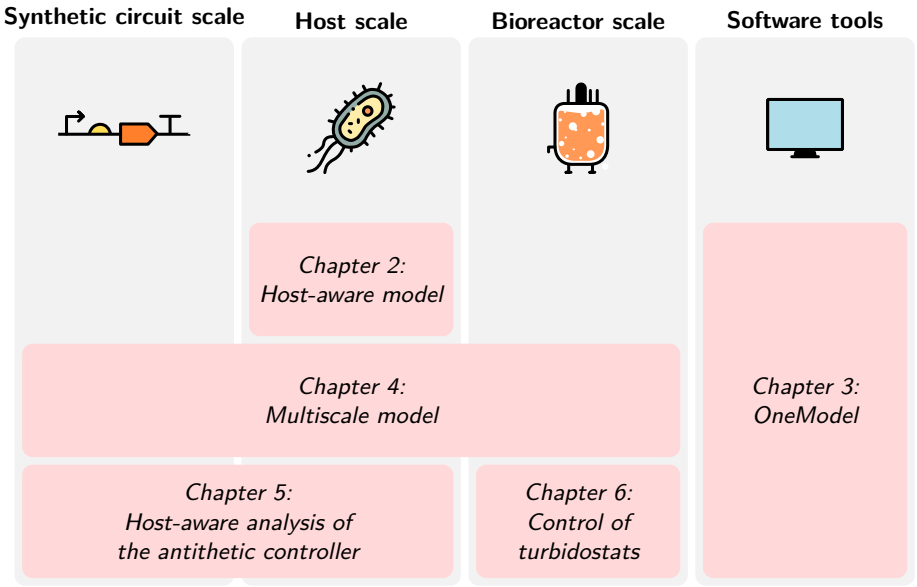

Figure 1.1: Schematic arrangement of the thesis chapters in relation to the main topics. The columns correspond to the four main topics of the thesis: synthetic circuit scale, host scale, bioreactor scale, and software tools. Each chapter covers one or more of the thesis topics. Chapter 3 is extended vertically to highlight its use in each of the other chapters.

to analyze the trade-offs in constitutive protein expression systems, paying attention to the effect of promoter and RBS (ribosome binding site) strengths on macroscopic titer, productivity (rate), and yield.

Constitutive protein expression is sensitive to perturbations and circuit parameters (as shown in Chapter 4), potentially leading to not robust bioproduction. The antithetic controller is a biomolecular integral feedback controller that regulates protein expression and increases bioproduction robustness. Chapter 5 analyzes the antithetic controller considering realistic conditions using the host-aware model, and in addition, this Chapter exemplifies the usefulness of the host-aware model and *OneModel*. The results show host-circuit unexpected interactions, which can be relevant to the practical implementation of the antithetic controller.

Industrial bioproduction is done in large-size fedbatch bioreactors because it is the most productive of the options (as we show in Chapter 4). However, small-size continuous bioreactors (turbidostats) are very useful for laboratory characterization of synthetic circuits. Chapter 6 focuses on the development of control laws for turbidostats.

Finally, once all the chapters are presented, the overall conclusions of the Thesis are drawn in Chapter 7. Appendix A includes the complete theoretical development of the host-aware model in Chapter 2.

1.3 Contributions

The main contributions of this work are the following:

- **(Chapter 2)** Novel minimal host-aware model for *E. coli*. This model is able to predict the protein mass distribution and growth rate as a function of substrate availability and the cell burden of exogenous protein expression.
- **(Chapter 2)** Analysis of the different strategies used by the host to express endogenous proteins. It is explained why the host uses a high transcription rate and a low translation rate for expressing ribosomal proteins and the opposite strategy for non-ribosomal proteins. Finally, this phenomenon is also studied on exogenous proteins: which strategies are more convenient for bioproduction and their relationships with substrate variations.
- **(Chapter 3)** Development of *OneModel*. A Systems Biology Markup Language (SBML) modeling tool that is used in this Thesis to define most of the models.
- **(Chapter 3)** Development of *sbml2dae*. A software tool that generates simulation-ready Matlab implementations from SBML models. We have used it to perform most of the simulations in this Thesis.
- **(Chapter 4)** Novel multi-scale model that incorporates the dynamics of the bioreactor, the host cell, and the synthetic genetic circuits. Analysis of titer, yield, and productivity of an exogenous protein taking into account the host and bioreactor dynamics.
- **(Chapter 5)** Host-aware antithetical controller analysis. This analysis reveals interactions between the host and the antithetic controller that could not be predicted without using the host-aware model and may be crucial for implementation in practice.
- **(Chapter 6)** Novel control law for the regulation of the concentration of biomass in turbidostats, using using non-linear time-scaled linearization and its experimental validation.

1.4 Publications

Refereed Journal Papers

- De Battista, H. et al. "Output Feedback Linearization of Turbidostats After Time Scaling". In: *IEEE Transactions on Control Systems Technology* (2018), pp. 1–9. ISSN: 10636536. DOI: 10.1109/TCST.2018.2834882
- Santos-Navarro, F. N. et al. "Gene Expression Space Shapes the Bioprocess Trade-Offs among Titer, Yield and Productivity". In: *Applied Sciences* 11.13 (2021), p. 5859. ISSN: 2076-3417. DOI: 10.3390/app11135859
- Santos-Navarro, F. N. et al. "RBS and Promoter Strengths Determine the Cell-Growth-Dependent Protein Mass Fractions and Their Optimal Synthesis Rates". In: *ACS Synthetic Biology* 10.12 (2021), pp. 3290–3303. ISSN: 21615063. DOI: 10.1021/acssynbio.1c00131
- Boada, Y. et al. "Modeling and optimization of a molecular biocontroller for the regulation of complex metabolic pathways". In: *Frontiers in Molecular Biosciences* 9 (2022). DOI: 10.3389/fmolb.2022.801032

Conference Presentation and Posters

- Santos-Navarro, F. N. et al. "Reference Conditioning Anti-windup for the Biomolecular Antithetic Controller". In: *IFAC-PapersOnLine* 52.26 (Jan. 2019), pp. 156–162. ISSN: 2405-8963. DOI: 10.1016/J.IFACOL.2019.12.251
- Santos-Navarro, F. N., Navarro, J. L., and Picó, J. "SBModEns : A Modular Toolbox for Model Building , Reduction , Analysis and Simulation in System Biology". In: *Proceedings of 12th IWBDA* (2020), pp. 49–50. URL: https://www.iwbdaconf.org/2020/docs/IWBDA2020Proceedings.pdf
- Santos-Navarro, F. N. and Picó, J. "Minimal model for protein expression accounting for metabolic burden". In: *Proceedings of 12th IWBDA* (2020), pp. 41–42. URL: https://www.iwbdaconf.org/2020/docs/IWBDA2020Proceedings.pdf
- Santos-Navarro, F. N. et al. "OneModel: an open-source SBML modeling tool focused on accessibility, simplicity, and modularity". In: *DYCOPS* (2022), accepted for publication

Preprint publications

The following publication was a preprint of [100]:

- Santos-Navarro, F. N. and Picó, J. "Resources allocation explains the differential roles of RBS and promoter strengths in cell mass distribution and optimal protein expression productivity". In: *bioRxiv* (2020). DOI: 10.1101/2020.11.19.390583. arXiv: 2020.11.19.390583

1.5 GitHub repository

Most of the *OneModel* and *Matlab* code that we have used to generate the simulations and figures in this Thesis is available in a GitHub repository at `https://github.com/sb2cl/thesis_fernandonobel`. For more information, read the `README.md` file in the repository.

Chapter 2

Host-aware model

> "This is the art of 'toy models' in physics: the belief that a few simple equations can capture some essence of a natural phenomenon."
>
> —Uri Alon, *An Introduction to Systems Biology*

The contents of this chapter appeared in the following journal publication:

- Santos-Navarro, F. N. et al. "RBS and Promoter Strengths Determine the Cell-Growth-Dependent Protein Mass Fractions and Their Optimal Synthesis Rates". In: *ACS Synthetic Biology* 10.12 (2021), pp. 3290–3303. ISSN: 21615063. DOI: 10.1021/acssynbio.1c00131

2.1 Abstract

Models of gene expression considering host-circuit interactions are relevant for understanding both the strategies and associated trade-offs that cell endogenous genes have evolved and for the efficient design of heterologous protein expression systems and synthetic genetic circuits. Here, we consider a small-size model of gene expression dynamics in bacterial cells accounting for host-circuit interactions due to limited cellular resources. We define the cellular resources recruitment strength as a key functional coefficient that explains the distribution of resources among the host and the genes of interest and the relationship between the usage of resources and cell growth. This functional coefficient explicitly takes into account lab-accessible gene expression characteristics, such as promoter and

ribosome binding site (RBS) strengths, capturing their interplay with the growth-dependent flux of available free cell resources. Despite its simplicity, the model captures the differential role of promoter and RBS strengths in the distribution of protein mass fractions as a function of growth rate and the optimal protein synthesis rate with remarkable fit to the experimental data from the literature for *E. coli*. This allows us to explain why endogenous genes have evolved different strategies in the expression space and also makes the model suitable for model-based design of exogenous synthetic gene expression systems with desired characteristics.

2.2 Introduction

The interrelations among the cell environment from which the cell uptakes substrates, its metabolism, and the engagement of cell resources needed for gene expression result in host-circuit interactions between gene circuits and their cell host. These interactions induce competition for common shared cell resources affecting gene expression and cell growth. Endogenous genes have evolved different strategies to deal with the problem of optimal protein expression under different needs and cell conditions [39]. As for exogenous genes, one of the fundamental problems in the rational design of synthetic genetic circuits of increasing complexity, partly explaining the current disparity between the ability to design biological systems their actual experimental performance, is the lack of systematic design methods considering the host-circuit interaction [18]. Cells have reached an optimal use of their resources during evolution. The over-expression of exogenous genes by a genetically modified microorganism as well as the production of metabolites by the addition and/or modification of their metabolic pathways introduces a cell burden that takes the microorganism off its natural state [117]. The resulting competition for common shared cell resources affects cell growth and introduces spurious dynamics [89] leading to problems of malfunctioning of the synthetic circuit. It also triggers its elimination by evolutionary mechanisms trying to restore the natural optimal state [65].

Mathematical models of gene expression accounting for cellular resources competition can be used to decipher the mechanisms underneath gene expression strategies that have evolved to optimize different criteria. Not only this is useful to understand natural systems but also addresses the rational design of synthetic genetic circuits. Therefore, in the last years there has been an increasing interest in the development of models and methods for model-based design of synthetic gene circuits considering host-circuit interactions [93].

The simplest burden-aware models deal with the interactions among genes in a gene network and consider shared cell resources as an external source, without considering the host behavior. This approach has proved very useful to deal with the so-called retroactivity [49], the burdening interaction among circuit modules and host originated from mass exchange. Retroactivity poses problems when predicting the behavior of a large network from that of the composing modules. It is a problem analogous to modeling the coupling between electrical circuits connected to a real energy source. Thus, the models accounting for it somewhat resemble Ohm's law [17, 89]. As these models do not explicitly consider the host behavior they cannot be easily used within a multi-scale framework integrating synthetic circuits of interest, host, and cell environment at the macroscopic level.

Alternatively, one may develop models relating substrates uptake, cell growth rate, and availability of free resources as a function of the gene circuits demand. These range from very coarse-grain ones [105, 6, 38, 14] to semi-mechanistic ones with varied degrees of granularity [125, 50, 66]. In this last case, the interplay between circuit, host and environment can be directly incorporated into the circuit model of interest to capture the impact of cellular trade-offs and resource competition on the circuit function.

Construction of a large-scale mechanistic model of *E. coli*has enabled to integrate and cross-evaluate a massive, heterogeneous dataset integrating measurements reported by various groups over decades [66]. On the other hand, medium-size detailed mechanistic models like the one developed in [125] have been used to study behavioral modulations of a gene switch [7] or a feedforward circuit [8, 34]. These medium and large-scale models, though very useful, are most often over-parametrized and cannot easily be integrated within a user-friendly and lightweight computational framework for model-based circuit design.

The small-size model presented here has enough granularity to provide good predictions of the host dynamics, the expression of the genes of interest and their interactions while having a small number of parameters to be estimated. We derived the dynamics of gene expression as a function of the fraction of free ribosomes relative to available mature ones considering protein synthesis on polyribosomes. We also defined the gene *resources recruitment strength* as the key functional coefficient that allowed us to explain the distribution of resources among the host and the genes of interest and the relationship between the use of resources and cell growth. An additional goal was to provide a model useful for model-based circuit design purposes. To this end, the model considers explicitly lab-accessible gene expression characteristics such as promoter and RBS strengths. We derived the equivalence between the relative resources recruitment strength and the relative mass fraction of a given protein at steady-state. From this equivalence, the

protein synthesis rate can be easily evaluated using the average host dynamics at steady-state. We used experimental data from the literature to estimate the average resources recruitment strength for both ribosomal and non-ribosomal proteins in *E. coli*. This allowed us to evaluate how the sensitivity of the resources recruitment strength to RBS and promoter can explain the variation of the cell protein mass fractions with growth rate and the differential roles they play. These data also can show how host-circuit interaction shapes the optimal abundance rates of both endogenous and exogenous proteins in the expression space.

2.3 Results

2.3.1 Burden-aware model of gene expression dynamics.

Our model considers, on the one hand, the dynamics of the expression of the cell host endogenous protein-coding genes. These are the genes that contribute to cell growth (Figure 2.1A). On the other hand, the model allows the possibility of considering the expression of protein-coding exogenous genes (Figure 2.1C). These do contribute to cell mass, but not to cell growth, akin to the consideration of unproductive proteins used in [105].

We started by modeling the polysomic gene expression dynamics for a generic k-th protein-coding gene in prokaryote cells. We considered that transcription is faster than translation so it can be assumed at steady-state, and that ribosomes are the limiting shared resource required for protein expression (see Appendix Section A.1 in the appendices). Under these assumptions, we derived the dynamics for the number of molecules of a k-th protein as:

$$\dot{p}_k = \frac{\nu_t(s_i)}{l_{pk}} J_k(\mu, r) r - (d_k + \mu) p_k \,, \tag{2.1}$$

where p_k is the number of copies of the k-th protein, l_{pk} is the protein length expressed as equivalent number of amino acids, d_k the protein degradation rate constant, μ the cell specific growth rate, r the number of free ribosomes and $\nu_t(s_i)$ is the substrate-dependent effective peptide chain elongation rate. This one is expressed using the Michaelis-Menten expression:

$$\nu_t(s_i) = \nu \frac{s_i}{K_{sc} + s_i} \,, \tag{2.2}$$

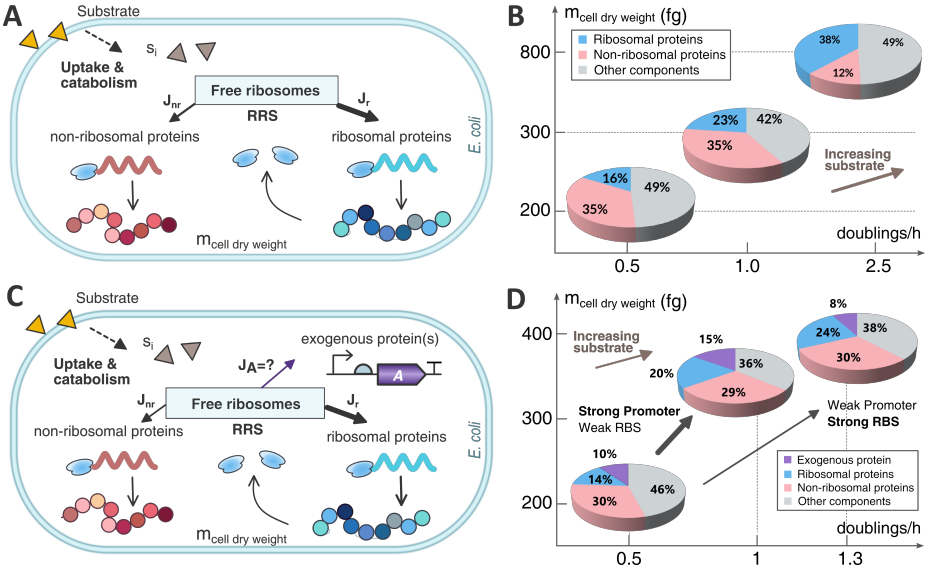

Figure 2.1: **A, C:** Schematic view of the partitioned use of cell resources (ribosomes) to synthesize both ribosomal and non-ribosomal proteins. The ribosomal proteins generate functional ribosome molecules to serve as the fundamental resource for protein synthesis. The *resources recruitment strength* (RRS) coefficient explains how the cell resources are allocated among the host endogenous genes (**A**), and both endogenous and exogenous genes of interest (**C**) (eg. exogenous protein(s) expressed by a synthetic genetic circuit). **B:** Resource allocation in a host cell in terms of the fractions of cell dry weight for the ribosomal proteins, non-ribosomal ones and other components. The pie charts represent different resources allocation scenarios, with rising growth rates when the available substrate is increasing in the x-axis, and the resulting cell dry weight in the y-axis. **D:** Resource allocation for a host cell expressing an exogenous protein. Two strategies were used for expressing the exogenous protein: strong-promoter with weak-RBS and weak-promoter with strong-RBS. The pie charts show the resource allocation distribution for both strategies (cell dry weight in the y-axis) for different growth rates (x-axis) caused when the availability of substrate is increased. Both strategies start from the same mass distribution at $0.5\,\text{doublings} \cdot \text{h}^{-1}$ (they share the same pie chart). The substrate was increased in the same quantity for both strategies. The arrows point to the resulting mass distribution pie chart for each strategy.

where ν is the maximum attainable peptide synthesis rate and K_{sc} is a Michaelis-Menten parameter related to the cell substrate uptake and catabolic capacity. As a first approximation, we considered that ν is organism dependent but does not depend on the sequence of nucleotides, (ie. on the particular gene being expressed) and K_{sc} is organism and substrate dependent but does not depend on the nucleotides sequence either.

The term $J_k(\mu, r)$, defined as the *resources recruitment strength*, is a dimensionless function of the growth rate μ and the number of free mature ribosomes r that quantifies the capacity of a k-th gene to engage cellular resources to get expressed (Figure 2.1A). It is the key functional coefficient in our model that explains the distribution of resources among the host and the genes of interest and the relationship between the usage of resources and cell growth (see its derivation in the Appendix A.1). Besides depending on the cell growth rate and the availability of cell resources, the resources recruitment strength is a function of the promoter and RBS strengths. For a generic protein-coding gene, its resources recruitment strength is defined as:

$$J_k(\mu, r) \triangleq E_{mk}(l_{pk}, l_e)\omega_k(T_f)\frac{1}{\frac{d_{mk}}{K^k_{C^0}} + \mu r}\,. \tag{2.3}$$

On the one hand, the *resources recruitment strength* $J_k(\mu, r)$ depends on the availability of cell resources: the flux of free ribosomes μr and the ribosomes density-related term $E_{mk}(l_{pk}, l_e)$. This one is obtained (see Appendix A.1) as:

$$E_{mk}(l_{pk}, l_e) \triangleq \frac{l_{pk}}{l_e}\left[1 - \left(\frac{l_{pk}}{l_{pk} + l_e}\right)^{\frac{l_{pk}}{l_e}}\right], \tag{2.4}$$

where $1/l_e$ is the specific ribosomes density, with l_e expressed as equivalent number of codons. The ribosomes density can be estimated as an inversely log-linear function of the protein length l_{pk} (see Equation (A.94) in Appendix A.13). Interestingly, E_{mk} can accurately be approximated as $E_{mk}(l_{pk}, l_e) \approx 0.62 l_{pk}/l_e$ (ie. a linear function of the number of ribosomes elongating along the transcript) for a wide range of values of l_{pk} and l_e (see Figure A.2).

On the other hand, the resources recruitment strength $J_k(\mu, r)$ also depends on gene expression characteristics: mRNA transcription rate $\omega_k(T_f)$ or the promoter strength, mRNA degradation rate constant d_{mk} and the effective ribosome binding site (RBS) strength $K^k_{C^0}(s_i)$. This one is defined as:

$$K^k_{C^0}(s_i) \triangleq \frac{K^k_b}{K^k_u + K_e(s_i)}\,, \tag{2.5}$$

where K^k_b and K^k_u are respectively the association and dissociation rate constants between a free ribosome and the RBS, and $K_e(s_i) = \nu_t(s_i)/l_e$ is the transla-

tion initiation rate constant, which depends on the availability of intracellular substrate (see Appendix A.1 for details).

The resources recruitment strength of a given protein-coding gene can be related with its translation rate and number of transcripts. Consider the standard dynamic model for the expression of a protein p [42]:

$$\dot{p} = \frac{\beta_p \beta_m}{d_m} - \mu p \,, \tag{2.6}$$

where β_m (mRNA/t) is the transcription rate, β_p (protein/(mRNA·t)) the translation one and d_m the mRNA degradation rate constant. Comparing with (2.1) we derived the relationship:

$$J_k = \frac{l_{pk}}{\nu_t(s_i)} \frac{\beta_p \beta_m}{d_m} \frac{1}{r} \,. \tag{2.7}$$

The expression (2.7) allows to calculate the theoretical maximum resources recruitment strength, $J_k|_{r=1}$, from available experimental data (see Appendix A.12).

The dynamics of the total number of ribosomes can be obtained by considering an analogous expression to (2.1) for each of the N_r proteins forming up a ribosome (see Appendix Section A.2). The total number of ribosomes in the cell at any one time instant, r_T, is the sum of the mature (r_a) and inmature (r_i) ones. In turn, the mature ribosomes r_a available for translation comprise the free ribosomes r and the ones already bound either to the RBSs or elongating along the transcripts. The number of available mature ribosomes is a fraction of the total number of ribosomes, so that $r_a = \Phi_m r_T$. We assumed the fraction Φ_m varies little in time (see Appendix Section A.2) so that the dynamics of the total number of ribosomes and that of the number of available ribosomes are the same but for a scale factor. Without loss of generality, we considered average protein-coding endogenous genes with resources recruitment strengts J_r and J_{nr} respectively. This allowed us to obtain the relationship between free and total number of ribosomes (see Appendix Section A.3) as:

$$r = \frac{r_a}{1 + \mathrm{WSum}(\mu, r)} \,, \tag{2.8}$$

with:

$$\begin{aligned}\mathrm{WSum}(\mu, r) = N_r \left[1 + \frac{1}{E_{mr}}\right] J_r(\mu, r) + N_{nr} \left[1 + \frac{1}{E_{mnr}}\right] J_{nr}(\mu, r) \\ + \sum_{k=1}^{N_{\mathrm{exo}}} \left[1 + \frac{1}{E_{mk}(l_{pk}, l_e)}\right] J_k(\mu, r)\,,\end{aligned} \tag{2.9}$$

where N_r and N_{nr} are the number of ribosomal and non-ribosomal protein-coding endogenous genes respectively, and N_{exo} allows for the existence of exogenous genes.

Cell growth can essentially be explained as the time variation of the protein fraction of the total cell mass (Figure 2.1B). Yet, not all protein mass contributes to cell growth. There are proteins which may be undergoing active degradation while other proteins, for example the exogenous ones, will not have any active role positively contributing to the cell growth. Therefore, we considered only the endogenous ribosomal and non-ribosomal proteins to computed the cell specific growth rate. We used the dynamics (2.1) and assumed an average amino acid mass m_{aa} to obtain the times variation of the total endogenous protein mass content m_h of the native *host* cell (see Appendix Section A.5):

$$\dot{m}_h = m_{aa} \nu_t(s_i) \Phi_t^h \Phi_m r_T - \mu m_h\,, \tag{2.10}$$

where:

$$\Phi_t^h = \frac{N_r J_r(\mu, r) + N_{nr} J_{nr}(\mu, r)}{1 + \mathrm{WSum}(\mu, r)} \tag{2.11}$$

is the fraction of ribosomes elongating along endogenous ribosomal and non-ribosomal transcripts relative to the total number of mature available ribosomes (see Appendix Section A.4).

Next, we considered that the total biomass dry weight variation (ie. that of the whole population of cells) is mainly caused by cell duplication (i.e. population growth), and the dynamics of cell mass accumulation are much faster than those of cell duplication. Under this assumption, the protein mass for each cell quickly reaches steady-state ($\dot{m}_h \approx 0$). Thus, from equation (2.10) we obtained the expression for the cell specific growth rate:

$$\mu(s_i) = \frac{m_{aa}}{m_h}\nu_t(s_i)\Phi_t^h\Phi_m r_T\,, \tag{2.12}$$

where notice $\Phi_t^h\Phi_m r_T$ is the number of ribosomes actively elongating along endogenous transcripts at a given time instant (see Appendix Sections A.4 and A.5).

To relate the growth rate $\mu(s_i)$ obtained from the intracellular dynamics with the extracellular measure of growth rate, $\mu(s)$, derived from cell population dynamics, we followed a reasoning derived from the model developed in [125], where the quantity of intracellular substrate s_i is related to the one of extracellular substrate s through the dynamics of nutrient import and catabolism (see Appendix Section A.6 for details). Under the condition of steady-state growth where the rate of total cell-mass growth is identical to the rate of cell number growth [128] and assuming that the maximum import and catabolism fluxes are balanced, we obtained the Monod population growth kinetics:

$$\mu(s) = \frac{\frac{m_{aa}}{m_h}\nu\Phi_t^h\Phi_m r_T s}{\frac{k_t}{V_m} + s}\,, \tag{2.13}$$

where V_m is a parameter related to the effective volume of culture broth for each cell and k_t is a Michaelis-Menten constant for substrate transport into the cell. Notice we recuperate the maximum specific growth rate μ_m as a linear function of the number of ribosomes actively elongating along transcripts at a given time instant. Finally, the Monod constant K_s as a function of the substrate transport capacity and the cell harvesting volume.

Our model accounts for the protein mass distribution (Figure 2.1B) but does not consider the relationship between growth rate and the total cell protein mass. Cells growing at faster growth rates are larger and heavier, thus affecting their total protein content [108]. To model the relationship between the cell protein content and the specific growth rate for the native host cell —ie. $m_h = m_h(\mu)$— we postulated the relationship:

$$\frac{dm_h}{d\mu} = \beta m_h\,, \tag{2.14}$$

with $m_h(0) = 77.375$ (fg) and $\beta = 61.781$ (min) as best fits obtained for *E. coli*cells using the data in [15]. We also considered an analogous model relating

the cell dry weight $m_{h,cDW}(\mu)$ with the growth rate (see Appendix Section A.7 for details).

Finally, we structured our model in such a way that it can be used to analyze the resource allocation trade-offs (see Figure 2.1D) among the endogenous protein-coding genes from the native *E. coli*host cell, and a given set of exogenous ones of interest (eg. a synthetic gene circuit). In the later case we have considered, without losing generality, a single exogenous protein of interest A to exemplify the model expressions and the interaction between the host cell and the exogenous additions.

For the endogenous protein-coding genes we considered the ensemble of ribosomal and non-ribosomal genes as lumped species with average values of $E_{mr}(l_p^r, l_e)$, $E_{mnr}(l_p^{nr}, l_e)$; and $J_r(\mu, r)$, $J_{nr}(\mu, r)$, respectively. Then we obtained the dynamics of the total mass of ribosomes m_{r_T} and non-ribosomal endogenous proteins m_{nr}, and the dynamics of the mass m_A of the exogenous protein (see Appendix Sections A.8 and A.9 for details) as:

$$\dot{m}_{r_T} = \mu \left[m_h(\mu) \frac{N_r J_r(\mu, r)}{N_r J_r(\mu, r) + N_{nr} J_{nr}(\mu, r)} - m_{r_T} \right], \tag{2.15a}$$

$$\dot{m}_{nr} = \mu \left[m_h(\mu) \frac{N_{nr} J_{nr}(\mu, r)}{N_r J_r(\mu, r) + N_{nr} J_{nr}(\mu, r)} - m_{nr} \right], \tag{2.15b}$$

$$\dot{m}_A = \mu \left[m_h(\mu) \frac{N_A J_A(\mu, r)}{N_r J_r(\mu, r) + N_{nr} J_{nr}(\mu, r)} - m_A \right], \tag{2.15c}$$

with:

$$J_r(\mu, r) = E_{mr} \omega_r \frac{1}{\frac{d_m^r}{K_{C^0}^r(s_i)} + \mu r}, \tag{2.16a}$$

$$J_{nr}(\mu, r) = E_{mnr} \omega_{nr} \frac{1}{\frac{d_m^{nr}}{K_{C^0}^{nr}(s_i)} + \mu r}, \tag{2.16b}$$

$$J_A(\mu, r) = E_{mA}(l_{pA}) \omega_A \frac{1}{\frac{d_m^A}{K_{C^0}^A} + \mu r}, \tag{2.16c}$$

where N_A is the gene copy number of A, the number of free ribosomes r is obtained using (2.8) and the specific growth rate μ is calculated using (2.12).

The denominator in the fraction of resources recruitment strengts only includes the host protein-coding genes. The protein mass $m_h(\mu)$ is that of the native host cell, comprising only the cell endogenous proteins. We defined the mass of the strain $m_s = m_h + m_A$ as the one comprising the mass of the host and the one of the exogenous proteins. We obtained the relation between the protein mass of the strain $m_s(\mu)$ and that of the native host $m_h(\mu)$ (see Appendix Section A.9) as:

$$m_s(\mu) = \frac{\Phi_t^s}{\Phi_t^h} m_h(\mu) = \frac{N_r J_r + N_{nr} J_{nr} + N_A J_A}{N_r J_r + N_{nr} J_{nr}} m_h(\mu)\,. \tag{2.17}$$

In addition, we also considered the cell dry weight $m_{cDW}(\mu)$, comprising the mass $m_h(\mu)$ of the endogenous ribosomal and non-ribosomal proteins, the mass of the exogenous proteins $m_A(\mu)$ and the mass of other constituents of the cell, denoted as $m_Q(\mu)$. Thus, $m_{cDW}(\mu) = m_h(\mu) + m_A(\mu) + m_Q(\mu) = m_s(\mu) + m_Q(\mu)$. To obtain $m_Q(\mu)$ we used the estimation of the cell dry weight $m_{h,cDW}(\mu)$ for the *E. coli*host native cell, i.e without expression of exogenous genes (see Appendix Section A.7 for details), assuming that $m_Q(\mu)$ does not depend on the expression of exogenous genes. This allowed us to estimate the mass fractions with respect to the total cell dry weight.

To evaluate the productivity rate of a given protein of interest, we obtained its mass synthesis rate as the steady-state mass of protein produced per cell and generation (see Appendix Section A.9). In the case of an exogenous protein A and using (2.15c)–(2.17), we obtained:

$$\Pi_A \triangleq m_{A,\mathrm{ss}}\mu = m_s(\mu) \frac{N_A J_A(\mu, r)}{N_r J_r(\mu, r) + N_{nr} J_{nr}(\mu, r) + N_A J_A(\mu, r)} \mu\,. \tag{2.18}$$

We defined the specific mass synthesis rate relative to the cell dry weight as:

$$\begin{aligned} \pi_A &\triangleq \frac{\Pi_A}{m_{cDW}(\mu)} \\ &= \frac{m_s(\mu)}{m_{cDW}(\mu)} \frac{N_A J_A(\mu, r)}{N_r J_r(\mu, r) + N_{nr} J_{nr}(\mu, r) + N_A J_A(\mu, r)} \mu\,. \end{aligned} \tag{2.19}$$

For a given protein A, both the protein mass synthesis rate ($\mathrm{g} \cdot \mathrm{min}^{-1}$) and the specific one ($\mathrm{g} \cdot \mathrm{min}^{-1} \cdot \mathrm{gCDW}^{-1}$) are directly related to its relative resources recruitment strength fraction.

From the results above we obtained the cell specific growth rate at steady-state exponential balanced growth as:

$$\mu_{ss}(s_i) = \frac{m_{aa}}{m_{rib}} \nu_t(s_i) \Phi_m \Phi_t^r \,, \tag{2.20}$$

where m_{rib} is the average ribosome mass (see Appendix Section A.9 for details). That is, the cell growth rate at steady-state depends linearly on the fraction $\Phi_t \Phi_r^t$ of bound ribosomes being actively used to build up ribosomes themselves (ie. ribosomes actively elongating along and translating ribosomal transcripts) relative to the total number of ribosomes.

2.3.2 Ribosomal and non-ribosomal genes differ in their average resources recruitment strength

Using the experimental data in [42], we evaluated the maximum expected magnitude of the resources recruitment strength for each gene using equation (2.7) with $r = 1$, ie. the theoretical maximum resources recruitment strength for a given availability of intracellular substrate, $J_{k,\max} = J_k|_{r=1,\nu_t(s_i)}$. The data in [42] corresponds to *E. coli* cells in fast growing conditions (doubling time $t_d = 21.5$ min). Therefore, we could assume saturation of substrate, allowing us to consider the maximum substrate-dependent effective translation elongation rate $\nu_t(s_i) = \nu$ to evaluate (2.7). Notice this is equivalent to estimating the maximum resources recruitment strength for the maximum specific growth rate. *E. coli* has around 4225 protein-coding genes [77, 52]. From [42] we got information for a representative enough set of genes, comprising 3551 non-ribosomal and 68 ribosomal ones, accounting for around 86% of all *E coli* genes.

Firstly, the results allowed us to estimate of the order of magnitude of the resources recruitment strength for ribosomal and non-ribosomal genes in *E coli* and their maximum average value. Then, we obtain how many genes of each class are active at any one time.

As expected, the values obtained spanned several orders of magnitude. For the ribosomal genes, the average value $J_{\max,r}^{\text{avg}} = 124.5$ and a coefficient of variation $CV_{J\max,r} \approx 1$, while for the non-ribosomal ones the values were $J_{\max,nr}^{\text{avg}} = 3.78$ and $CV_{J\max,nr} \approx 6$. The average maximum resources recruitment strength for the ribosomal genes was two orders of magnitude higher than for non-ribosomal ones. Yet, the coefficient of variation was much smaller for the ribosomal resources recruitment strengths than for the non-ribosomal ones. Figure 2.2 shows the values of $J_{k,\max}$ we obtained for each gene sorted by the log-magnitude of the

ratio between the maximum resources recruitment strength and the length of the associated protein. The results did not essentially change from the non-normalized ones (see Figure 2.2). That is, the resources recruitment strength of *E. coli* genes is not fundamentally determined by the lengths of the proteins they code. This suggests that factors such as the effective transcription and translation rates are more relevant.

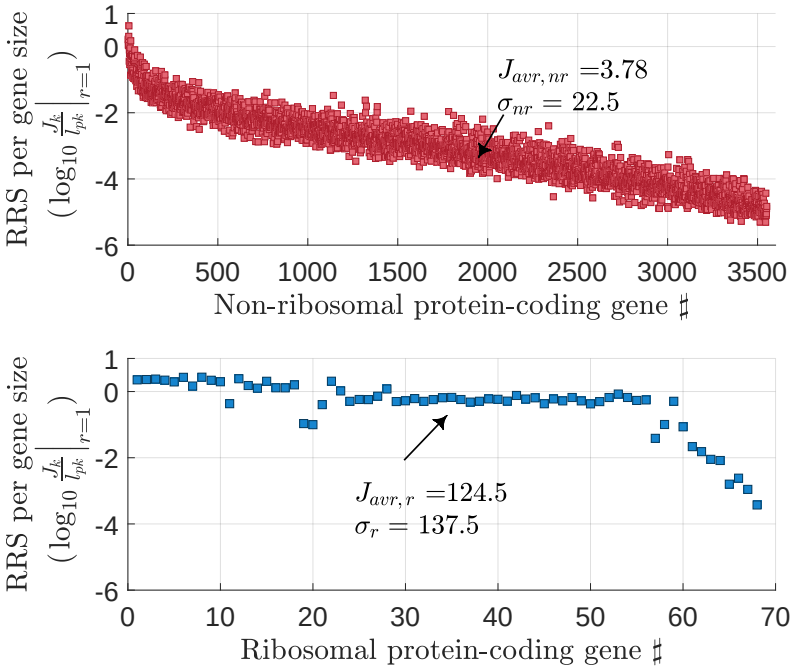

Figure 2.2: Log-magnitude of the ratio between the maximum resources recruitment strength and the length (aa) of the associated protein for the set of non-ribosomal (top) and ribosomal (bottom) protein-coding genes in [42]. The genes were ordered by decreasing value of the ratio.

But not all genes are expressed all the time. As a proxy to estimate how many genes are active at any given time, we calculated the cumulative sum of the maximum resources recruitment strength and obtained how many genes being expressed are required to explain both 95% and 99% of the total cumulative sum (see Figure A.8). We did this independently for both ribosomal and non-ribosomal proteins. Our results showed that out of the 68 ribosomal genes, 49 of them (72%) explained 95% of the cumulative sum of the maximum resources recruitment strength. To explain 99% we needed 57 ribosomal genes (84% of

them). However for non-ribosomal genes, we needed 875 out of 3551 genes (25%) to explain 95% of the cumulative sum and 1735 (49%) to explain the 99% .

2.3.3 The resources recruitment strength explains the distribution of ribosomal and non-ribosomal protein mass fractions

The relative mass fractions of ribosomal and non-ribosomal proteins in the cell depend on the cell growth rate, so that the ribosome content increases linearly with growth rate [15, 105, 74, 21]. Existing resource allocation models explain this as a result of optimal allocation of cell resources between the ribosomal and non-ribosomal fractions, balancing the demands of protein synthesis and nutrient influx under the constraint that the sum of both fractions remain constant [105]. In our model, the relative resources recruitment strength of a given protein equals its relative mass fraction in the cell at steady-state balanced growth (see equations (2.15a)–(2.15c) and Appendix Section A.8). Therefore, the relative distribution of mass between ribosomal and non-ribosomal proteins must be reflected in the relative distribution of their resources recruitment strengths.

We first studied the *E. coli* host cell, ie. without any exogenous protein-coding genes. We used the data in [15] to check whether our model was able to predict the linear increase of ribosomes content with growth rate and the relative distribution of endogenous ribosomal and non-ribosomal protein mass fractions as a function of growth rate. We did not estimate the model parameters to try and directly fit the experimental relative distribution of resources recruitment strengths, as this would not inform on the capability of the model to capture the intrinsic relationship among growth rate, use of cell resources or distribution of protein mass fractions. Instead, we analyzed if a good fit of the specific growth rate implied our model could generalize and predict the relative mass fractions in the cell. This, in turn, implies fitting the ribosomal and non-ribosomal resources recruitment strengths.

To this end, we fitted the model parameters using the experimental growth rate as output to predict. We used the values of the peptide chain elongation rates $\nu_t(s_i)$ as a function of growth rate available from [15] as the only input information given to the model. This is tantamount to feed the model only with the available amount of substrate s_i (see Appendix Section A.14). Then, we minimized the sum over the experimental data points of the absolute growth rate prediction error (see Appendix Section A.14). We considered the lumped resources recruitment strengths for both the ribosomal and non-ribosomal endogenous proteins (see expressions (2.15a)–(2.15b)) and estimated the fraction of mature ribosomes and the parameters corresponding to the RBS-strength and transcription rates. So

that would provide our model a good fit of the specific growth rate. The best fit estimated parameters are given in Table 2.1.

Table 2.1: Average best fit estimated values for *E. coli*of the RBS-strength related parameters K_b^k, K_u^k and transcription rates ω_k for ribosomal ($k = r$) and non-ribosomal ($k = nr$) proteins and the fraction Φ_t of mature ribosomes with respect to the total number of ribosomes.

Parameter	**Mean**	**Standard deviation**	**Units**
K_u^r	129.9	4.07	min^{-1}
K_u^{nr}	3.09	0.14	min^{-1}
K_b^r	5.57	0.78	$\text{molecule}^{-1} \cdot \text{min}^{-1}$
K_b^{nr}	12.86	1.50	$\text{molecule}^{-1} \cdot \text{min}^{-1}$
ω_r	5.65	0.29	$\text{mRNA} \cdot \text{min}^{-1}$
ω_{nr}	0.028	0.25×10^{-3}	$\text{mRNA} \cdot \text{min}^{-1}$
ϕ_m	0.90	0.5×10^{-2}	adimensional

The estimated values of the RBS-strength related parameters K_b^k, K_u^k implied ribosomal RBSs much weaker than the non-ribosomal ones. Interestingly, the values we obtained for the transcription rates were in the same order of magnitude as the mean values obtained from the data in [42] —$\omega_r = 2.4$ and $\omega_{nr} = 0.05$ respectively—. Therefore, this demonstrates a much higher value for the average transcription rate of ribosomal proteins than for the non-ribosomal ones. Our results also estimated an average high transcription-low translation rate expression strategy for the ribosomal endogenous genes and the opposite strategy for the non-ribosomal ones.

Figure 2.3A shows the results of the model parameter fitting and the good agreement between the experimental and the estimated growth rate. The estimation of the number of free ribosomes for cells growing at doubling time $t_d = 25$ minutes ($\mu \approx 0.028\,\text{min}^{-1}$) was consistent with the result $r \approx 350$ obtained using the experimental data in [42] (see Appendix Section A.15). For cells growing faster, the number of free ribosomes much increased. Notice though, that also the total number of ribosomes (both experimental and estimated) greatly increased for very fast growing cells. Thus, the fraction of free ribosomes with respect to the total number only increased from 0.08% up to 1.37% for cell doubling times between 100 and 24 minutes respectively (even though the number of free ribosomes varied by almost 200-fold). Similarly, the computed fraction Φ_m of mature ribosomes with respect to the total number of ribosomes was consistent with the estimated

fraction of active bound ribosomes $\Phi_t^h \Phi_m \approx 0.78$ (see Figure A.11) in agreement with [14, 15].

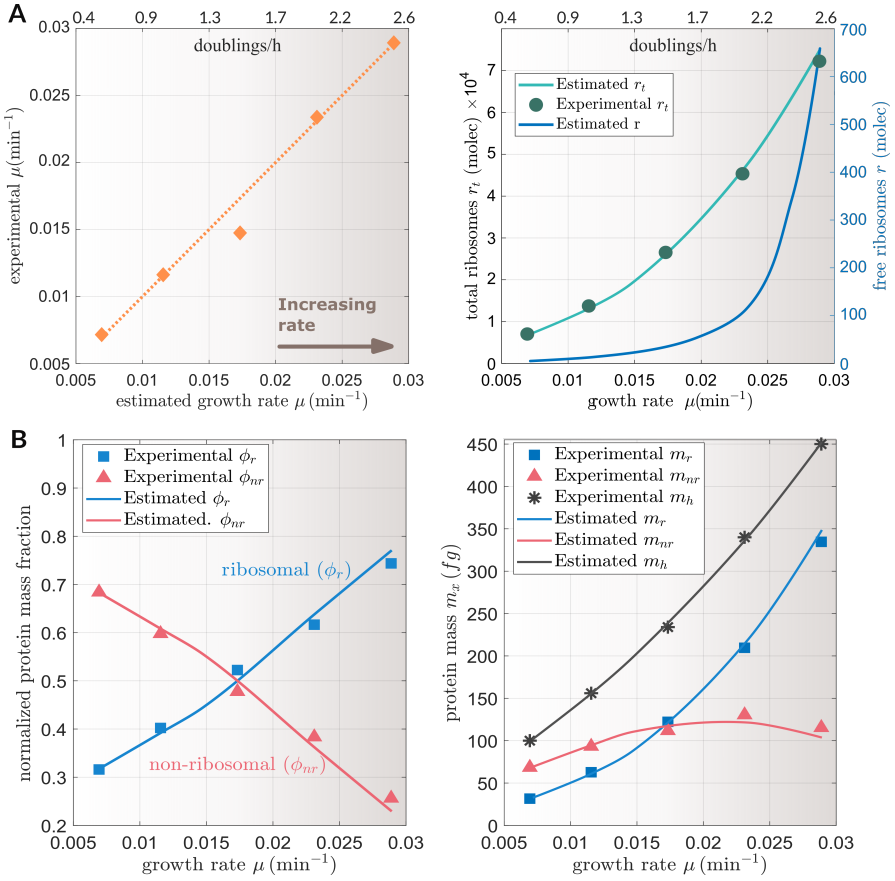

Figure 2.3: **A:** Estimated versus experimental growth rate (*left*). Experimental and estimated number of total ribosomes as a function of the growth rate and estimated number of free ribosomes (*right*). **B:** Estimated versus experimental mass fractions of ribosomal ($\phi_r = m_r/m_h$) and non-ribosomal ($\phi_{nr} = m_{nr}/m_h$) proteins in *E. coli* (*left*). Ribosomal (m_r), non-ribosomal (m_{nr}) and total host cell protein mass (*right*). In all plots the x-axis corresponds to the estimated and experimental growth rates evaluated for the range of peptide chain elongation rates $\nu_t(s_i)$ extracted from [15]

We evaluated the mass fractions of the endogenous ribosomal and non-ribosomal proteins at steady-state using the expressions (2.15a)–(2.15b). The model predictions were in very good agreement with the experimental values, as shown

in Figure 2.3B. Therefore, our model reproduced the known linear increase of the ribosomal fraction with growth rate. The differential behavior between the ribosomal and non-ribosomal resources recruitment strengths was behind the differential protein mass distribution as the cell growth rate increases.

The effective RBS strength used in our model is a function of the intracellular substrate because it varies with the cell growth rate according to equation (2.5). Figure 2.4 shows the estimated values as a function of the specific growth rate μ. The estimated effective RBS strength of the non-ribosomal protein-coding genes ($K^{nr}_{C^0}$) was much higher than that of the ribosomal ones ($K^{r}_{C^0}$). As the growth rate increased—tantamount in our model to an increasing intracellular substrate s_i—the ribosomal effective RBS strength kept almost constant (with a slight decrease around 12%) while the non-ribosomal one decreased by almost a 40%. We could explain this trend as result of the difference in the ratio between the transcript degradation rate and the RBS strength, $d_{mk}/K^{k}_{C^0}$ for both ribosomal and non-ribosomal genes. The ribosomal genes kept much higher values of $d_{mk}/K^{k}_{C^0}$ for all values of the flux of free resources μr. This, taking into account the monotonous increasing power-law relationship between the growth rate and the number of free ribosomes predicted by our model (see Appendix Section A.15) implies the observed trends in the values of the resources recruitment strength in Figure 2.4(*bottom*). The ribosomal resources recruitment strength $J_r(\mu, r)$ decreases much slower than that of the non-ribosomal ones as the growth rate increases.

Results 2.3.1 from endogenous genes, the steady-state is reached for balanced exponential growth when their relative fraction of resources recruitment strength equals their mass relative to that of the host cell. Since the ribosomal resources recruitment strength decreases much slower than the non-ribosomal one as the growth rate increases, the fraction of ribosomal resources recruitment strength with respect to the total sum of ribosomal and non-ribosomal resources recruitment stenghts will increase. As a consequence, its relative mass fraction will increase.

It is important to stress again that we estimated the parameters in our model so as to fit not the experimental mass fractions but the cell growth rate. By doing that, the internal structure of the model—substantiated in the structure of the resources recruitment strength functional coefficients—captured the correct differential mass distribution between ribosomal and non-ribosomal cell protein content as a function of growth rate.

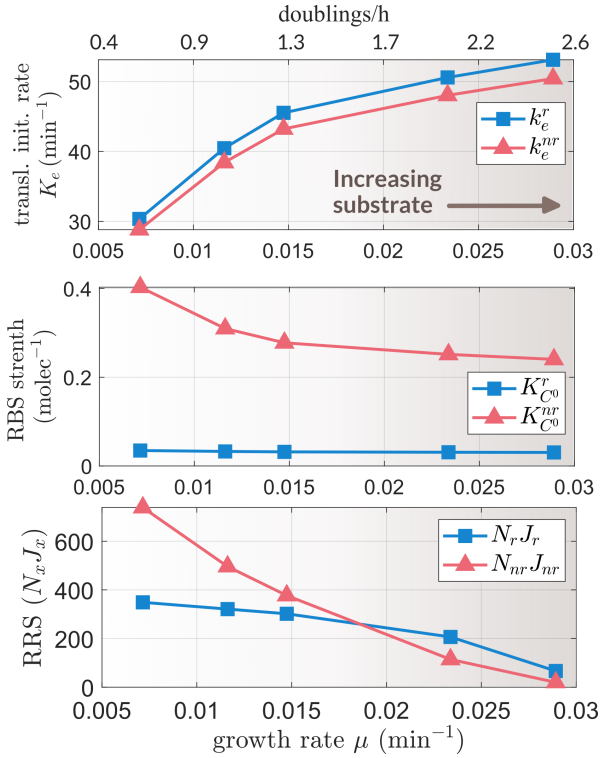

Figure 2.4: Estimated translation initiation rate k_e for the average ribosomal and non-ribosomal endogenous genes as a function of the specific growth rate μ (*top*). Estimated effective RBS strengths K_{C0}^r and K_{C0}^{nr} (*middle*). Estimated total resources recruitment strengths $N_r J_r$ and $N_{nr} J_{nr}$ as a function of growth rate μ (*bottom*).

2.3.4 Host-circuit interaction shapes the optimal synthesis rate of exogenous proteins

There are multiple ways to increase the expression of an exogenous protein of interest, including the choice of the expression vector of the synthetic gene circuit, optimizing the use of codons, co-expression of chaperones to aid protein folding, etc [92]. We focused on varying the expression space—ie. the gene induction space defined by the values of the mature mRNA synthesis rate and the effective RBS strength—at the same values of cell growth conditions and intracellular substrate availability. We used the average host dynamics at steady-state balanced growth to evaluate the distribution of cell mass fractions and the specific protein mass synthesis rate (specific synthesis rate for short or spMSR) of a given exogenous

protein of interest A as defined in (2.19) (see also Appendix Section A.9) as a function of variations in the expression space.

We first considered the RBS-strength related parameters K_b^A, K_u^A of the exogenous gene to be constant with values equal to the estimated averages for an endogenous non-ribosomal protein in *E. coli* (see Table 2.1) and only the mRNA synthesis rate was varied. To this end, we changed the gene copy number times the transcription rate (or promoter strength) $N_A\omega_A$ in the range $[10^{-1}, 10^5]$ times the average promoter strength of endogenous non-ribosomal genes given in Table 2.1. This gave a maximum value $N_A\omega_A \approx 3.3 \cdot 10^3$ mRNA $\cdot$ min^{-1}, which is an attainable value for *E. coli* considering an average transcription rate $\omega_A = 3\,\text{mRNA}\cdot\text{min}^{-1}$ and a high-copy number plasmid with $N_A = 1100$.

Figure 2.5A shows the variation across the mRNA synthesis space $N_A\omega_A$ of the mass fractions and the cell growth rate (Left) and the spMSR, π_A, of the exogenous protein (Right). The distribution of mass fractions was consistent with the behavior of the cell. As the mRNA synthesis rate of the gene A was increased (moving towards the right in the plot 2.5A-left), the mass fraction corresponding to the protein A also increased (purple) while that of ribosomal proteins decreased (blue) with a corresponding decrease in the cell growth rate (white line). Consequently, there appeared a maximum specific protein mass synthesis rate value (Figure 2.5A yellow dot, $\pi_A \approx 2.8 \cdot 10^3$ g $\cdot$ min^{-1} $\cdot$ gCDW^{-1}) which was achieved for a mRNA synthesis rate of 100 mRNA$\cdot$min^{-1}. This value represents a low-copy number plasmid $N_A \approx 20$ and a constitutive promoter with a transcription rate $\omega_A \approx 5$ mRNA $\cdot$ min^{-1}.

The model predicted an increasing mass fraction of the protein A as we continue increasing the value of the mRNA synthesis rate. However, this situation happens at the cost of reducing the fraction of ribosomal proteins, resulting in a very small growth rate. The relationship between the fraction of exogenous protein and growth rate in our model is a decreasing exponential (something consistent given its mathematical smooth differential continuous-time nature). Therefore, even if the zero growth rate is achieved in the limit for 100% of exogenous protein, notice this is a theoretical point only achieved in the limit, ie. at infinite cell doubling time. In practice, the cell viability will be lost before.

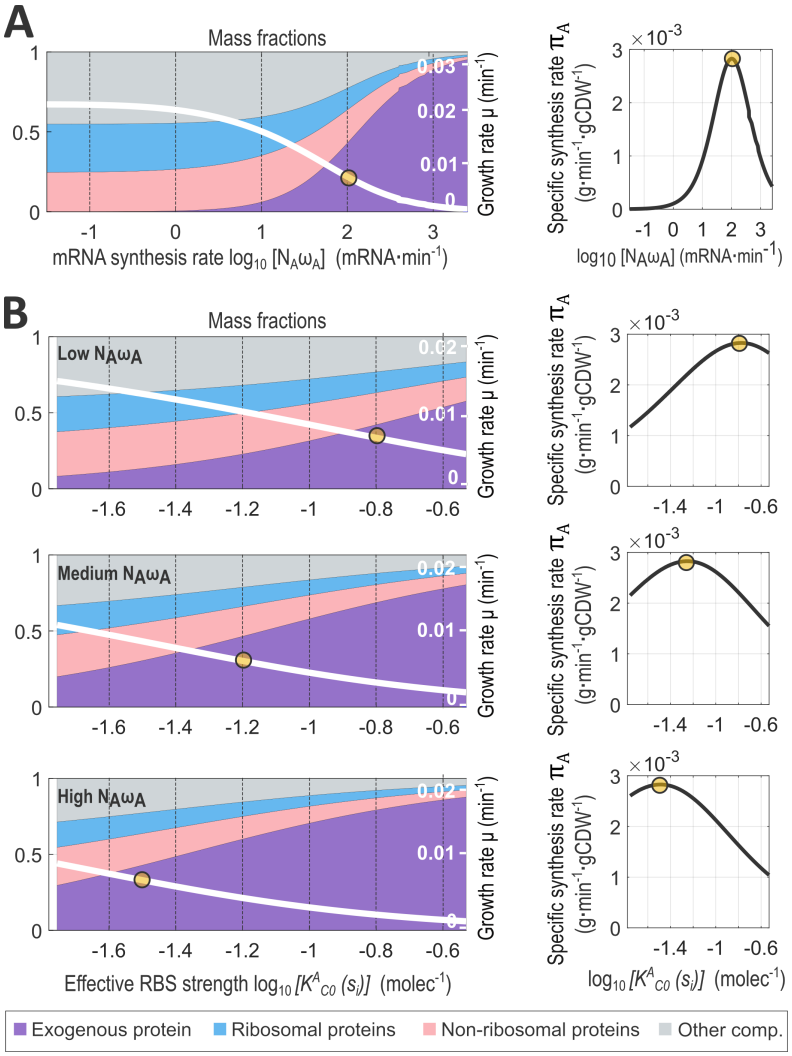

Figure 2.5: A: *Left.* Effect of increasing the mRNA synthesis rate of an exogenous protein A on the cell growth rate (*right*-axis) and on the cell mass fractions (*left*-axis). *Right.* Specific protein mass synthesis rate (spMSR) for the exogenous protein A as a function of its mRNA synthesis rate. Even though the growth rate decreases for increasing mRNA synthesis rates, the spMSR increases, reaching a maximum value (yellow dot) at fast mRNA synthesis rates around 2000 mRNA· min^{-1} and eventually decreases for larger mRNA synthesis values. **B:** *Left.* Effect of varying the RBS strength on the cell growth rate (*right*-axis) and the protein mass fractions (*left*-axis) for three increasing values of the mRNA synthesis rate (low, medium, high). *Right.* spMSR of the exogenous protein a a function of RBS-strength variation.

Figure 2.5B Left shows the results obtained when we analyzed three representative values of the mRNA synthesis rate $N_A\omega_A = \{150, 400, 800\}$ corresponding to an average transcription rate in *E. coli* ($\omega_A \approx 3$ mRNA $\cdot$ min^{-1}) combined with a low-, medium- and high-copy plasmid copy number respectively. Then, we varied K_b^A, K_u^A in the ranges considered in Appendix Section A.14 to obtain a range of values for the effective RBS-strength $K_{C0}^A(s_i)$. The mass fraction corresponding to protein A increased and the cell growth rate decreased for high levels of the RBS-strength. Figure 2.5B (Right) shows that the main factor affecting the spMSR is the mRNA synthesis rate $N_A\omega_A$. Thus, for low values of the mRNA synthesis rate, the spMSR increased for strong RBSs until a maximum appeared for one of the stronger ones (eg. $K_{C0}^A = 10^{-0.8} = 0.15$ molec^{-1}). For medium values of $N_A\omega_A$ there soon appeared a maximum spMSR for the exogenous protein as a function of the RBS-strength. Finally, for high values of the mRNA synthesis rate, increasing the RBS strength rapidly produced a decrease in the specific protein mass synthesis rate. Our model correctly predicted that there is a critical (optimal) protein synthesis rate that is achieved for lower RBS strength as the mRNA synthesis rate increases.

The location of the optimal spMSR as a function of variations in the full range of the expression space can be seen in Figure A.12, which shows the variation of the specific synthesis rate of the exogenous protein across the expression space $N_A\omega_A, K_{C0}^A(s_i)$ in log-log scale. The optimal subspace corresponded to a line in the log-log promoter-RBS space, showing the existence of a trade-off between the mRNA synthesis rate (tantamount to the gene induction) and the RBS strength. The pronounced slope of the optimal subspace explains the different sensitivity of the specific synthesis rate to the variations of either the promoter or the RBS strengths that were obtained in Figure 2.5A and B. Our model predicted that the specific synthesis rate is more sensitive to variations of the mRNA synthesis rate than to variations of the RBS strength. Thus, for intermediate values of $N_A\omega_A$, there is a wide range of RBS strengths that keep the specific synthesis rate close to its optimal value. This is also reflected in the smoother transition between the mass fractions resulting when the RBS strength is modified as compared to changing the mRNA synthesis rate. Notice that, as predicted by equation (2.19), for given a substrate availability, different expression strategies resulting in the same specific synthesis rate will correspond to the same distribution of mass fractions.

Differently from modifying the mRNA synthesis rate for a fixed RBS-strength value, or viceversa, the maximum spMSR of the exogenous protein significantly changed when the substrate availability does not remain constant. The effect of

the differential role of RBS and promoter combinations for scenarios with varying substrate is analyzed in the next section.

2.3.5 *The substrate level emphasizes the differential role of RBS and promoter strengths.*

It is well known that varying combinations of transcription and translation rates affect the stability of metabolic networks [80] and the trade-off between desired expression levels and noise [42] and between expression of endogenous and synthetic genes and growth [125, 38]. In the previous section we showed that for a constant availability of substrate rich in nutrients there are different promoter and RBS combinations that can achieve the same expression level (tantamount the same specific protein mass synthesis rate) of an exogenous protein A. This leads to a multimodal design problem. One can choose between design strategies ranging from using a combination of a weak promoter strength and a strong RBS ($N_A\omega_A \downarrow K_{C^0}^A \uparrow$) to using a strong promoter and a weak RBS ($N_A\omega_A \uparrow K_{C^0}^A \downarrow$). The results depicted in Figure 2.5 show that for the case of constant substrate there is no difference between using one promoter-RBS combination or another as long as the desired spMSR of the protein A remains the same.

However, changes in the substrate have a different impact on the protein expression depending on which one is the promoter-RBS combination selected. Figure 2.1(B and D) illustrates how the mass fractions of ribosomal, non-ribosomal and exogenous proteins change as function of the the growth rate μ, which is indirectly dependent on the availability and quality of the substrate. For a given gene following the weak-RBS strong-promoter strategy (the one followed by the endogenous ribosomal genes) the mass fraction corresponding to the exogenous protein increased as the availability of substrate increased. On the contrary, the strong-RBS weak-promoter strategy (as followed by the endogenous non-ribosomal genes) caused the exogenous protein mass fraction to decrease with increasing availability of substrate.

To understand the differential role of RBS and promoter strengths we first evaluated the dependence of the specific protein mass synthesis rate of an exogenous protein A on the mRNA synthesis rate and the effective RBS strength as a function of the substrate. Figure 2.6A shows the results for two representative substrate levels: low substrate $\nu_t(s_i) = 720$ min^{-1} (left), and high substrate $\nu_t(s_i) = 1260$ min^{-1} (right). The maximum protein synthesis rates (black dashed lines) are located at different places in the design space. Increasing the substrate had the effect of rising the spMSR (the right plot is whiter than the left one). In addition, the optimal synthesis rate moved to the right, ie. for the same mRNA

synthesis rate a higher effective RBS was required to reach the optimum. This implies that a cell configured to obtain the optimum protein synthesis rate for some substrate level will become suboptimal when changing the substrate level.

The resources recruitment strength (RRS) explains this differential effect of RBS and promoter strength on protein expression. For a given protein, its RRS (2.3) is directly proportional to the mRNA synthesis rate. Figure 2.6B and C shows that the mRNA synthesis rate effectively modifies the RRS value regardless of the substrate or the growth rate. As the mRNA synthesis rate increases (displacement to the right in the x-axis) the value of the RRS increases. Therefore, tuning the promoter strength implies tuning the RRS level without affecting the RRS sensitivity to changes in the substrate, the growth rate or the changing availability of free ribosomes.

Different from the promoter strength, the RBS strength determines the sensitivity of the resources recruitment strength to changes in the substrate. It has two different effects on the value of the resources recruitment strength. The first effect is related to the definition of the RBS in equation (2.5). It depends on the association-dissociation rate constants K_b^k and K_u^k and indirectly on the substrate through $K_e(s_i)$. For a given substrate s_i there is a set of infinite combinations of K_b^k and K_u^k that might provide the same RBS strength level. This causes the strength of the RBS to vary with changes in the substrate, so that it decreases as the substrate increases. However, the RBS strength (and therefore the resources recruitment strength value) with a high dissociation constant rate $K_u^k \gg K_e(s_i)$ will be less sensitive to changes in the substrate.

On the other hand, notice from equation (2.5) that the RBS strength defines the sensitivity of the resources recruitment strength to the flux of free resources μr. Decreasing the RBS strength will always reduce the resources recruitment strength value. However increasing the RBS strength will increase the resources recruitment strength value until eventually it saturates. In particular, when $d_m/K_{C^0}^k \ll \mu r$, the resources recruitment strength equation (2.5) becomes:

$$J_k(\mu, r) = E_{mk}(l_{pk}, l_e)\frac{\omega_k(T_f)}{\mu r}\,. \tag{2.21}$$

In this case, the resources recruitment strength value becomes independent of the RBS strength. Thus, there is maximum resources recruitment strength value that can be obtained by increasing the RBS strength. Figure 2.6B shows that for low substrate availability the RBS can increase and yet the resources recruitment

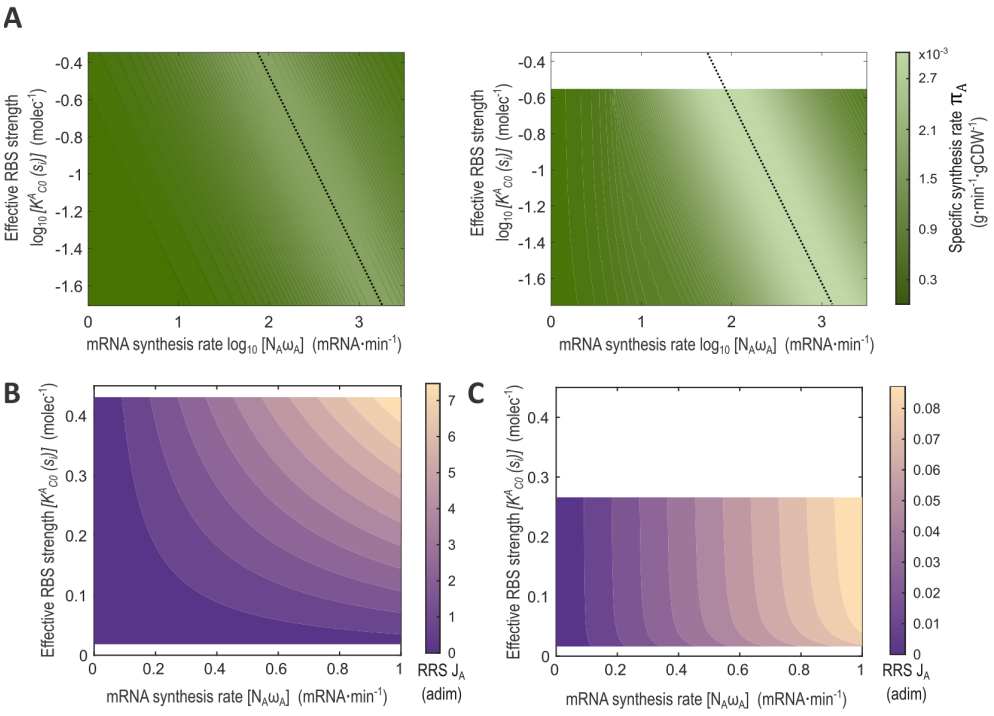

Figure 2.6: Effect of varying the mRNA synthesis rate and the effective RBS strength on **A:** the specific synthesis rate of protein A and **B,C:** the resource recruitment strength (J_A) for different substrate. **B:** Low substrate scenario $\nu_t(s_i) = 720$ min^{-1}. **C:** High substrate scenario $\nu_t(s_i) = 1260$ min^{-1}. The value of J_A was evaluated for the full range of RBS values ($K^A_{C0}(s_i)$) and a representative range of promoter values ($N_A\omega_A$), with E_{mA} and d_{mA} equal to endogenous ribosomal values (without loss of generality).

strength value decrease, and Figure 2.6C shows saturating effect of increasing the RBS strength for a high substrate.

For exogenous protein-coding genes the situation is different depending on whether they do add or not a relevant burden on the cell. In case the exogenous genes do not overburden the cell, the expression patterns will be the same as those for the endogenous genes analyzed above. In case the exogenous genes impose an important burden on the cell, the effects of RBS and the promoter change. In this case μr will be very small and the differential effect of the promoter and RBS strengths is partly lost. In this overburdened scenario, the resources recruitment strength can be approximated as:

$$J_k(\mu, r) = E_{mk}(l_{pk}, l_e)\frac{\omega_k(T_f)K^k_{C^0}(s_i)}{d_{mk}}\,. \tag{2.22}$$

Therefore, the resources recruitment strength only depends on the substrate through the substrate-dependent value of the RBS strength. This causes the RBS-promoter strength strategies to become less differentiated. Yet, it is still possible to apply the analysis above for the distribution of mass fractions as a function of the resources recruitment strength. Thus, the strong-promoter weak-RBS strategy will allow to have resources recruitment strength whose value is less sensitive to changes in the substrate as compared to the weak-promoter strong-RBS one, as observed in Figure 2.1D.

2.4 Discussion

Our model defined the gene *resources recruitment strength* as the key functional coefficient that explains the distribution of resources among the host-circuit and the relationship between the use of these resources and cell growth. The resources recruitment strength generalizes similar proposals in the literature, allowing us to analyze not only scenarios with high cell burden but also scenarios where the competition for cell resources does not overburden the cell extremely. Conversely from the resource demand coefficient defined in [89], where the resource limitation effect is local, we considered that the cell resources (ribosomes) are accessible to all genes in the cell, so exogenous and endogenous host genes compete to recruit cellular resources. The assumption of constant growth rate, constant total number of ribosomes, and highly overburden cell in [89] implies a static resource demand coefficient that is independent of the availability of free resources or the growth rate. This assumption is equivalent to the overburdened scenario in our model with resources recruitment strength given in expression (2.22). However, our RBS strength does depend on the substrate. Therefore, our model can be used in scenarios where the demand on resources changes over time since the resources recruitment strength explicitly captures the mass distribution dependence on cellular growth and substrate availability.

The resources recruitment strength of a gene plays an important role in the value of the specific protein mass synthesis rate. Notice from (2.19) that resources recruitment strength and the spMSR are related. The specific mass synthesis rate is essentially a function of the ratio between the resources recruitment strength of the gene of interest and the total sum of ressources recruitment strengths of the cell. So it provides information about the resources that the gene of interest is

capturing and sharing with other cell components to get expressed. In this sense, the spMSR is a context-dependent magnitudes that requires knowledge of the spMSRs of the remaining genes. The resources recruitment strength is somewhat a more fundamental characterization of a protein-coding gene than the specific protein mass synthesis rate. It is kind of a context-dependent intrinsic magnitude. Its *shape* only depends on the gene characteristics. Its actual value is only defined by the generic flux of free resources μr and the effect of the substrate availability (which may integrate the nutrient quality) on the effective RBS strength. Therefore, the resources recruitment strength measures the intrinsic *avidity* of a given protein-coding gene for cell resources.

Interestingly, the spMSR, i.e the mass synthesis rate of a given protein per cell mass can be related to the definition of *capacity* as proposed in [19] and [79]. There, a cell capacity monitor is implemented by including the constitutive expression of a GFP (Green Fluorescent Protein) gene and determining capacity as the GFP production rate per cell of their capacity monitor. Both concepts, capacity and spMSR, are not the same but are related. In [79] the authors show the existence of a critical capacity. Our results, as seen in Figure 2.5A and B, also showed an upper bound or 'critical' spMSR as a function of the mRNA synthesis rate (mRNA$\cdot$ min^{-1}). The existence of this critical spMSR is not directly related to energy limitation, but it is the result of the peptide optimal allocation for building blocks (amino acids) to synthesize either a given (possibly exogenous) protein or more ribosomes. Indeed, energy limitations will indirectly affect the critical capacity value insofar as they interfere with the flow of building blocks to build up the peptide chains. From the perspective of energy as a resource, our model implicitly incorporates this concept as a fundamental part of the substrate. That is, all the resources needed by the cell eventually come from the substrate. Consequently, our model captures this substrate-energy interaction and it is quantified by the resource recruitment strength and the substrate-dependent effective RBS strength. This approach differs from others such as [125] where energy is modeled explicitly after defining additional gene expression thresholds and a sigmoidal transcription/translation dependence on the energy levels.

The results obtained with our model were relevant both for the analysis of the native host cell, ie. without exogenous protein-coding genes, and the case of having a strain hosting exogenous protein-coding genes.

In the first case, we showed that endogenous ribosomal and non-ribosomal genes clearly differ in their average resources recruitment strength and, therefore, in their average requirement for cell resources. The ribosomal proteins, essential for the cell and continuously being expressed, have higher resources recruitment strength values than the non-ribosomal ones. Moreover, its range of variation

over the ribosomal proteins was much lower than for non-ribosomal ones. This result was not fundamentally determined by the lengths of the coded proteins and is consistent with the fact that to great extent all ribosomal proteins are equally important for the cell. Transcription and translation are energetically expensive processes. It is usually accepted that around 60% of genes are expressed in standard laboratory conditions at any one time in *E. coli*, with only a small fraction making up a large percentage of the total protein. The cumulative sum of the maximum resources recruitment strength gave a good estimation of the percentage of genes expressed at any time. This is consistent with the fact that ribosomal genes are continuously needed for the cell so they are continuously expressed. On the contrary, non-ribosomal genes are regulated to be expressed only when they are required. This also explains the very low resources recruitment strength values obtained for them and reflects these genes are down-regulated most of the time.

It is known that weakly expressed endogenous genes exhibit low RNA polymerase/ribosome ratios, while strongly expressed genes have higher RNA polymerase/ribosome ratios, as this is metabolically efficient [38]. Our model predicted that it is not possible to achieve high expression and high robustness with respect to the resources recruitment strength by only adjusting the RBS strength. There is a trade-off among protein expression, the RBS strength, the robustness and the flux of free resources. The RBS strength sets the sensitivity of the resources recruitment strength with respect to the flux of free resources. Thus, strong RBSs were predicted to be associated to resources recruitment strengths more sensitive to variations in the flux of free resources (i.e. at different growth rates) while weak RBSs provide robustness with respect to the growth rate. This defines how much of a given protein (e.g ribosomal or non-ribosomal) will be expressed at different growth rates.

This trade-off was consistent with the estimated values of the average transcription rates and RBS strengths we obtained for the cell endogenous ribosomal and non-ribosomal genes. We found that the low RBS strength and high transcription rate of ribosomal genes make their resources recruitment strength robust with respect to changes in the flux of free resources with growth rate. On the contrary, for non-ribosomal genes our model predicted an average high RBS strength and a low transcription rate expression strategy. This differential strategies allowed us to explain the relative mass fractions distributions of endogenous ribosomal and non-ribosomal proteins as a function of growth rate. Thus, the differential expression strategies in *E. coli*encode the mass distribution of ribosomal and non-ribosomal proteins for varying growth rates. Our model suggested that the cell achieves a fairly constant absolute expression of non-ribosomal proteins by using

a high RBS strength to express them. On the other hand, the cell uses much weaker RBSs to express the ribosomal proteins. This way the value of the total ribosomal resources recruitment strength remains mostly constant with respect to the non-ribosomal one. As a consequence, the absolute expression of ribosomal proteins increases with growth rate.

The results were applicable to the expression of exogenous protein-coding genes. For a given ratio of ressources recruitment strengths, increasing the expression of exogenous genes decreases the growth rate thereby reducing the absolute mass of endogenous proteins. However, the mass of exogenous proteins accumulates in the cell, which allows the total mass of the cell to increase even if the growth rate decreases. Two extreme cases can be differentiated: either the exogenous genes imposing negligible burdening on the cell or strongly overburdening it. In the first case, the exogenous proteins behave in an equivalent way to the endogenous ones. Therefore, all the results obtained for the last are applicable. This situation is of interest in situations like eg. when designing gene synthetic circuits for feedback regulation of enzymes expression in metabolic pathways. In this case, one of the goals is that the exogenous circuit does not burden the cell in excess, as this will affect the overall performance of the regulated pathway. In the highly overburdened scenario, the resources recruitment strength does not longer depend on the flux of free resources. This causes a diminished differential effect of the RBS and the promoter strengths. Yet, the different sensitivity of the RBS to the available substrate as a function of its strength still has consequences in scenarios with variable substrate. In between, the definition of the our resources recruitment strength allows us to consider a wide range of scenarios with varying cell burden.

2.5 Conclusions

In this Chapter, we have presented a small-size model of gene expression dynamics accounting for host-circuit interactions. The good agreement between the predictions of our model and experimental data highlight the relevance of the cellular resources recruitment strength defined in our model as a key functional coefficient. Our resources recruitment strength coefficient allows us to explain the distribution of resources between the host and the genes of interest. Additionally, it shapes the relationship between the use of resources, cell growth and protein productivity. This functional coefficient explicitly considers the interplay between the flux of available free resources and lab-accessible gene expression characteristics. In particular, the promoter and RBS strengths.

Though we only considered *E. coli*, our findings can be extrapolated to other microorganisms, and the model can be easily fitted using a small amount of experimental data of the host cell.

Among other predictions, the model provides insights into how the differential role of promoter and RBS strengths in protein expression may have evolved in *E. coli* and other micro-organisms to encode the mass distribution between ribosomal and non-ribosomal proteins as a function of cell growth rate. Weak transcription and strong translation and the complementary strong transcription and weak translation emerge as a two potentially equally optimal strategies in the expression space but with different characteristics from the point of view of the sensitivity of the specific synthesis rate of the expressed protein to variations in the cell growth. The capacity of the defined resources recruitment strength functional coefficients to capture the interaction between growth, cell resources and gene expression characteristics is reflected in the fact that the model was able to infer good predictions of the experimental distribution of the cell endogenous ribosomal and non-ribosomal protein mass fractions when fitted to estimate the cell specific growth rate.

The model also explains some of the phenomena typically encountered when building protein expression systems in synthetic biology. Thus, for instance, it explains the limited effect that increasing the RBS strength has to increase the expression of a given protein of interest, saturating at high RBS strengths. Design of synthetic genetic circuits without considering the impact of host–circuit interactions results in an inefficient design process and lengthy trial-and-error iterations to appropriately tune a circuit's expression levels [125]. In this context, our model may also be useful for design purposes in synthetic biology where it can be used to design the proper promoter-RBS strategy depending on the desired behavior of the genes expression as a function of growth rate. In this sense, the resources recruitment strength can be used as a context-dependent intrinsic magnitude for the standard characterization of protein-coding transcription units.

Further extensions of the model can be easily implemented. Thus, the model explicitly considers the relationship between the cell specific growth rate and the population dynamics. As a consequence, it can be integrated within a multi-scale framework that considers the macroscopic extracellular dynamics of the substrate and population of cells in a bioreactor. The model only requires as input a measure of the fraction of available substrate with respect to the saturated case, and predicts both the resulting cell specific cell growth rate and the mass and mass rates of the expressed proteins. This makes its integration with constraint-based models of metabolism rather straightforward. The possibility to consider expression systems using orthogonal ribosomes can also be implemented without

much difficulties. All this makes the model useful in the context of model-based design of gene synthetic circuits and protein expression systems.

Chapter 3

OneModel: a SBML modeling tool

> "Now I'm a pretty lazy person and I am prepared to work quite hard in order to avoid work."
>
> —Martin Fowler, *Refactoring: Improving the Design of Existing Code*

This chapter is an extended version of the conference publication:

- Santos-Navarro, F. N. et al. "OneModel: an open-source SBML modeling tool focused on accessibility, simplicity, and modularity". In: *DYCOPS* (2022), accepted for publication

3.1 Abstract

With the advent of the Systems Biology Markup Language (SBML), a large community of SBML-compliant tools has been created. However, these tools can only be used to their full potential by expert users with advanced programming knowledge. *OneModel* is an open-source text-based tool for defining SBML models in a modular and incremental way that minimizes the user's programming knowledge requirements. It is focused on accessibility, simplicity, and modularity. *OneModel* syntax allows the user to define models based on chemical (and pseudo-chemical) reactions, differential equations, and algebraic equations. *OneModel* is written in *Python*, and it provides two interfaces: a command-line interface for expert-users,

and a graphical user interface for non-expert users. In this chapter, we show the fundamentals of *OneModel* development and a set of guided *OneModel* examples; in particular, we show how to model an antithetical controller and how to integrate it into our host-aware model, which is freely distributed with *OneModel* at `https://github.com/sb2cl/onemodel`.

3.2 Introduction

The Systems Biology Markup Language (SBML) is the software data format for describing models in biology [45]. With the advent of SBML, many SBML-compliant tools have been created. These tools fulfill the syntax and semantics of SBML through different approaches: text-based tools such as *Antimony* [111], *Little b* [70], *BioCRNpyler* [85]; or graphical user interface based tools such as*CellDesigner* [35] and *iBioSim* [78].

Models are often constructed as a monolithic set of equations, reactions, parameters, and species [70]. This leads to inefficient modeling practices in which (i) new models are implemented from scratch, rather than extending previous models; (ii) models have to be validated as a whole, rather than validating the constituent parts of the model; and (iii) models tend to be large and repetitive, rather than defining and reusing modules in their implementation. *Antimony*, *BioCRNpyler* and *Little b* solve this problem by implementing different degrees of modularity. However, these tools are aimed at, or can only be used to their full potential by, expert users with advanced programming knowledge.

OneModel is an open-source text-based tool for defining and compiling SBML models in a modular way. This modularity allows incremental implementations of several simple models to obtain a more complex one. *OneModel* also minimizes the user's programming knowledge requirements. *OneModel* was designed in this Thesis to be easy-to-use and easy-to-incorporate into pre-existing workflows. We used well-documented *Python* libraries to avoid custom code development in its implementation. Therefore advanced programmers will be able to tweak, expand or hack *OneModel* functionality easily. The syntax of our tool implements modularity through object-oriented programming. We were inspired by the Arduino community, where a simple graphical user interface enables non-expert users to contribute their work and ideas to the community.

The rest of the chapter is organized as follows. First, Section 3.3 describes the *OneModel* workflow. Next, Section 3.8 shows a set of guided *OneModel* examples to illustrate its capabilities. Section 3.9 delves into the syntax of *OneModel*. Section 3.4 illustrates the design philosophy principles we followed to develop

OneModel. Section 3.5 shows the software tools we used to build *OneModel*. Section 3.6 presents the *SBML2dae* subpackage that allows us to generate simulation-ready *Matlab* implementations from SBML models. Finally, in Section 3.10 we show the main conclusions of this chapter.

3.3 *OneModel* workflow

This section describes the *OneModel* workflow (Figure 3.1) to help understand its use and usefulness.

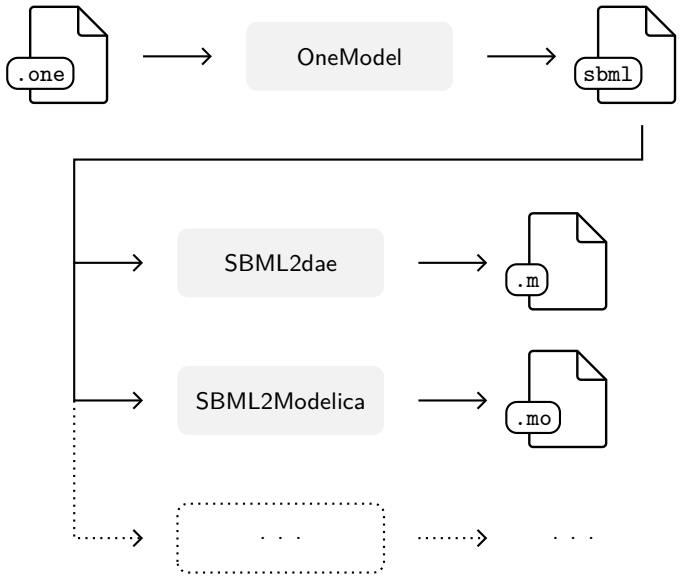

Figure 3.1: *OneModel* workflow. The user writes a model using *OneModel* syntax (".one"). Then, the model is exported into SBML using *OneModel*. Finally, the SBML-compliant tools can be used as (i) *SBML2dae* generates a *Matlab* implementation of the model (".m"), or (ii) *SBML2Modelica* [67] generates a *Modelica* representation (".mo").

The first step is to write a model, as a plain text file with ".one" or ".onemodel" as extension, using *OneModel* syntax. The user can use either the *OneModel*'s editor (available in the graphical user interface) or his own text editor. Our goal is to get non-expert users to use our editor, but we prefer that they eventually switch to working with the command-line interface and their preferred text editor (such as Vim, SublimeText, or Atom)—as they become proficient with *OneModel*—.

The second step is to export this model as an SBML file. Both the graphical user interface and the command line interface can export the model. Actually, the graphical user interface calls the command-line interface in the background to perform the export.

Then, the SBML file can be fed into any available SBML-compliant tools to perform the computational simulations, analysis, etc. The scope of *OneModel* is limited only to the definition of SBML models, so it relies on other tools to make use of them. In this Thesis, we have implemented the tool *SBML2dae* (included with *OneModel*) to generate simulation-ready *Matlab* implementations of SBML models. As an alternative to our *SBML2dae*, we could also have used *SBML2Modelica* [67], a translator tool available in the literature, to generate a *Modelica* implementation instead. In this way, the users can choose which of the large and powerful tools of the SBML community they want to incorporate into their workflow.

Finally, once the model has been validated, we can repeat this loop, generating a new model that imports the code and functionality of the previously defined models.

3.4 *OneModel* design philosophy

OneModel was developed to meet the following design requirements in systems and synthetic biology:

- **Reactions:** to define models based on reactions with linear or rational rates (e.g. a Hill function) that depend on reactant concentrations.
- **ODE:** to define models based on ordinary differential equations (ODE).
- **DAE:** to define models based on ODE and differential-algebraic equations (DAE).
- **Modularity:** to define models incrementally and reuse specific model parts or functions.
- **Accessibility:** low entry barriers for non-expert users, and ease to integrate with other available tools.
- **Simplicity:** the tool's scope is limited to the definition and generation of SBML models; and the simplicity of the tool's internal implementation.
- **Open source:** the code is freely available to the public.

Table 3.1: Software available compared with the requirements. Ticks green (fully met), and yellow (partially met).

Requirements	Antimony	Little b	BioCRNpyler	OneModel
Reactions	✓	✓	✓	✓
ODE	✓	✓	✓	✓
DAE	–	–	–	✓
Modularity	✓	✓	✓	✓
Accesibility	✓	–	✓	✓
Simplicity	✓	✓	✓	✓
Open source	✓	✓	✓	✓

Most of the available text-based tools fail to meet these requirements. Table 3.1 enumerates the tools which best met our design requirements.

The three of them (*Antimony*, *Little b* and *BioCRNpyler*) allow the user to define models based on reactions and ODE, but none of these tools supports algebraic equations (DAE), an inherent element of the reduced-order models generated by the quasi-steady-state (QSS) approximation.

They also provide enough modularity for model definition. *Antimony* had some minor problems that limited its full potential (but they will most likely be fixed in the following versions).

About accessibility, *Antimony* is very accessible through the use of *Tellurium* [73], it defines its domain-specific language (as *OneModel* does), and the need of knowing *Python* is just for simulation and analysis of the generated models (not the definition of them). *BioCRNpyler* is very accessible but does not define a domain-specific language, and it relies on *Python* knowledge for the definition of the models. *Little b* does not meet our requirements for accessibility.

Concerning simplicity, the three tools are focused on the definition of SBML models. *Antimony* defines its custom syntax parser that is a handicap when one looks for the simplicity of the tool's internal implementation: it will make it harder to understand, extend, and maintain the tool's code by external developers. *Little b* source code was not found by the author. *BioCRNpyler* internal implementation is available and is based on Python; therefore, the simplicity requirement is satisfied.

All three tools are freely distributed, but we did not find the source code for *Little b*.

OneModel allows the user to define models with chemical or biochemical pseudo-reactions; and differential and algebraic equations. It has sufficient modularity to implement complex models efficiently. In addition, *OneModel* defines a domain-specific language (to avoid learning *Python* by the user) and incorporates two interfaces: the graphical user interface, which lowers the entry barriers for non-expert users to the minimum, and the command-line interface for expert users to integrate *OneModel* into their workflows. It is focused on definition of SBML models and it minimizes the use of custom code in its implementation. Finally, it is freely distributed and its source code can be found at `https://github.com/sb2cl/onemodel`.

3.5 *OneModel* implementation

OneModel was implemented in *Python* because it is an open-source programming language, is easy to learn, and bridges the gap between compatibility with other programs. Lastly, its extensive libraries facilitated *OneModel* development.

OneModel defines a domain specific language: the *OneModel* syntax. This syntax has been implemented using *TatSu*, which allows us to create syntax parsers conveniently and powerfully. This makes the *OneModel* syntax easily modifiable and adaptable. One advantage of developing a domain-specific language (instead of having implemented just a *Python* library) is that it lowers the entry barriers for the user: there is no need to learn *Python*. Examples of successful domain-specific languages are HTML (HyperText Markup Language) and CSS (Cascading Style Sheets), pseudo programming languages for non-expert users. In addition, the use of a domain-specific language allows the definition of high-level concepts (such as functions, classes, etc.) that are not currently available in SBML.

> *Domain-specific languages are an excellent tool for automating repetitive tasks and improving productivity—they are at the core of OneModel's development—. I highly recommend [33] to learn the why and how of creating domain-specific languages.*

OneModel uses libSBML, a library that simplifies reading and writing SBML files, and it is widely used in the SBML community to export models as SBML code.

Figure 3.2 shows the internal structure of *OneModel*. The core functionality was developed as a *Python* package. *OneModel* provides two different interfaces to simplify and abstract the use of the *Python* package: the command-line interface, and the graphical user interface. The command-line interface can be used directly by an expert user, and it has been developed with Click, a package that allows us to implement professional command-line interfaces. However, using a command-line interface is far from ideal for a non-expert user. Figure 3.3 shows the *OneModel* graphical user interface. It abstracts the use of the command-line interface, and it is a good interface for non-expert users. The graphical user interface was built using *PyQt5*, a *Python* package for developing graphical-user interfaces that can run in any operating system.

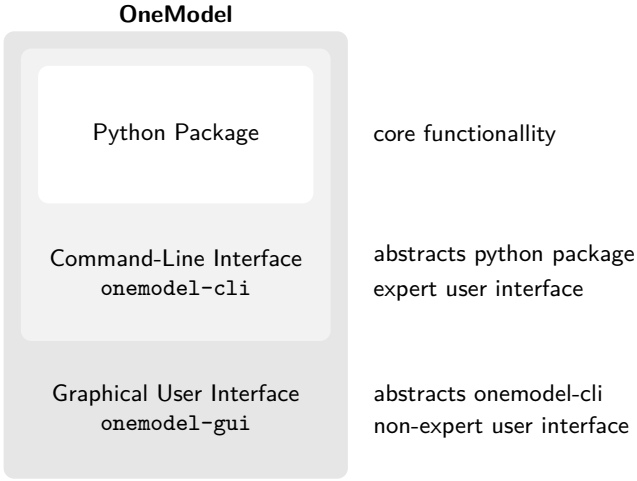

Figure 3.2: Internal structure of *OneModel*, where its core is a *Python* package. The command-line interface abstracts the functionality of the python package, and the graphical user interface represents the functionality of the command-line interface.

3.6 Subpackage *SBML2dae*

At the same time as we developed *OneModel*, we created *SBML2dae*: a *OneModel* subpackage, which provides programming tools to generate SBML exporters to other programming languages for simulation or analysis. *SBML2dae* is open-source, written in *Python*, and complies with *OneModel*'s design philosophy.

By default, *SBML2dae* only allows exporting SBML to *Matlab*. However, it is straightforward for an expert user to create a new parser for another programming

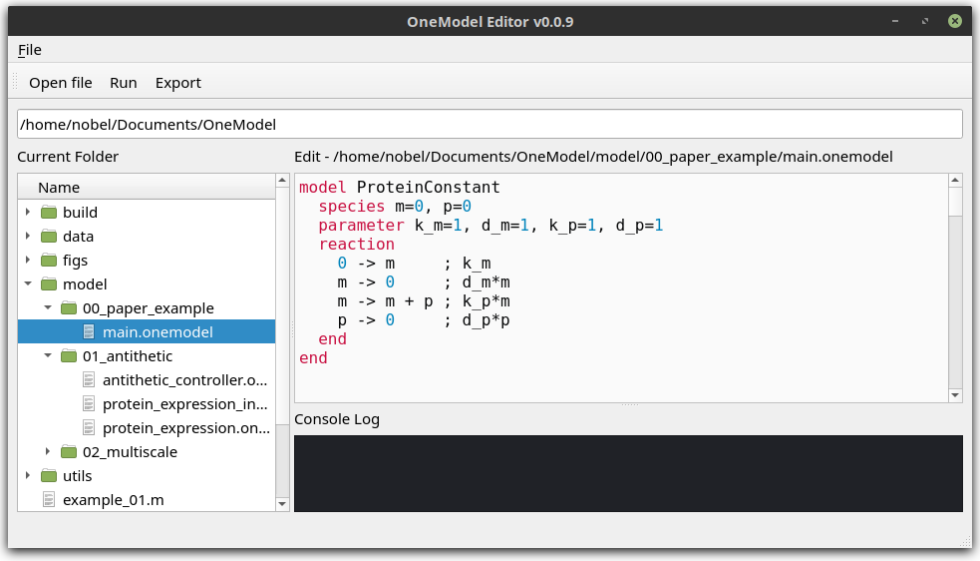

Figure 3.3: *OneModel* graphical user interface running in Linux Mint 19. This graphical interface can be setup in Windows, Mac, and Linux, and it provides a simple text-editor with a syntax highlighter for *OneModel*.

language such as *Modelica*, *Julia*, or *Python*. We expect more syntactic parsers to be incorporated using *SBML2dae* (by our group or by the community).

The differences with other *Matlab* parsers are (i) *SBML2dae* allows the simulation of algebraic loops (an indispensable element for the simulation of reduced-order models, using the quasi-steady-state approximation), (ii) it generates *Matlab* code using classes that significantly facilitates the integration of the models with the rest of *Matlab* tools and (iii) *SBML2dae* is easily modifiable to change the way of exporting the models.

There are excellent tools for simulation and analysis of SBML models, but one of the most significant drawbacks is when the tool does not fit the needs of pre-existing workflows. *SBML2dae* solves this problem by allowing the user to implement customized SBML parsers that fit their particular workflow quickly.

We have used *SBML2dae* to generate most of the *Matlab* simulation code of this Thesis.

3.7 *OneModel* installation

OneModel can be installed from the Python Package Index (PyPI) repository as a package for *Python* 3.8.

For the impatients:

`pip install onemodel`	*install OneModel*
`pip install onemodel -u`	*update to the newest version*
`onemodel-gui`	*open the graphical user interface*
`onemodel-cli`	*access the command-line interface*
`onemodel-cli --help`	*show help message*

OneModel can be installed in Windows, Mac, or Linux; the only requirement is to have installed *Python* 3.8.

Therefore, the first step is to install *Python* 3.8 in your system. The installation process of *Python* varies depending on the operating system you are using: there are great tutorials regarding each operating system on the internet.

Once you have installed *Python*, you have to verify that the correct version of `python` and `pip` are installed. For this, write in the command prompt of your system: `python --version` and `pip --version`, the output of these two commands should indicate that the version of *Python* is 3.8 in both cases. If another version of *Python* appears, it means that you have multiple versions of *Python* installed in your system, then you should use `pip3.8` instead of `pip` for installing *OneModel*.

Now you can install *OneModel* by writing `pip install onemodel` in the command prompt (or `pip3.8 install onemodel`). This command will install *OneModel* and all its dependencies in your system. The examples of this Thesis were done for *OneModel* 0.0.10, they should still work in future versions, but we recommend using this specific version to follow the examples (you can install this version with `pip install onemodel==0.0.10`), and to check the Thesis GitHub repository for more information.

The graphical user interface is opened by writing `onemodel-gui` in the command prompt and the command-line interface is accessed with `onemodel-cli`.

Lastly, if you want to update *OneModel* to the newest version, you can use the following command: `pip install onemodel -u`.

3.8 Examples

To simplify and streamline the modeling tasks, *OneModel* allows us to create incremental implementations of a model by using smaller models previously defined. In the following sections, we present two case scenarios to show how to work with *OneModel* and how these ideas can be carried out with its syntax in an easy way.

These case scenarios demonstrate how to use *OneModel* software. A more complete and commented version of these examples can be found on *Github* (`https://github.com/sb2cl/thesis_fernandonobel`). Further details can be found in the *OneModel* syntax documentation (`https://onemodel.readthedocs.io/en/latest/`).

We used the following software tools to perform the examples: *Matlab* R2020b, *Python* v3.8.0, and *OneModel* v0.0.10. We recommend using the same software versions to replicate the examples.

3.8.1 Constitutive protein expression

In this section, we show an example of how to model constitutive protein expression using *OneModel* syntax. The code examples are easy-to-follow, and they are well commented: there is no need to know *OneModel* syntax to follow them. Moreover, it shows the benefits of using *OneModel* syntax for modeling synthetic biology circuits.

Protein expression—in this chapter, we use protein expression and gene expression interchangeably, as we will implement models of gene expression leading to the synthesis of proteins—is a complex process that involves many reactions and interactions. For this example, we simplify this process, and we take into account mRNA transcription, protein translation, and degradation of mRNA and protein:

$$\begin{aligned} \emptyset &\xrightarrow{k_m} \text{mRNA} \\ \text{mRNA} &\xrightarrow{d_m} \emptyset \\ \text{mRNA} &\xrightarrow{k_p} \text{mRNA} + \text{protein} \\ \text{protein} &\xrightarrow{d_p} \emptyset \end{aligned} \tag{3.1}$$

where $mRNA$ and $protein$ are the mRNA and protein concentration (we could alternatively work with the number of molecules of each species instead of working with concentrations); k_m and k_p are the rate constants related to transcription

and translation, respectively; and d_m and d_p are the degradation rate constants of the mRNA and the protein.

In the set of reactions (3.1), there are three different classes of model-elements: species (mRNA and protein), parameters (k_m, k_p, d_m and d_p) and the reactions. These are the elements that we have to implement using the *OneModel* syntax.

Code 3.1 implements the model-elements in (3.1) using *OneModel* syntax. Lines 3–6 define mRNA and protein as species and set their initial values to zero. Lines 8–13 define the parameters and set their values to one (for example purposes). Moreover, lines 15–20 define the reactions, with their explicit reaction rates placed after the `;` symbol. Note that texts after a `#` symbol are comments which are only used to explain the function of the code.

`ex01_simple_gene_expression.one`

```
1  ### Simple gene expression model. ###
2
3  species       # Start declaring species.
4    mRNA=0      # mRNA concentration.
5    protein=0   # Protein concentration.
6  end           # End declaring species.
7
8  parameter     # Start declaring parameters.
9    k_m=1       # mRNA transcription rate.
10   d_m=1       # mRNA degradation rate.
11   k_p=1       # Protein translation rate.
12   d_p=1       # Protein degradation rate.
13 end           # End delaring parameters.
14
15 reaction                                # Start declaring reactions.
16   0 -> mRNA              ; k_m          # mRNA transcription.
17   mRNA -> 0              ; d_m*mRNA     # mRNA degradation.
18   mRNA -> mRNA + protein ; k_p*mRNA     # Protein translation.
19   protein -> 0           ; d_p*protein  # Protein degradation.
20 end                                     # Stop declaring reactions.
```

Code 3.1: Example of modeling constitutive gene expression (transcription, translation and degradation) using *OneModel*.

Figure 3.4 shows a simulation of Code 3.1. Once we have the *OneModel* implementation, we can use *SBML2dae* to generate a *Matlab* implementation of the

OneModel code to simulate it. This way, if the simulation result is coherent, we can validate the *OneModel* code and continue with this example.

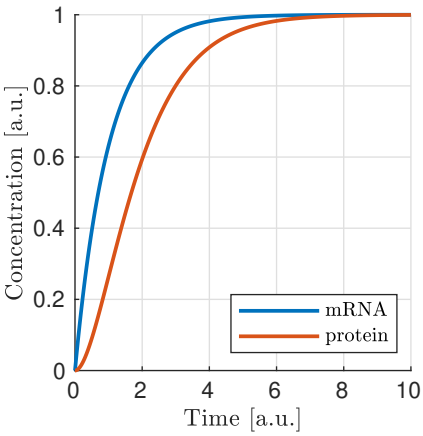

Figure 3.4: Simulation of `ex01_simple_gene_expression.one`. In blue is shown mRNA concentration and in red protein concentration. All units are arbitrary.

A synthetic circuit rarely consists of the expression of just one protein. Typically they are composed of several proteins (two, or three, even hundreds of them). As a result, the mathematical models tend to get very repetitive because it is necessary to replicate the set of reactions for each of the proteins.

One of the most common approaches to this problem is to implement each protein's reactions and equations by hand: copy-pasting the code and thus making a monolithic model. Copy-pasting is a bad programming practice, and it should be avoided. Models of this type are hard to maintain and to use. Indeed, this bad programming practice is due to using software that does not allow incremental and/or modular modeling.

Code 3.2 is an example of this bad programming practice. What we have done in this example is to copy-paste the Code 3.1 twice. We have changed the name of species and parameters to create one set of species and parameters for the constitutive expression of protein A and another set for protein B. This code is hard to read, and this situation will worsen with each extra protein we want to add to the model: we developed *OneModel* to avoid this type of situations.

ex02_two_genes_expression.one

```
### How NOT to model the expression of two genes. ###

species
  mRNA_A=0, protein_A=0   # Gene A mRNA and protein concentration.
  mRNA_B=0, protein_B=0   # Gene A mRNA and protein concentration.
end

parameter
  k_m_A=1,  d_m_A=1   # Transcription and degradation rates of mRNA A.
  k_p_A=1,  d_p_A=1   # Translation and degradation rates of protein A.
  k_m_B=1,  d_m_B=1   # Transcription and degradation rates of mRNA B.
  k_p_B=1,  d_p_B=1   # Translation and degradation rates of protein B.
end

reaction
  0 -> mRNA_A                  ; k_m_A           # Transcription mRNA A.
  mRNA_A -> 0                  ; d_m_A*mRNA_A    # Degradation mRNA A.
  mRNA_A -> mRNA_A + protein_A ; k_p_A*mRNA_A    # Translation protein A.
  protein_A -> 0               ; d_p_A*protein_A # Degradation protein A.
  0 -> mRNA_B                  ; k_m_B           # Transcription mRNA B.
  mRNA_B -> 0                  ; d_m_B*mRNA_B    # Degradation mRNA B.
  mRNA_B -> mRNA_B + protein_B ; k_p_B*mRNA_B    # Translation protein B.
  protein_B -> 0               ; d_p_B*protein_B # Degradation protein B.
end
```

Code 3.2: Example of bad programming practices to avoid. Here is modeled the expression of two genes by copy and pasting the Code 3.1.

The efficient solution to this problem is to use modularity. *OneModel* syntax allows us to wrap Code 3.1 as a `model`. We group all the species, parameters, and reactions as a module which we can reuse by instantiating it as objects instead of copy-pasting the code for each protein. This way, we avoid copy-pasting the code for each protein. Instead, we can create multiple instances of this model.

Code 3.3 shows how to implement Code 3.1 as a `model`. This process is easy to do; we need to wrap the previous code inside the `model` and `end` keywords (lines 4–19). This way, `ProteinConstitutive` is a constructor which will generate instances of the model for us.

In the `standalone` block, we show an example of using `ProteinConstitutive`. We have just created object A which is an instance of model `ProteinConstitutive`. Object A has a copy of all the model-elements of `ProteinConstitutive`, and they

are accessible by the use of the `.` operator. For example, the mRNA concentration of object A can be accessed as `A.mRNA`.

ex03_protein_constitutive.one

```
### Definition of ProteinConstitutive. ###

## ProteinConstitutive models constitutive gene expression. ##
model ProteinConstitutive  # Start declaring model.

  species mRNA=0, protein=0  # mRNA and protein concentration.

  parameter
    k_m=1, d_m=1  # mRNA transcription and degradation rate.
    k_p=1, d_p=1  # Protein translation and degradation rate.
  end

  reaction
    0 -> mRNA               ; k_m          # mRNA transcription.
    mRNA -> 0               ; d_m*mRNA     # mRNA degradation.
    mRNA -> mRNA + protein ; k_p*mRNA     # Protein translation.
    protein -> 0            ; d_p*protein  # Protein degradation.
  end
end  # End declaring model.

## Example of how to use ProteinConstitutive. ##
standalone
  A = ProteinConstitutive()
end
```

Code 3.3: Example of how to build a reusable model for constitutive gene expression using *OneModel* syntax.

Code 3.4 shows how easy it is to model the expression of two proteins taking advantage of the previously defined model. First, we must import the previous code into the new model (line 5). And then, we just need to create as many proteins as we need by writing lines 8–9. Declaring models and instantiating objects is an efficient way to model the expression of two proteins.

ex04_two_genes_expression.one

```
### How to model the expression of two proteins. ###

# Import ProteinConstitutive model.
# (note that the standalone code is not imported).
import './ex03_protein_constitutive.one'

# Initialize A and B as instances of ProteinConstitutive.
A = ProteinConstitutive()
B = ProteinConstitutive()

# We could easily add more proteins by writing:
# C = ProteinConstitutive()
# ...
```

Code 3.4: Example of how to use the model defined in Code 3.3 to model the expression of two genes.

3.8.2 Induced protein expression

`ProteinConstitutive` models genes that are constitutive expressed. In many synthetic circuits, the presence of a transcription factor (which could also be a protein) can induce gene expression. This section, we show how to create another model for induced protein expression.

As a first approach, we could create a new model by copy-pasting the code of `ProteinConstitutive` and modifying it to make the expression inducible: this would be another type of bad programming practice. Doing that is equivalent to what we did in Code 3.2, and it would lead to an inefficient workflow because each time we want to define a new `model` we will have to duplicate the transcription and translation reactions.

> *Whenever you are tempted to copy-pasting any part of your code: stop doing it; it is an indicator that there is a better way to do it. Take the time to see if someone else has stumbled upon your problem—it's a golden opportunity to improve your programming skills—.*

In the previous case, the solution was to implement a `model` instead of copy-pasting the code. Here the solution is to create a new `model` by extending the functionality of `ProteinConstitutive`.

`ex05_protein_induced.one`

```
1  ### Definition of ProteinInduced. ###
2
3  import 'ex03_protein_constitutive.one'
4
5  ## ProteinInduced extends the ProteinConstitutive model to make ##
6  ## the expression inducible by a transcription factor.          ##
7  model ProteinInduced(ProteinConstitutive)
8
9    input TF       # Define the transcription factor as an input.
10   species k_m=0  # Override the parameter k_m to be a species.
11
12   parameter
13     h = 1        # Half-activation threshold.
14     k_m_max = 1  # Maximum transcription rate.
15   end
16
17   # Set the value of k_m as an substitution equation.
18   rule k_m := k_m_max * TF/(TF+h)
19 end
20
21 ## Example of how to use ProteinInduced. ##
22 standalone
23   A = ProteinConstitutive()
24   B = ProteinInduced()
25
26   rule B.TF := A.protein  # Set protein A as the transcription factor of B.
27 end
```

Code 3.5: Example of modeling induced gene expression by extending the previously defined `ProteinConstitutive` model.

Code 3.5 shows the definition of a new model `ProteinInduced` by extending `ProteinConstitutive`. First, we have to import the code of `ProteinConstitutive` (line 3). We declare `ProteinInduced` model and we set `ProteinConstitutive` as its parent (note that the name of `ProteinConstitutive` is in the parentheses in line 7). This way, `ProteinInduced` will have all the model-elements defined in its parent. The rest of the work is to add the inducible part to the model. For this, we do not need to change the reactions; we just need the value of parameter `k_m`

to change depending on the transcription factor concentration. To this end, we declare the transcription factor `TF` as an input (line 9). We override the parameter `k_m` to be a species in line 10 (note that the declaration of species refers both to chemical species or to state variables). The last step is to declare the parameters for a Hill-like function (lines 13–14) and assign the value of `k_m` as the Hill function using the substitution rule.

The `standalone` example (lines 22–26) models a constitutively expressed protein A and a protein B which is induced by A. Figure 3.5 shows a simulation of `ProteinInduced`.

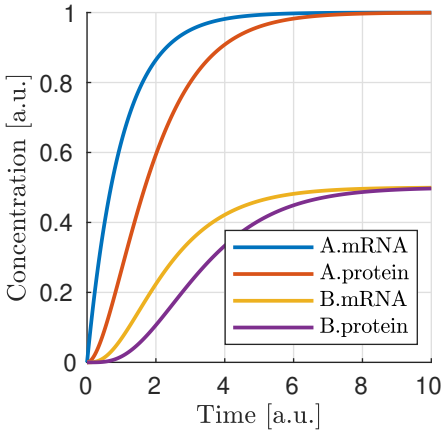

Figure 3.5: Simulation of `ex05_protein_induced.one`. Protein A is expressed constitutively, and protein B expression is induced by protein A. The mRNA and protein concentration of gene A are shown in blue and red, and the ones of gene B are shown in yellow and purple. All units are arbitrary.

3.8.3 Antithetic controller

To exemplify more complex gene circuits, in this section, we model an antithetic controller making use of the models for constitutive and induced protein expression defined in the previous sections.

The antithetic controller is a synthetic gene system to robustly control the expression of a protein of interest [2]. This circuit is implemented using three genes coding three proteins: sigma z_1, anti-sigma z_2, and the protein of interest x. Normally, z_1 is constitutively expressed and induces the production of x. In turn,

x activates the expression of z_2. Finally, z_1 and z_2 annihilate each other in a sequestration reaction, thus closing the loop.

The following set of reactions shows all the biochemical reactions in this gene circuit. The reactions related to protein expression and degradation:

$$\begin{array}{lll}
\emptyset \xrightarrow{k_m^{z1}} \mathrm{m_{z1}} & \emptyset \xrightarrow{k_m^{z2} f(x)} \mathrm{m_{z2}} & \emptyset \xrightarrow{k_m^{x} g(z_1)} \mathrm{m_x} \\
\mathrm{m_{z1}} \xrightarrow{d_m^{z1}} \emptyset & \mathrm{m_{z2}} \xrightarrow{d_m^{z2}} \emptyset & \mathrm{m_x} \xrightarrow{d_m^{x}} \emptyset \\
\mathrm{m_{z1}} \xrightarrow{k_p^{z1}} \mathrm{m_{z1}} + \mathrm{z_1} & \mathrm{m_{z2}} \xrightarrow{k_p^{z2}} \mathrm{m_{z2}} + \mathrm{z_2} & \mathrm{m_x} \xrightarrow{k_p^{x}} \mathrm{m_x} + \mathrm{x} \\
\mathrm{z_1} \xrightarrow{d_p^{z1}} \emptyset & \mathrm{z_2} \xrightarrow{d_p^{z2}} \emptyset & \mathrm{x} \xrightarrow{d_p^{x}} \emptyset
\end{array} \tag{3.2}$$

and the sequestration reaction:

$$\mathrm{z_1} + \mathrm{z_2} \xrightarrow{\gamma} \emptyset\,, \tag{3.3}$$

where m_i are the mRNA concentration of z_1, z_2 and x; k_m^i and k_p^i are the rate constant parameters related to transcription and translation, respectively; d_m^i and d_p^i are the degradation rate constants of mRNA and protein; $f(x)$ and $g(z_1)$ are activation Hill-like functions; and γ is the antithetical sequestration rate constant.

Note that in (3.2) there are a lot of repetitive reactions to model the expression of z_1, z_2, and x. We can take advantage of this repetitive structure and use the model for protein expression that we defined before.

Code 3.6 is an implementation of the `AntitheticController` model using the models `ProteinConstitutive` and `ProteinInduced`.

First, we have to import the previous models into this new one (line 3). Note that `'ex05_protein_induced.one'` already imports `ProteinConstitutive`. We define the three proteins which make the circuit. We use `ProteinConstitutive` for protein `z1` and `ProteinInduced` for proteins `z2` and `x` (lines 6–8). We define the annihilation rate constant `gamma` (line 9), and we add the annihilation reaction to the model (line 14).

Note that if we define models using reactions (instead of rules), we can add more reactions to previously defined models, and *OneModel* will update all the rates of changes of the species automatically—this makes it very easy to expand the functionality of models as we have done with adding the antithetic reaction in Code 3.6.

ex06_antithetic_controller.one

```
### Definition of AntitheticController. ###

import 'ex05_protein_induced.one'  # ProteinConstitutive and ProteinInduced.

model AntitheticController
  z1 = ProteinConstitutive()  # Sigma factor.
  z2 = ProteinInduced()       # Anti-sigma factor.
  x  = ProteinInduced()       # Protein of interest to control.
  parameter gamma = 1         # Antithetic sequestration rate.

  reaction
    # We have to add the antithetic reaction.
    # Note that we can access species inside objects using '.' operator.
    z1.protein + z2.protein -> 0 ; gamma*z1.protein*z2.protein
  end

  rule
    x.TF  := z1.protein  # Set z1 as the transcription factor of x.
    z2.TF := x.protein   # Set x as the transcription factor of z2.
  end
end

standalone  # Example of how to use the AntitheticController.
  circuit = AntitheticController()
end
```

Code 3.6: Example of modeling an antithetic controller using *OneModel* syntax.

Then, we set `z1` as the transcription factor of protein `x`. In turn, `x` will be the transcription factor of protein `z2` (lines 18–19). Finally, we set the standalone example as just the AntitheticController. Figure 3.6 shows a simulation of the `AntitheticController` model.

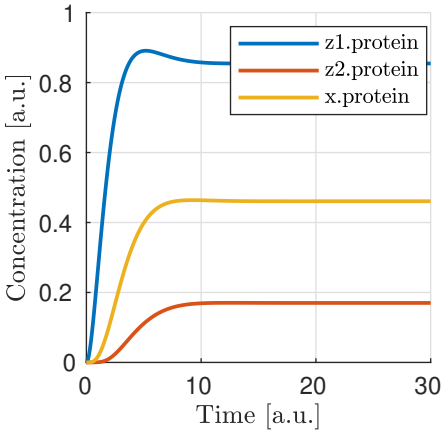

Figure 3.6: Simulation of `ex06_antithetic_controller.one`. The concentration of sigma protein z_1 is shown in blue, in red the anti-sigma concentration z2 and in yellow the protein of interest x. All units are arbitrary.

3.8.4 Host-aware antithetic controller

This last example shows the use of *OneModel* with complex and large models. The first approach to model a synthetic gene circuit is usually done by neglecting the interactions between the host cell and the gene circuit. However, there is an increasing need to include host dynamics to improve model prediction capabilities. These host-aware dynamic models are complex and not easy to implement since they may contain several states and equations [100].

Code 3.7 depicts the `WildType` model that represents the host-aware model freely distributed with *OneModel*. Lines 8–17 show an incomplete representation of it just for demonstration purposes. This model implements the equations from the host-aware model of Chapter 2 and takes into account the host dynamics, the competition for cell resources in protein expression, and its effect on cell growth. The model `WildType` is rather complex. However, from a user perspective, we only need to know how to modify its input `WSum` (line 10), which is a value that keeps track of the burden introduced by the expression of exogenous proteins like the ones introduced by the antithetic controller. `WildType_ProteinConstitutive` is a model provided also by `WildType`, which defines the base protein expression mechanism. The user may use this model as a building block for its own circuits (similarly to section 3.8.1). There were no inputs in the original `ProteinConstitutive`. However `WildType_ProteinConstitutive` is a more complex model which has in-

puts to calculate protein expression as function of the effective translation rate of ribosomes `nu_t`, and the cell growth rate `mu`.

`ex07_wild_type.one`

```
### Definition of WildType. ###

# We show here an incomplete implementation of WildType and
# WildType_ProteinConstitutive and WildType_ProteinInduced
# (the 'dot-dot-dot' indicates that the original original code continues).

## Host aware model of a E.coli. cell##
model WildType
  input WSum_exo  # Burden generated by exogenous genes.
  # The model takes into account many relevant variables as:
  # the effective ribosome elongation rate, the growth rate, etc.
  species nu_t, mu, ...
  ...
end

## Model for consitutive protein expression. ##
model WildType_ProteinConstitutive
  input nu_t, mu, ...  # The inputs of this model are species of WildType.
  species W # Burden generated by the expression of this protein.
  ...
end

## Model for induced protein expression. ##
model WildType_ProteinInduced (WildType_ProteinConstitutive)
  input TF  # Transcription factor which induced expression.
  ...
end
```

Code 3.7: `WildType` is the host-aware model freely distributed with *OneModel*, `WildType_ProteinConstitutive` is the constitutive protein expression mechanism, and `WildType_ProteinInduced` is the inucible protein experssion mechanism. The figure shows an incomplete representation of these models only for the example purposes.

Code 3.8 shows the `WildType_AntitheticController` model that is the implementation of the antithetic controller taking into account the host dynamics.

First, in line 2 we import the `WildType`, the `WildType_ProteinConstitutive` and the `WildType_ProteinInduced` models. Then in line 4, we declare a new model

`WildType_AntitheticController` as an extension of `WildType` model; this way, we have all the dynamics of the host, and we only need to add the remaining dynamics of the antithetic controller. In lines 5–8, we define the proteins of the antithetic controller, sigma `z1` as a constitutive expressed protein and anti-sigma and the protein of interest (`z2` and `x`) as induced expressed proteins. The `WildType` model needs to know if any of the proteins are under active degradation to ensure that the cell mass is calculated correctly. Therefore we cannot implement the antithetic reaction as we did in Code 3.6. However, there is a simple workaround; we can define the complex sigma and anti-sigma `z12` (line 7) that is not expressed directly by the cell (line 10) but is generated as a result of the sequestration reaction of the antithetic controller (line 17). The rest we have to do in the model is to set up the antithetic controller (lines 21–22) and calculate the total burden generated by the exogenous proteins (line 26). Finally, we have to satisfy the inputs for each protein (lines 30–32); this step is omitted in the example for brevity.

ex08_antithetic_controller.one

```
### Host-aware antithetic controller circuit. ###
import 'ex07_wild_type.one'

model WildType_AntitheticController (WildType)
  z1  = WildType_ProteinConstitutive() # Sigma.
  z2  = WildType_ProteinInduced()      # Anti-sigma.
  z12 = WildType_ProteinConstitutive() # Sigma and anti-sigma complex.
  x   = WildType_ProteinInduced()      # Protein of interest.

  parameter z12.omega=0  # z12 is not expressed.
  parameter gamma=10     # Antithetic sequestration rate.

  # We have to add the antithetic reaction, but note that in the
  # wild-type model we cannot degrade proteins directly, instead
  # we have to redirect the mass of z1 and z2 into z12.
  reaction
    z1.protein + z2.protein -> z12.protein ; gamma*z1.protein*z2.protein
  end

  rule
    x.TF  := z1.protein  # Set z1 as the transcription factor of x.
    z2.TF := x.protein   # Set x as the transcription factor of z2.
  end

  # We have to take into account the burden of the exogenous proteins.
  rule cell.WSum_exo := z1.W + z2.W + z12.w + x.W

  # Lastly, we have to satisfy the inputs of z1, z2, z12, and x.
  rule
    z1.nu_t := cell.nu_t
    z1.mu := cell.mu
    ...
  end
end

standalone  # Example of how to use the WildType_AntitheticController.
  ac = WildType_AntitheticController()
end
```

Code 3.8: Example of how to model a host-aware antithetic controller with *OneModel* syntax.

Figure 3.7 shows a set of simulations of the host-aware antithetic controller performed with *SBML2dae*. In these simulations, we have considered two simplifications of the model: (i) to neglect the burden produced by the antithetic controller to the host—this is done by removing the `z1.W + z2.W + z12.W` term from line 26 of Code 3.8—, and (ii) to neglect the dilution of the sigma and anti-sigma factors of the antithetic controller.

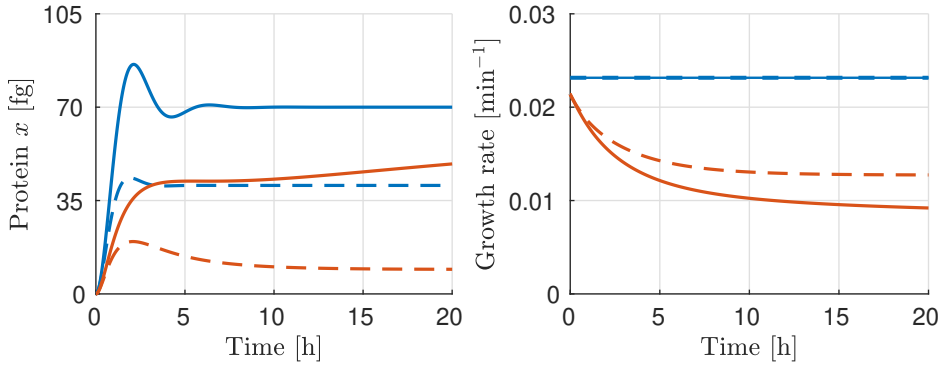

Figure 3.7: Set of simulations of the host-aware antithetic controller, each line corresponds to a simulation with different conditions. The solid lines correspond to simulations that neglect the dilution of the sigma and anti-sigma factors on the antithetic controller, while the dashed lines correspond to simulations that take dilution into account. The blue lines correspond to simulations that neglect the burden produced by the antithetic controller to the host, and the red lines correspond to simulations in which this burden is taken into account. The left plot shows the protein of interest x to be controlled by the antithetic controller, the reference of the antithetic controller was set to 70 fg (see Section 5.2.1 for more information about the antithetic controller's reference) . The right plot shows the host growth rate.

If we neglect the burden produced by the antithetic controller (blue lines), we can see how the growth rate does not change during simulation time (both solid and dashed blue lines overlap in the plot). However, if we consider that burden (red lines), we see how the growth rates vary accordingly—because now the host is using resources to express the antithetic controller instead of using them to grow.

Suppose we neglect the dilution of the sigma and anti-sigma factors (solid lines). In that case, the antithetic controller will preserve the integral action, which makes x to reach the reference of 70 fg in the solid blue line, however in the solid red line (where the burden is taken into account) the x still gets to the reference (this is not shown in the figure) but it takes much more time—due to effect of the integral action is diminished due to the cell is growing slower—. However, if

we consider the dilution (dashed lines), the antithetic controller loses the integral action and never achieves the reference value.

We have done this because it is an excellent example of showing the flexibility of the *OneModel* workflow; it is straightforward to perform multiple simulations with different conditions taking advantage of the modularity.

3.9 *OneModel* syntax

The *OneModel* syntax simplifies the definition of SBML models and extends the functionality of SBML by introducing high-level elements. The models can be defined using base SBML elements such as parameters, species, reactions, rules (substitution, differential or algebraic equations); or using *OneModel* high-level elements such as functions, classes, inheritance. High-level elements, which do not have an SBML representation, are converted into equivalent low-level representations when the model is exported to SBML.

Figure 3.8 shows the typical structure of a model developed with *OneModel*. There are three main sections: import dependencies from previously defined models (line 1), the model definition (lines 2–8), and the standalone example (lines 9–11).

```
1  import dependencies
2  model ModelName(ParentModelName)
3    define inputs
4    define species
5    define parameters
6    define reactions
7    define rules
8  end
9  standalone
10   example script on how to use the model
11 end
```

Figure 3.8: Pseudo-code representing the structure of a model written with *OneModel* syntax. In line 1, the user can import previously defined models. In lines 2–8, the user can define one o more models using: inputs, species, parameters, reactions, and rules. The models might be an extension code of the ones previously defined. Finally, in lines 9–11, the user can define an example-of-use instance for the models in this file and within the standalone block.

The first section imports all the dependencies. Here, the user imports previously defined models to use and combine them as building blocks for developing more complex models or to extend their functionality.

The second section is the model definition. In this section, several models can be defined using: (i) inputs, (ii) species, which can be interpreted both as chemical species and state variables, (iii) parameters used by reactions and rules, (iv) biochemical reversible and irreversible reactions, and (v) rules which can be substitution, differential or algebraic equations. Models can extend the functionality of previous ones; for example the model with name `ModelName` will inherit all the elements (inputs, species, parameters, etc) of the parent model with name `ParentModelName` (line 2).

The last section is the standalone instance. Within this block, the user can define an example code that shows how to use the model (or models) defined above. It is important to note that the code inside the standalone block is not imported: the standalone code is only read when we export this model file directly. The advantages are: (i) each model file can always be exported as a standalone model allowing us to test each model individually for coherence, and (ii) the user always has an example of how to use the model.

We will briefly show the model-elements available in *OneModel* syntax in the following subsections.

3.9.1 Comments

Single-line comments start with a numeric hash symbol `#` and finish at the end of the line. Code 3.9 shows an example *OneModel* code with comments.

```
1  # This a single-line comment.
2
3  # SimpleReaction models a simple reaction.
4  model SimpleReaction   # Start definition of SimpleReaction.
5    species A = 0, B = 0 # Define species A and B.
6    parameter k = 1      # Define k as a parameter.
7    reaction             # Start reaction block.
8      A -> B ; k*A       # Conversion reaction.
9    end                  # End reaction block.
10 end                    # End the definition of SimpleReaction.
```

Code 3.9: Example of using comments with *OneModel* syntax.

Note: comments are not saved into the SBML files when the model is exported.

3.9.2 Parameters

Parameters are time-invariant values that do not change during the simulation of the model. They are used in the definition of the reaction rate constant and rules. Parameters are declared using the keyword `parameter`. Code 3.10 shows the different alternatives for declaring parameters.

```
# Parameters can be defined with a single-line command.
# Here we declare a new parameter 'k1' with value 1.
parameter k1 = 1

# We can also declare multiple parameters using commas.
# (if we don't set the value, the parameter is set to 0)
parameter k2 = -1, k3 = 3.14, k4 = 1e+5, k5

# Other alternative is to use a 'parameter' 'end' block, without using commas.
# This way be can comment easily the each parameter.
parameter
  # Transcription rate of mRNA. molec/min
  k = 1
  # Degradation rate of mRNA. 1/min
  d = 0.12
end
```

Code 3.10: Example of declaring parameters with *OneModel* syntax.

Note: parameters are saved in SBML as “parameter” elements.

3.9.3 Species

Species represent both (pseudo-)chemical and (pseudo-)biological species and state variables; values that change during simulation time due to reactions or rules. Species are declared using the keyword `species`.

```
# Similar to parameters, species can be defined with a single-line command.
# Here, we declare a species S1 with initial value set to 1.
species S1 = 1

# We can also declare multiple species using commas.
# If we don't set the initial value, it will be set to 0 by default.
species S2 = -1, S3 = 1e-1, S4

# species also allows multiple declaration with a 'species' 'end' block.
species
  mRNA = 0     # mRNA concentration. nM/cell
  protein = 0  # protein concentration. nM/cell
end
```

Code 3.11: Example of declaring species with *OneModel* syntax.

Note: species are saved in SBML as "species" elements.

3.9.4 Reactions

Reactions represent a process in which the reactants are consumed to generate the products at a given reaction rate. They are one of the ways to define how the value of the `species` will be modified during simulation time—being the other way to use rules.

If a species is present in a reaction (as a reactant or a product), the species value cannot be set by a rule and vice versa. The value of a species can only be set by (i) a set of reactions or (ii) one unique rule. If we define the rate of change of a species with a reaction, we cannot add a new rule which sets its value, but we can still add more reactions that interact with that species (as a reactant or a product).

Reactions are declared using the keyword `reaction` and are defined using the following syntax:

```
name: reactants -> products; rate
```

where `name:` is the name of the reaction (this is an optional part), `reactants` are the name of the species which the reactions will consume, `->` is the arrow of the reaction and indicates the directionality of it, `products` are the name of the species produced by the reaction, `;` is used to separate the reaction rate from the rest of the reaction, and `rate` is an equation (composed by parameters or species) which calculates the reaction rate. If multiple species are consumed or produced at the same time, their names must be separated by a `+` sign.

```
# Reactions are declared within a 'reaction' 'end' block.
reaction
  # Species S is consumed to form P at rate k*S.
  S -> P ; k*S
  # mRNA transcription at a constant rate k_mRNA.
  0 -> mRNA ; k_mRNA
  # mRNA degradation proportional to mRNA concentration.
  mRNA -> 0 ; d_mRNA*mRNA
  # Antithetic sequestration.
  sigma + anti_sigma -> 0 ; gamma*sigma*anti_sigma
  # We can name a reaction writing its name followed by a ':'.
  # In this way we can refer to this reaction later in the code.
  R1: 0 -> A; k_A
end
```

Code 3.12: Example of declaring reactions with *OneModel* syntax.

Note: reactions are saved in SBML as “reaction” elements.

3.9.5 Substitution, rate, and algebraic rules

Rules represent three types of equations (substitution, rate, and algebraic equations), and they are used to define how the value of a species varies in simulation time. They are declared using the keyword `rule`.

– *Substitution rules*

Substitution rules are defined as `name := equation`, and they are used to assign the value of a species with a mathematical expression. The substitution rules

are evaluated in each simulation step; therefore, the value of a species set by a substitution rule can change over time. Note that substitution rules do not introduce new constraints in a model; they are just an assignment of the value of a variable.

For example, `x_total := x_active + x_inactive` is a valid substitution rule.

– *Algebraic rules*

Algebraic rules are declared as `name == equation`, and they are used to introduce algebraic constraints (the value of one species must hold a mathematical constraint during simulation time). The algebraic rules are evaluated in each one of the simulation steps. In SBML algebraic rules take the form of `0 == equation`, however in *OneModel* is necessary to explicitly write `name == equation` where `name` is the name of the species which the solver has to iterate to satisfy the algebraic rule.

For example, `0 == x_active + x_inactive - x_total` is not a valid algebraic rule—we have to explicitly tell to *OneModel* which one of the species will have its value set by the algebraic rule—. Then, `x_total == x_active + x_inactive` is the valid form of a algebraic rule.

– *Substitution vs. algebraic rules*

Substitution rules are mathematically equivalent to algebraic rules. However, the main difference is that the substitution rules are exact, and the value of the algebraic rules is solved numerically, so the accuracy of the result will depend on the error tolerances of the simulator.

Tips:

- If you can find an analytical solution for your model, use substitution rules instead of algebraic rules.
- Algebraic rules are more challenging to simulate, and many simulators do not support them. Use algebraic rules as a last resort.
- When using the quasi-steady state approximation, it may be difficult—or even impossible—to obtain an analytical solution of the derivatives of your model. In this case, it is unavoidable to use algebraic rules to set the derivatives to zero.

– *Rate rules*

Rate rules are declared as `name '= equation`, and they are used to define the rate of change of a species over time (to set its derivative).

The Code 3.13 shows an example of the use of each type of rule in *OneModel*.

```
# Rules are declared within a 'rule' 'end' block.
rule
  # Substitution rule.
  S := 10*s
  # Algebraic rule.
  # (note that this could be changed into a substitution rule).
  y == 10 - x
  # Rate rule.
  x '= S - x
  # As reactions, we can give a name to the equation with ':'.
  E1: z := x + y
end
```

Code 3.13: Example of declaring rules with *OneModel* syntax.

Note: the rules are saved in SBML as "assignment rules", "rate rules", and "algebraic rules" respectively. The value of a species can be set by a set of reactions or by a rule. It is not possible to combine both methods.

3.9.6 Model and objects

OneModel syntax implements modularity through class-based programming.

A class groups a set of model-elements (parameter, species, rules, etc.) under a class name. Classes cannot be used directly to simulate; they are used to instantiate objects that will have a copy of each of the model-elements of the class. A class is defined using the keywords `model` and `end`. An object is defined using the name of the class we want to instantiate, followed by `()`.

Classes can inherit the model-elements of previous classes. Inheritance is done by writing the parent class name in parentheses in the definition of a new class.

Code 3.14 shows an example of using classes and objects.

```
# Define 'Protein' model.
model Protein  # Start model definition.
  species protein
  parameter k = 1, d = 1
  reaction
    0 -> protein ; k
    protein -> 0 ; d*protein
  end
end  # End model definition.

# Define 'ProteinInduced' as an extension of 'Protein'.
model ProteinInduced (Protein)
  input TF
  parameter h = 1
  rule k := TF/(h + TF)  # Override the value of 'k' with a rule.
end

standalone
  A = Protein()           # Instantiate an object of 'Protein'.
  B = ProteinInduced()    # Instantiate an object of 'ProteinInduced'.
  rule B.TF := A.protein  # Object properties can be accessed with '.'.
end
```

Code 3.14: Example of declaring models, extending them and instantiating objects with *OneModel* syntax.

Note: classes are not saved in SBML, and objects are saved in SBML by saving their model-elements with a prefix related to the object's name. For example: the species `A.protein` in Code 3.14 will be saved as a species with name `A__protein`.

3.9.7 Input

Inputs represent species or states that are not calculated within a model but are necessary to calculate the rest of the equations and reactions of the model. They are defined with the keyword `input`, and the value of an input is set using a substitution rule.

```
model ProteinInduced  # Note that we don't set the value of TF in the model.
  input TF                      # Declare TF as an input.
  species protein
  parameter k = 1, d = 1, h = 1
  reaction
    0 -> protein ; k*TF/(TF+h)  # Use the value of TF.
    protein -> 0 ; d*protein
  end
end

standalone  # It is here where we set the value of TF.
  species A = 10
  B = ProteinInduced()
  rule B.TF := A                # Set the value of B.TF to A.
end
```

Code 3.15: Example of defining inputs in a model with *OneModel* syntax.

Note: inputs are saved as species in SBML.

3.9.8 Import

OneModel syntax allows us to import code from other files into the current one using the keyword `import`. The code from the imported file is executed as it were present in the current file, but the code inside the `standalone` block is omitted.

```
# To import a model, we have to write the path of the file we want to import
# relative to the current file path.
import './03_protein_constitutive.one'

# The code inside '03_protein_constitutive.one' will be now accesible,
# and we can use the models defined in it.
A = ProteinConstitutive()
```

Code 3.16: Example of importing code with *OneModel* syntax.

Note: SBML models do not support importation. When we export into SBML, all the model-elements from the imported files will be saved in the generated SBML.

3.9.9 Standalone

The code inside the standalone block will not be imported to other files: the standalone code is only executed when we run the model directly. The standalone is declared using the keyword `standalone` and `end`.

```
# Here we define a model which we will import later into other file.
model MyModel
  input u
  species x=0
  rule x '= u - x
end

# In the standalone we can define an example to show how to use it.
# Another benefit is that all our models will run as a standalone model,
# making it easier to debug models and maintain them.
standalone
  myModel = MyModel()
  rule myModel.u := 10
end
```

Code 3.17: Example of use of the standalone keyword with *OneModel* syntax.

Note: only if the model is directly exported into SBML (instead of importing it into another model), the contents of its standalone section are saved into the SBML.

3.10 Discussion and conclusions

We developed *OneModel*, a new SBML-compliant tool for defining models focused on user accessibility, simplicity, and modularity. Instead of developing monolithic files that contain all the equations and model parameters values, *OneModel* syntax allows the user to add new models to connect with the old ones—splitting models into modules and re-programming them by making small changes that fulfill the new requirements but always have the option to go back. *OneModel* reduces the modeling efforts by increasing modularity. The user can develop and test each module of a model separately, avoiding starting from scratch every time the user implements a new model.

We used *OneModel* for almost all of the simulations of this Thesis. The guided examples in this chapter showed the benefits of modular incremental implemen-

tations and how it would be easy for a non-expert user to take advantage of previously defined models (such as our host-aware model).

One of the original motivations for building *OneModel* was that several biological processes could only be modeled with algebraic loops or differential-algebraic equations (DAE). Moreover, our host-aware model also includes this type of equations to consider the competition for resources during protein production. Currently, *Antimony* does not support DAE. As a first iteration, we tried to add the algebraic loops to the *Antimony* source code. However, it uses a custom parser (equivalent to *TatSu*) for its syntax, which restricts modifying it. On the contrary, *OneModel* uses a well-documented parser (*TatSu*) which lowers the entry barriers for modifying *OneModel*. This way, the community will easily extend its original functionality.

Modularity is a crucial feature of *OneModel*. Both software tools, *BioCRNpyler* and *Little b*, have great modularity capabilities. Nevertheless, they require that the user has at least some knowledge of *Python* or *Lisp*.

OneModel requires *Python* version 3.8 and is installed as a package with PyPI. Although the installation process is simple, it can be challenging for non-expert users. An executable version and a web interface will avoid this step in the future. *OneModel* syntax is Turing incomplete on purpose because that could divert its purpose from being just a tool to define SBML models (this case is similar to HTML or CSS), and this would compromise simplicity and accessibility.

Error feedback is often one of the weaknesses of domain-specific languages and is key to ensuring accessibility. Currently, *OneModel* provides simple error feedback, but our goal is to improve it. Object-oriented programming in *OneModel* is class-based, but we are experimenting with prototype-based programming which could simplify the internal implementation of *OneModel*.

Antimony, *Little b*, *BioCRNpyler*, *OneModel*, and many other tools not listed here, paved the way to define more complex and larger models efficiently. However, these models are difficult to debug, test, and maintain. Researchers have performed these tasks manually, but this is inefficient for models of this size or sometimes even impossible. Therefore, there is an increasing need for tools to automatically debug and test SBML models.

Model development is a crucial task for researchers and programmers. *OneModel* allows you to simplify and streamline this task.

Chapter 4

Multi-scale model

> "For the great doesn't happen through impulse alone, and is a succession of little things that are brought together. What is drawing? How does one get there? It's working one's way through an invisible iron wall that seems to stand between what one feels and what one can do. [...] And the great isn't something accidental; it must be willed."
>
> —Vincent van Gogh, *The Letters*

The contents of this chapter appeared in the following journal publication:

- Santos-Navarro, F. N. et al. "Gene Expression Space Shapes the Bioprocess Trade-Offs among Titer, Yield and Productivity". In: *Applied Sciences* 11.13 (2021), p. 5859. ISSN: 2076-3417. DOI: 10.3390/app11135859

This paper used an older and slightly less accurate host-aware model than the one described in Chapter 2. In this Chapter, all simulations were redone using the most recent version of the host-aware model. The expressions of the multi-scale model have been updated accordingly in the Chapter. The qualitative results obtained are unchanged from those in the publication above.

4.1 Abstract

Optimal gene expression is central for the development of both bacterial expression systems for heterologous protein production, and microbial cell factories for industrial metabolite production. Our goal is to fulfill industry-level overproduction demands optimally, as measured by the following key performance metrics: titer, productivity rate, and yield (TRY). Here we use a multi-scale model incorporating the dynamics of (i) the cell population in the bioreactor, (ii) the substrate uptake and (iii) the interaction between the cell host and expression of the protein of interest. Our model predicts cell growth rate and cell mass distribution between enzymes of interest and host enzymes as a function of substrate uptake and the following main lab-accessible gene expression-related characteristics: promoter strength, gene copy number and ribosome binding site strength. We evaluated the differential roles of gene transcription and translation in shaping TRY trade-offs for a wide range of expression levels and the sensitivity of the TRY space to variations in substrate availability. Our results show that, at low expression levels, gene transcription mainly defined TRY, and gene translation had a limited effect; whereas, at high expression levels, TRY depended on the product of both, in agreement with experiments in the literature.

4.2 Introduction

Optimal gene expression is central to the development of both bacterial expression systems for production of heterologous proteins and microbial cell factories to produce metabolites of industrial interest. Both applications seek to obtain high levels of products of interest by means of metabolic engineering. The goal is to reach industry-level overproduction demands in an optimal way, as measured using the key performance indices titer, productivity (rate) and yield (henceforth TRY).

In practice, some trade-offs in the TRY space must be reached and be adaptive to the growth and environmental dynamic changes that occur in a bioreactor set-up [107, 124]. Indeed, biomass growth and product yields cannot be simultaneously maximized. For a given substrate uptake rate, a higher growth yield will lead to a higher growth rate at the expense of the product yield.

In silico constraint-based metabolic genome-scale models have proved very valuable in engineering biosynthetic metabolic pathways in which the combined catalytic activity of a collection of enzymes is coordinated so as to produce the desired metabolites. These models provide predictions on maximum theoretical yields,

optimal flux distribution to maximize flux towards some metabolite reaction bottlenecks and the required gene expression leading to increases in fluxes towards the final products [62, 87, 121]. It is possible to deal with the trade-off between yield and productivity using dynamic flux balance analysis (dFBA), which incorporates both process dynamics and the constraint-based metabolic network model and relies on dynamic optimization methods [68, 63, 46].

In practice, the metabolic costs of producing proteins, enzymes, and other cell macromolecules during gene over-expression often lead to shifts away from the optimal predictions. Yet, the focus of these constraint-based models on metabolism excludes proteins and other macromolecules, the major biomass constituents, from the model. Models that integrate metabolism, biomass composition and the cost for the cell to produce a certain protein or enzyme can potentially yield better predictions than those focused on metabolism in isolation [126]. Thus, for instance, dynamic enzyme-cost FBA [123, 47] includes dynamic changes in substrate concentration, a detailed description of biomass composition, and takes into account how much it costs to the cell to produce the desired enzyme.

As an alternative to metabolism-centered models, in the recent years there has been a increasing interest in the development of models of gene expression accounting for cellular resources competition [93, 13]. These range from very coarse-grain ones [105, 6, 38, 14] to semi-mechanistic ones with varied degrees of granularity [125, 50, 66]. This class of models may consider the interplay between substrates uptake, cell growth rate, gene expression and interaction with the cell host caused by competition for cellular resources. Therefore, they have the potential to deal with the related issues of dynamic gene regulation, cell resource allocation and the varying industry-scale bioreactor environment.

As an intermediate modeling strategy, whole-cell models connecting gene expression, metabolism and growth, such as the coarse-grained self-replicator models, take into account the dynamic feedback from gene expression and growth to metabolism [36, 26].

So far, the main interest in the literature has been in determining the required optimal gene expression levels that fulfill the design specifications using any of the modeling approaches described above. The dynamic regulation of the specified expression levels has also been addressed both for heterologous protein expression systems [9] and for metabolic pathways [10]. Indeed, major improvements in yield, titer and productivity of engineered metabolic pathways can be accomplished by dynamic balancing of pathway gene expression [60, 41], where the application of dynamic feedback and feedforward regulation of gene expression addresses the robustness pitfalls of static regulation.

It is known that weakly expressed endogenous genes exhibit low RNA polymerase/ribosome ratios, while strongly expressed genes have higher RNA polymerase/ribosome ratios, as this is metabolically efficient [38]. However, to the author's knowledge, little research has been done about the differential roles that gene transcription and translation can play in shaping the trade-offs between titer, yield and productivity at the bioreactor level.

Here, we use a multi-scale model that connects the dynamics of the population of cells in the bioreactor, those of substrate uptake with the dynamic interaction between the host and the synthetic circuits expressing proteins of interest. Our model predicts the cell growth rate and the distribution of cell mass between the protein of interest and the host ones as a function of the substrate uptake and the main lab-accessible gene expression-related characteristics: promoter strength, gene copy number and ribosome binding site (RBS) strength. While in this work we do not consider metabolism explicitly, we do it implicitly through the substrate dynamics. For a wide range of gene expression levels, we evaluate the differential roles of gene transcription and translation in shaping the trade-offs between titer, yield and productivity rate for the three main operational modes of industrial bioreactors. We also evaluate the sensitivity of the mapping between the expression and the TRY spaces as a function of variations in the substrate availability.

4.3 Methods

In this section the mathematical models and methodological elements used in this work are presented. First we introduce our multi-scale model including the bioreactor model, the host model and the synthetic circuit model. Then we dive into the synthetic circuit gene expression space and into the bioreactor modes of operation. Finally, we present the TRY performance indices and their relative variation indices for substrate variations.

4.3.1 Multi-scale model

In order to take into account the different scales involved in the model of a bioprocess, we consider a multi-scale model integrating the different occurring phenomena. From top to bottom (Figure 4.1A), our model describes the interactions in the three levels considered: the biomass and substrate dynamics in the bioreactor, the dynamics of the host and those of the synthetic gene circuit expressing the proteins of interest.

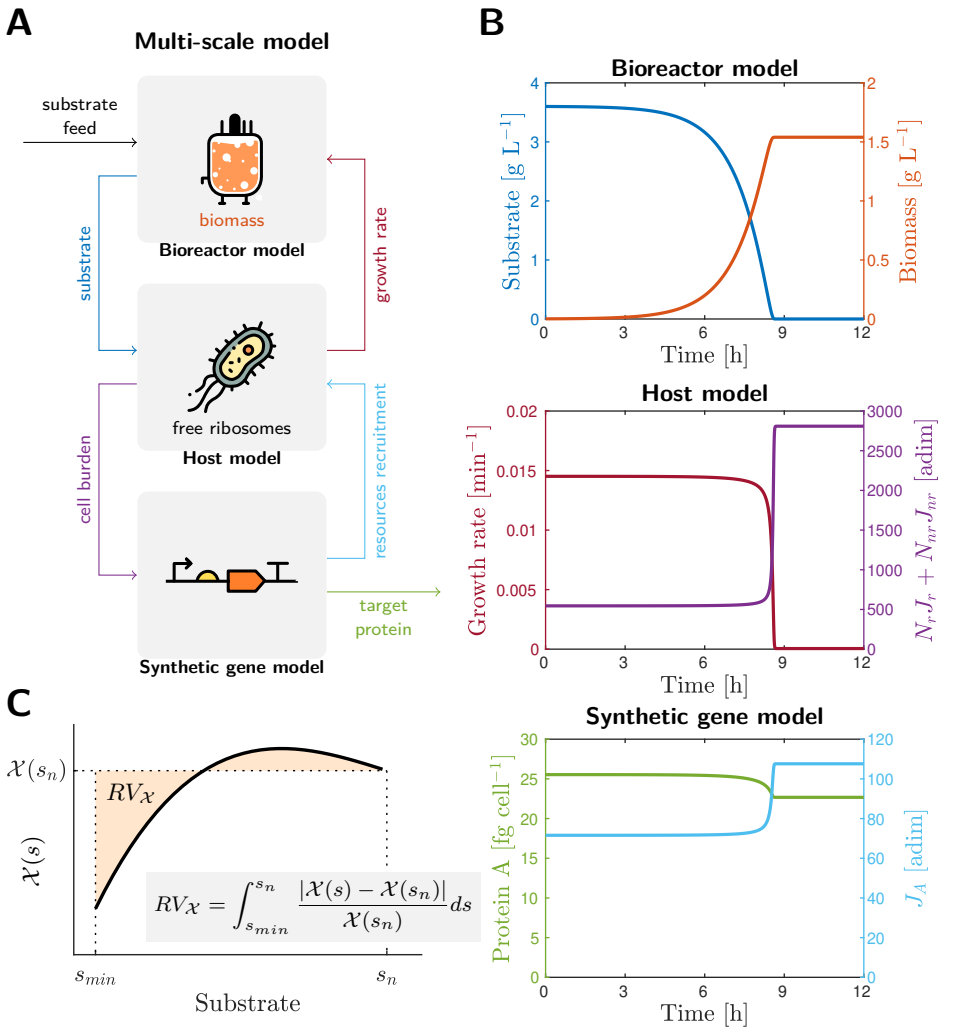

Figure 4.1: Bioprocess multi-scale model. (**A**). Schematic representation of the multi-scale model. The three levels considered including the bioreactor, the host cell and the synthetic circuit models and their interactions. (**B**). Simulation obtained with the multi-scale model. The colors correspond to the interactions in panel (A). An example of a batch operation, where the synthetic circuit parameters are $\omega_A = 50$, $k_b^A = 5$ and $k_u^A = 117$. Top panel shows the macroscopic scale with the substrate availability (left axis) and biomass concentration (right axis). Middle panel shows the growth rate (left axis) and a measure of the cell burden (right axis). Bottom panel shows the amount of protein A expressed in a cell (left axis) and its resource recruitment strength J_A (right axis). (**C**). Titer, productivity rate and yield relative variation indices for substrate variations. Shaded area RV_χ represents the integral from minimum substrate s_{min} to the nominal substrate s_n of the difference between the nominal index value and the actual value.

At the top of our multi-scale model we find the macroscopic dynamics of the substrate and biomass in the bioreactor (Figure 4.1A, Bioreactor model). This bioreactor model takes as inputs the substrate inflow and outflow, together with the specific growth rate of the cell population in the bioreactor, and it provides the concentration of available substrate and biomass in the bioreactor. These dynamics are defined by the following set of equations:

$$\dot{V} = F_{in} - F_{out}\,, \tag{4.1}$$

$$\dot{n} = \mu n - \frac{F_{in}}{V} n\,, \tag{4.2}$$

$$\dot{s} = \frac{F_{in}}{V}(s_{feed} - s) - y\mu x(n,\mu)\,, \tag{4.3}$$

where V is the volume of culture in the bioreactor (L), n is the concentration of cells in the bioreactor (cell $\cdot\, \mathrm{L}^{-1}$), and s is the concentration of available substrate ($\mathrm{g} \cdot \mathrm{L}^{-1}$). F_{in} and F_{out} represent the input and output fluxes ($\mathrm{L} \cdot \mathrm{min}^{-1}$), and μ the specific cell growth rate (min^{-1}). y is the biomass yield on glucose ($\mathrm{g}_{\mathrm{substrate}} \cdot \mathrm{g}^{-1}_{\mathrm{biomass}}$) and $x(n,\mu)$ is the concentration of biomass in the bioreactor ($\mathrm{g} \cdot \mathrm{L}^{-1}$).

At the intermediate level (Figure 4.1A) the host model represents the dynamics of the cell and takes as inputs the available substrate from the bioreactor model and a measure of the gene expression resources demanded by the synthetic gene circuit. This host model provides the cell-specific growth rate and a measure of the cell burden, and it is defined as follows:

$$\mu = \frac{m_{aa}}{m_h(\mu)} \nu_t(s) \phi_t^h(\mu, r, s) \phi_m r_t(m_{r_t})\,, \tag{4.4}$$

$$r = \frac{\phi_m r_t(m_r)}{1 + W_{sum}(\mu, r, s)}\,, \tag{4.5}$$

$$\dot{m}_{r_t} = \left[m_h(\mu) \frac{N_r J_r(\mu, r, s)}{N_r J_r(\mu, r, s) + N_{nr} J_{nr}(\mu, r, s)} - m_{r_t} \right] \mu\,, \tag{4.6}$$

$$\dot{m}_{nr} = \left[m_h(\mu) \frac{N_{nr} J_{nr}(\mu, r, s)}{N_r J_r(\mu, r, s) + N_{nr} J_{nr}(\mu, r, s)} - m_{nr} \right] \mu\,, \tag{4.7}$$

where the main variables can be found in Table 4.1, and the terms $J_r(\mu, r, s)$, $J_{nr}(\mu, r, s)$ are the resources recruitment strengths for the host ribosomal and non-ribosomal lumped ensembles of genes (see Chapter 2), and $W_{sum}(\mu, r, s) = \sum_{i=r,nr,A} N_i J_i(\mu, r, s)$ is the total resources recruitment strength of the cell which

represents the cell burden. Recall these are the key functional coefficients that allow us to explain the distribution of resources in the host and the relationship between the usage of resources, cell growth and protein productivity. The parameters used in this work were fitted from experimental data from [42, 15] and correspond to the wild-type *E. coli* K-12 strain MG1655 [59] and *E. coli* B/r strain.

The internal variables in the model and the values of the parameter can be found in Table 4.2 and Table 4.3 respectively.

Table 4.1: States and main variables of the host model.

Name	Description	Units
r	Free mature ribosomes in the cell.	$\text{molec} \cdot \text{cell}^{-1}$
m_{r_t}	Total mass of ribosomal proteins in the cell.	$\text{fg} \cdot \text{cell}^{-1}$
m_{nr}	Total mass of non ribosomal proteins in the cell.	$\text{fg} \cdot \text{cell}^{-1}$
m_A	Total mass of protein A in the cell.	$\text{fg} \cdot \text{cell}^{-1}$

Finally, at the bottom level, a model of the synthetic gene circuits considers the interaction dynamics with the host and estimates production of the mass fraction of the proteins of interest (4.8). In this work, as an example, we consider the expression of a generic theoretical gene A. More complex synthetic circuits or metabolic pathways with their corresponding enzymes can be easily incorporated into the model by adding more equations such as (4.8).

$$\dot{m}_A = \left[m_h(\mu) \frac{N_A J_A(\mu, r, s)}{N_r J_r(\mu, r, s) + N_{nr} J_{nr}(\mu, r, s)} - m_A \right] \mu \,, \tag{4.8}$$

where m_A is the total mass of protein A in the cell, N_A is the number of copies of gene A, and $J_A(\mu, r, s)$ is the resources recruitment strength of the gene A.

The resources recruitment strength of a given gene, gene A in this case, $J_A(\mu, r, s)$ can be understood as a measure of its eagerness to capture cell resources to get expressed. As seen from Table 4.2, it can be expressed as:

$$J_A(\mu, r, s) = E_{mA}(l_{pA}, l_e) \frac{\omega_A}{\frac{d_m^A}{K_{C_0}^A(s)} + \mu r} \,, \tag{4.9}$$

Table 4.2: Internal variables of the model.

Name	Description	Equation	Units
$\nu_t(s)$	Effective translation rate per ribosome.	$\nu_t(s) = \nu_{max} s/(s + K_s)$	$\text{aa} \cdot \text{min}^{-1}$
$k_e(s)$	Translation initation rate.	$k_e(s) = \nu(s)/l_e$	min^{-1}
$W_{sum}(\mu, r, s)$	Total sum of the resource recruitment strengths of all genes.	$W_{sum}(\mu, r, s) = \sum_{i=\{r,nr,A\}} N_i J_i(\mu, r, s) N_r$	adim
$K^r_{C_0}(s)$	Effective RBS affinity of ribosomal mRNA.	$K^r_{C_0}(s) = k^r_b/(k^r_u + k_e(s))$	$\text{cell} \cdot \text{molec}^{-1}$
E^r_m	Ribosomes density related term for ribsomal mRNA.	$E^r_m = 3.459$	adim
$J_r(\mu, r, s)$	Average J value of one ribosomal *E. coli* gene.	$J_r(\mu, r, s) = E^r_m \omega_r/(d^r_m/K^r_{C_0}(s) + \mu r)$	adim
$r_t(m_r)$	Number of mature and inmmature ribosomes.	$r_t(m_r) = m_r/r_w$	$\text{molec} \cdot \text{cell}^{-1}$
$\phi^h_t(\mu, r, s)$	Fraction of mature ribosomes translating endogenous transcripts.	$\phi^t_h(\mu, r, s) = (N_r J_r + N_{nr} J_{nr})/(1 + W_{sum}(\mu, r, s))$	adim
$K^{nr}_{C_0}(s)$	Effective RBS affinity of non-ribosomal mRNA.	$K^{nr}_{C_0} = k^{nr}_b/(k^{nr}_u + k_e(s))$	$\text{cell} \cdot \text{molec}^{-1}$
E^{nr}_m	Ribosomes density related term for non-ribsomal mRNA.	$E^{nr}_m = 6.3492$	adim
$J_{nr}(\mu, r, s)$	Average J value of one non-ribosomal *E. coli* gene.	$J_{nr}(\mu, r, s) = E^{nr}_m \omega_{nr}/(d^{nr}_m/K^{nr}_{C_0}(s) + \mu r)$	adim
$K^A_{C_0}(s)$	Effective RBS affinity of protein A mRNA.	$K^A_{C_0}(s) = k^A_b/(k^A_u + k_e(s))$	$\text{cell} \cdot \text{molec}^{-1}$
E^A_m	Ribosomes density related term for protein A mRNA.	$E^A_m = 0.62 l^A_p/l_e$	adim
$J_A(\mu, r, s)$	Average J value of one protein A gene.	$J_A(\mu, r, s) = E^A_m \omega_A/(d^A_m/K^A_{C_0}(s) + \mu r)$	adim
$m_h(\mu)$	Total host protein mass.	$m_h(\mu) = c_1\mu^2 + c_2\mu + c_3$	$\text{fg} \cdot \text{cell}^{-1}$
$x(n, \mu)$	Concentration of biomass in the biorreactor.	$x(n, \mu) = n\,(m_{r_t} + m_{nr} + m_A)$	$\text{g} \cdot \text{L}^{-1}$

Table 4.3: Parameters of the model.

Name	Description	Value	Reference
K_s	Half activation threshold of growth rate.	$0.1802\,\mathrm{g \cdot L^{-1}}$	[129]
ν_{max}	Maximum effective translation rate per ribosome.	$1260\,\mathrm{aa \cdot min^{-1}}$	[77]
m_{aa}	Average amino acid mass.	$182.6 \times 10^{-9}\,\mathrm{fg \cdot aa^{-1}}$	[113]
l_e	Ribosome occupancy length.	$25\,\mathrm{aa}$	estimated [32, 30, 82, 109]
ϕ_m	Fraction of mature available ribosomes relative to the total.	$0.9473\,\mathrm{adim}$	[100, 14, 15] *
l_p^r	Mean length of ribosomal proteins.	$195\,\mathrm{aa}$	calculated from [42]
d_m^r	Mean degradation rate of ribosomal mRNA.	$0.16\,\mathrm{min^{-1}}$	calculated from [42]
k_u^r	Dissociation rate RBS-ribosome for ribosomal mRNA.	$119.7956\,\mathrm{min^{-1}}$	[100] *
k_b^r	Association rate RBS-ribosome for ribosomal mRNA.	$4.7627\,\mathrm{cell \cdot min^{-1} \cdot molec^{-1}}$	[100] *
N_r	Number of proteins that make up a ribosome.	$55\,\mathrm{adim}$	[100]
ω_r	Average transcription rate for ribosomal proteins.	$7.33\,\mathrm{molec \cdot min^{-1} \cdot cell^{-1}}$	[100] *
r_w	Weight of a ribosome.	$0.0045\,\mathrm{fg}$	[15]
l_p^{nr}	Mean length of non-ribosomal proteins.	$333\,\mathrm{aa}$	calculated from [42]
d_m^{nr}	Mean degradation rate of non-ribosomal mRNA.	$0.2\,\mathrm{min^{-1}}$	calculated from [42]

Table 4.3: *Continuation.*

Name	Description	Value	Reference
k_u^{nr}	Dissociation rate RBS-ribosome for non-ribosomal mRNA.	$10.0454\,\mathrm{min}^{-1}$	[100] *
k_b^{nr}	Association rate RBS-ribosome for non-ribosomal mRNA.	$12.4404\,\mathrm{cell}\cdot\mathrm{min}^{-1}\cdot\mathrm{molec}^{-1}$	[100] *
N_{nr}	Number of non ribosomal proteins expressed at one time.	1735 adim	[100]
ω_{nr}	Average transcription rate for non ribosomal proteins.	$0.0361\,\mathrm{molec}\cdot\mathrm{min}^{-1}\cdot\mathrm{cell}^{-1}$	[100] *
l_p^A	Length of protein A.	195 aa	**
d_m^A	Mean degradation rate of protein A mRNA.	$0.16\,\mathrm{min}^{-1}$	**
k_u^A	Dissociation rate RBS-ribosome for protein A mRNA.	$6\text{–}135\,\mathrm{min}^{-1}$ †	[110, 53]
k_b^A	Association rate RBS-ribosome for protein A mRNA.	$3\text{–}15\,\mathrm{cell}\cdot\mathrm{min}^{-1}\cdot\mathrm{molec}^{-1}$ †	[110, 53, 95]
N_A	Number of copies of gene A.	1–70 adim †	**
ω_A	Average transcription rate for protein A.	$0\text{–}5\,\mathrm{molec}\cdot\mathrm{min}^{-1}\cdot\mathrm{cell}^{-1}$ †	**
c_1	First coefficient of mass equation.	$239\,089\,\mathrm{fg}\cdot\mathrm{cell}^{-1}\cdot\mathrm{min}^2$	[100] *
c_2	Second coefficient of mass equation.	$7432\,\mathrm{fg}\cdot\mathrm{cell}^{-1}\cdot\mathrm{min}$	[100] *
c_3	Third coefficient of mass equation.	$37.06\,\mathrm{fg}\cdot\mathrm{cell}^{-1}$	[100] *
y	Biomass yield on glucose.	$2.2\,\mathrm{g}_{\mathrm{substrate}}\cdot\mathrm{g}_{\mathrm{biomass}}^{-1}$	[76]
s_{feed}	Fresh media substrate concentration.	$3.6\,\mathrm{g}\cdot\mathrm{L}^{-1}$	[129]

* These parameters were re-optimized following the methods described in [100] to better fit the wild-type at low growth rate, since that range is the most relevant for this work. ** Without loss of generality in the results, we choose l_p^A and d_m^A to be equal to the ribosomal parameters, and the range of N_A and ω_A to be in the order of ribosomal and non-ribosomal parameters. † These parameters define the gene expression space and their value varies in simulations between the minimum value (first value in parentheses) and the maximum value (last value in parentheses).

where l_p^A/l_e is the ribosomes density (protein length over ribosomes occupancy length), the product μr of growth rate and number of free ribosomes is the flux of free resources, and d_m^A is the degradation rate of the transcript. The transcription rate is ω_A, the RBS strength is $K_{C_0}^A(s)$, and the gene copy number is N_A. In this work we analyze the roles that gene transcription and translation may play in shaping the trade-offs between titer, yield and productivity at the bioreactor level. Thus, on the one hand we will consider the transcription rate per gene ω_A times the gene copy number N_A. Notice ω_A is directly related to the promoter strength. On the other hand we consider the RBS strength. The effective translation rate depends on the availability of the substrate.

Recall from Chapter 2 that the RBS strength for a gene expressing the protein A is defined as:

$$K_{C_0}^A(s) = \frac{k_b^A}{k_u^A + k_e(s)}\,, \tag{4.10}$$

where k_b^A and k_u^A are the association and dissociation rates between the ribosome binding site and the transcript, respectively, and $k_e(s)$ is a Monod-like function of the extracellular substrate s that models the translation initiation rate as a function of the maximum translation rate per ribosome, the ribosomes density, the availability of substrate that can be catabolized to build amino acids, and the affinity of the host for the substrate (see Tables 4.2 and 4.3 and reference [100]).

The fact that this RBS strength definition depends on the available substrate as in Equation (4.10) has several implications. First, notice that an infinite number of different combinations of k_b^A and k_u^A can give the same RBS strength. This is not different from the situation when considering the substrate-independent saturated RBS strength obtained for substrate saturation:

$$K_{C_0,\mathrm{sat}}^A = \frac{k_b^A}{k_u^A + \nu_{\max}/l_e}\,. \tag{4.11}$$

Most important, the sensitivity of the substrate-dependent RBS strength to changes in the substrate concentration in the culture depends on the actual values of k_b^A and k_u^A. Notice that for higher values of the saturated RBS strength, the variation of the substrate-dependent RBS-strength as a function of $k_e(s)$ is larger, with:

$$\frac{1}{K_{C_0}^A(s)}\frac{\partial K_{C_0}^A(s)}{\partial k_e(s)} = -\frac{1}{k_b^A}K_{C_0}^A(s)\,. \tag{4.12}$$

Therefore, in our analysis we evaluated the values of titer, productivity rate and yield for a nominal value of the substrate concentration in the culture (see Section 4.3.2) as a function of the expression space determined by the parameters $N_A\omega_A$, k_b^A and k_u^A. In addition, we also evaluated the sensitivity of the TRY space to changes in the substrate concentration as a function of the expression space.

4.3.2 Bioreactor model—modes of operation

The extracellular macroscopic inflow F_{in} and outflow F_{out} in model (4.1)–(4.3) depend on the mode of operation considered for the bioreactor: batch, fedbatch or continuous.

We assumed a well-mixed, homogeneous culture and the values of parameters defined in Table 4.3. The initial conditions for the concentration of cells in the bioreactor, the substrate concentration and the culture initial volume were set to industrially plausible values for a small size bioreactor with volume up to 10 L: $n(0) = 5 \times 10^9\,\mathrm{cell}\cdot\mathrm{L}^{-1}$, $s(0) = 3.6\,\mathrm{g}\cdot\mathrm{L}^{-1}$ and $V(0) = 1\,\mathrm{L}$, respectively. To calculate the initial conditions for the host cell parameters μ, r, m_r, m_{nr} and m_A, we ran a simulation for constant substrate $s = s(0)$, and we obtained the steady-state values for all the variables in the system. These were used as initial conditions for the host cell parameters.

For the batch mode of operation in the bioreactor, we set $F_{in} = F_{out} = 0$, and we ran the simulations until the substrate concentration in the bioreactor decreased below 2% of its initial value.

For the fedbatch and continuous modes, substrate feeding policies were applied such that the concentration of substrate in the bioreactor was kept constant to the nominal value $s_n = s(0) = 3.6\,\mathrm{g}\cdot\mathrm{L}^{-1}$. For the fedbatch mode we used $F_{out} = 0$ and the substrate feeding law

$$F_{in}(\mu, n, V, s) = y\mu x(\mu, n)\frac{V}{s_{feed} - s}\,. \tag{4.13}$$

The feeding law (4.13) makes the substrate concentration in the bioreactor to remain constant and equal to the initial one [129]. For the substrate feeding concentration we used $s_{feed} = 180.156\,\mathrm{g}\cdot\mathrm{L}^{-1}$, and the bioreactor was fed until the culture volume reached 10 L. At this point the feeding inflow was set to zero ($F_{in} = 0$), and the simulation was continued in batch mode until the substrate concentration in the bioreactor decreased below 2% of its initial value.

Finally, for continuous mode, we used the feeding law $F_{out} = F_{in} = VD(\mu, n)$ with:

$$D(\mu, n) = \mu \frac{x(n, \mu)}{x_{ref}}\,, \tag{4.14}$$

where $D(\mu, n)$ is the dilution and x_{ref} a reference value for the concentration of biomass in the bioreactor. We used $x_{ref} = 1\,\mathrm{g} \cdot \mathrm{L}^{-1}$ to get a production comparable with batch mode. When the reference is achieved, the concentration of glucose is kept at $s = 1.4\,\mathrm{g} \cdot \mathrm{L}^{-1}$. However, the biomass starts in the simulations approximately at $0.01\,\mathrm{g} \cdot \mathrm{L}^{-1}$, and it takes almost all the simulation time to get to the reference. Therefore, for most of the simulation time the substrate was close to $s_n = 3.6\,\mathrm{g} \cdot \mathrm{L}^{-1}$ as in fedbatch case and in the exponential phase of the batch one.

The simulation ran for a time interval equivalent to the turnover time, so that 10 L of culture was introduced and removed form the bioreactor.

Notice the conditions in the three modes of operation of the bioreactor were chosen so that the metabolic state of the cells in all three cases were equivalent so as to achieve a fair comparison.

To keep track of the volume of media and product A added and removed from the bioreactor we extended the model with the expressions:

$$\dot{V}_{feed} = F_{in}\,, \tag{4.15}$$

$$\dot{V}_{out} = F_{out}\,, \tag{4.16}$$

$$\dot{S}_{out} = F_{out}\, s\,, \tag{4.17}$$

$$\dot{M}_{A_{out}} = F_{out}\, n\, m_A,\,, \tag{4.18}$$

where V_{feed} is the total volume of media fed to the bioreactor, V_{out} the total volume of media removed from the bioreactor, S_{out} the total mass of substrate removed from the bioreactor and $M_{A_{out}}$ the total mass of protein A removed from the bioreactor. We assumed that the substrate removed from the bioreactor due to F_{out} is recovered and that cells stop growing once the substrate is removed.

4.3.3 TRY performance indices

Three measures are commonly used to determine the performance of a bioprocess [129]: titer T, productivity rate R and yield Y. We evaluated them for the production of the protein A in *E. coli* under the three modes of operation of the bioreactor and for different points covering the expression space (promoter and RBS strength).

The titer T, i.e., the concentration of the molecule of interest at the end of the bioprocess, was measured in units of grams per liter and was calculated as:

$$T = \frac{V^f n^f m_A^f + M_{A_{out}}^f}{V^f + V_{out}^f}\,, \tag{4.19}$$

where the superscript f indicates the final time of the fermentation, m_A is the amount of protein A mass in one cell and n is the concentration of cells in the bioreactor.

The average volumetric productivity rate R, i.e., the production of the molecule of interest (protein A) per time unit, was measured in units of grams per liter per hour and calculated as:

$$R = \frac{T}{t^f}\,, \tag{4.20}$$

where t^f is the final time of the fermentation.

Finally, the yield, i.e., the conversion factor of substrate into the product (protein or metabolite) of interest, was measured in units of product grams per substrate grams and was calculated as:

$$Y = \frac{T(V^f + V_{out}^f)}{s^0 V^0 - s^f V^f + s_{feed} V_{feed}^f - S_{out}^f}\,, \tag{4.21}$$

where the superscripts $\{0, f\}$ indicate the initial and final time of the fermentation process, V is the culture volume in the bioreactor, s the substrate concentration in the bioreactor and s_{feed} is the substrate concentration in the feed stream.

We used the nominal substrate concentration $s_n = 3.6\,\mathrm{g}\cdot\mathrm{L}^{-1}$ (in [129] the authors used 20 nM that is approximately $3.6\,\mathrm{g}\cdot\mathrm{L}^{-1}$) to evaluate the nominal values of

titer, productivity rate and yield as a function of the expression space determined by the parameters $N_A\omega_A$, k_b^A and k_u^A.

4.3.4 TRY relative variation indices for substrate variations

Fluctuations in the availability of limiting substrate are one of the main disturbances that cells may encounter within the bioreactor environment. Indeed, if the concentration of limiting substrate in the bioreactor decreases (e.g., because of mixing heterogeneity or saturation), the achieved TRY will change as the availability and distribution of cell resources vary.

To evaluate the sensitivity of the TRY space to changes in the substrate concentration in the culture as a function of the expression space, we used a measure of the relative variation of titer, rate and yield with respect to their nominal values for a range of variation in the substrate concentration, as it can be seen in Figure 4.1C. Thus, we defined the index:

$$RV_{\mathcal{X}} = \int_{s_{\min}}^{s_n} \frac{|\mathcal{X}(s) - \mathcal{X}(s_n)|}{\mathcal{X}(s_n)} ds \,, \tag{4.22}$$

where $\mathcal{X} = \{T, R, Y\}$ for titer, productivity rate and yield, respectively. The index is the cumulative relative difference between $\mathcal{X}$ evaluated for the nominal substrate concentration and for a range $s_{\min} \leq s \leq s_n$ using Equations (4.19)–(4.21).

4.4 Results

4.4.1 Nominal TRY as a function of the expression space and bioreactor operation

The analysis performed with different conditions of bioreactor operation and several combinations spanning the expression space reveals, as expected, that the cell growth rate decreases as the cell burden induced by higher expression levels of A increases. Figure 4.2 shows the results for the nominal variation of the TRY space as a function of the expression space for the three modes of operation in the bioreactor. We varied the parameters $N_A\omega_A$ and $K_{C_0}^A(s_n)$ of the expression space in a range from low to high expression of the protein A (see Figure 4.2C).

Titer and yield show a decreasing monotonous relationship with growth rate. Thus, as observed in Figure 4.2A, titer and yield decrease as the cell burden decreases (i.e., weak promoter and RBS strengths are used) and, consequently, the cell growth increases. Indeed, to obtain higher titer and yield, the cell must invest a higher fraction of its mass to synthesize the protein A. As a result, it uses fewer cell resources to build ribosomes, and the cell growth decreases [100].

On the contrary, the maximum productivity rate depends non-monotonously on the growth rate, achieving a maximum for $\mu \approx 0.075\,\mathrm{min}^{-1}$.

Thus, we find the same qualitative trade-offs between titer, yield and rate as a function of growth rate encountered when trying to optimize the metabolic flux towards a product of interest using metabolic flux analysis [51, 112]. There is a trade-off between titer and yield on the one hand, and productivity rate on the other. High titer and yield cannot be attained without eventually decreasing the productivity rate.

Additionally, as expected, the fedbatch mode achieves higher values of productivity rate and titer of the protein A for all the combinations in the expression space (see Figure 4.2A). This simply reflects that in fedbatch mode the total amount of substrate fed to the bioreactor is larger than in batch and continuous modes. Yet, the normalized results show no relevant differences (Figure 4.2B). That is, the normalized titer rate and yield do not depend on the mode of operation of the bioreactor but only on the promoter and RBS strengths in the expression space.

This result provides a principle of space-scale separation for multi-scale models. For a given substrate availability, and assuming homogeneity in the population of cells in the bioreactor, the cell growth for each individual local cell and the mass of heterologous proteins it will express depend on the interactions between the cell host and the genes, being independent of the way the substrate is fed to the population of cells. The bioreactor mode of operation will affect the total amount of substrate fed during the fermentation process and, therefore, the size of the population of cells. Although the geometry and other physical characteristics of the bioreactor are not considered here, they could be important to provide a way to characterize heterogeneous distributions of the limiting substrate within the bioreactor. Individual cells with different substrate concentrations will be subject to different interactions between the host and the synthetic circuit, and then they might synthesize the heterologous proteins at different rates.

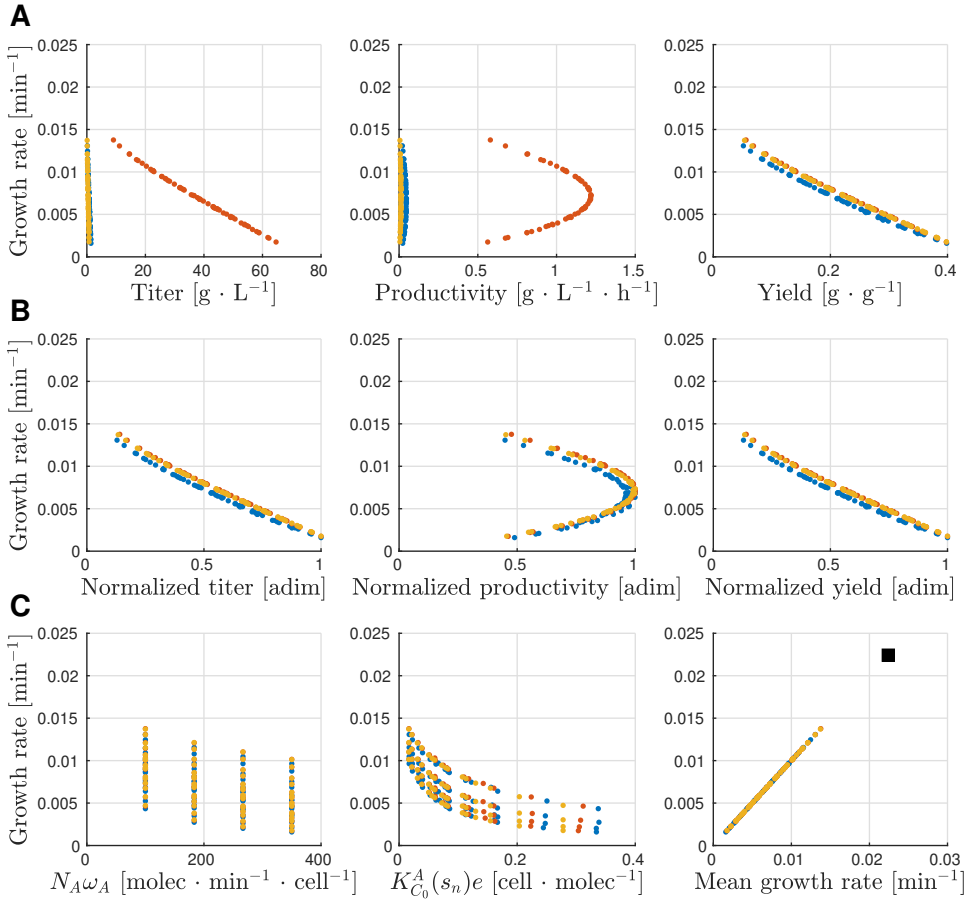

Figure 4.2: Titer, productivity rate and yield of protein A across the gene expression space for batch, fedbatch and continuous operation obtained by simulation with the multi-scale model. All plots share the growth rate as the y-axis. Each dot corresponds to a different combination of transcription $N_A\omega_A$ and RBS $K^A_{C_0}(s_n)$ strengths. The dot color corresponds to the following: blue, batch; red, fedbatch; yellow, continuous. (**A**) Absolute titer, productivity rate and yield. (**B**) Normalized titer, productivity rate and yield. (**C**) Shows the combinations of $N_A\omega_A$ and $K^A_{C_0}(s_n)$ used and their associated induced growth rate (note that we plot $K^A_{C_0}(s_n)$ at nominal substrate $s_n = 3.6\,\mathrm{g}\cdot\mathrm{L}^{-1}$, since $K^A_{C_0}(s)$ is not constant because s changes in simulation time). The wild-type growth rate is shown with a black square (right).

4.4.2 Mapping between gene expression and TRY spaces

In this section, we investigated different combinations of promoter and RBS strength and their corresponding regions in TRY space. Figure 4.3 shows the results of TRY space for fedbatch mode. The colored dots represent different combinations of $N_A\omega_A$ and $K^A_{C_0}(s_n)$ and TRY space, and the gray dots show the entire expression space.

At low protein expression levels, the promoter strength mainly determines the TRY values, while the RBS strength weakly influences its value. Figure 4.3A shows that the low protein expression region ($\mu \in [0.015, 0.025]$ min^{-1}) corresponds to low values of $N_A\omega_A$ (blue dots), and the medium to high values (yellow and red dots) are outside this region. However, varying the value of $K^A_{C_0}(s_n)$ (represented in the size of the dots) hardly changes the value of the TRY space, and they remain within the region of low protein expression levels. This distinct behavior between varying $N_A\omega_A$ and $K^A_{C_0}(s_n)$ is because in (4.9) the value of $J_A(\mu, r, s)$ is proportional to the value of $N_A\omega_A$. Whereas the value of $K^A_{C^0}(s)$ is in the denominator of $\omega_A/(d^A_m/K^A_{C_0}(s) + \mu r)$, so the value of μr limits the effect of $K^A_{C_0}(s_n)$ on increasing the value of $J_A(\mu, r, s)$. Moreover, in the case of low protein expression is when the value of μr is maximized with respect to the case of high protein expression, as there is less competition for recruiting ribosomes.

At high protein expression, both the promoter and RBS strength determine the TRY value. Figure 4.3A shows that the high protein expression region ($\mu \in [0, 0.015]$ min^{-1}) corresponds to medium to high values of $N_A\omega_A$ (yellow and red dots), as we discussed in the previous paragraph. Nevertheless, in contrast to the low protein expression case, the value of $K^A_{C_0}(s_n)$ does influence the value of the TRY measures (size of the yellow and red dots). In particular, high values of $N_A\omega_A$ and $K^A_{C_0}(s_n)$ (big red dot) maximizes the values of titer and yield. Otherwise, balanced combinations of $N_A\omega_A$ and $K^A_{C_0}(s_n)$ (medium yellow dot or small red dot) maximize the value of productivity rate.

In summary, on the one hand, increasing the promoter strength makes the value of $J_A(\mu, r, s)$ rise proportionally, increasing the protein A expression level (if there are cellular resources that can be recruited by increasing the value of $J_A(\mu, r, s)$). On the other hand, increasing the RBS strength will not always increase $J_A(\mu, r, s)$ since its effect is limited by the value of μr. Thus, the promoter strength mainly determines the TRY values, and the RBS strength can adjust it with some limitations. Finally, there are combinations of different promoter and RBS strength values that result in the same protein expression value for a given substrate, which what we focus on in the next section.

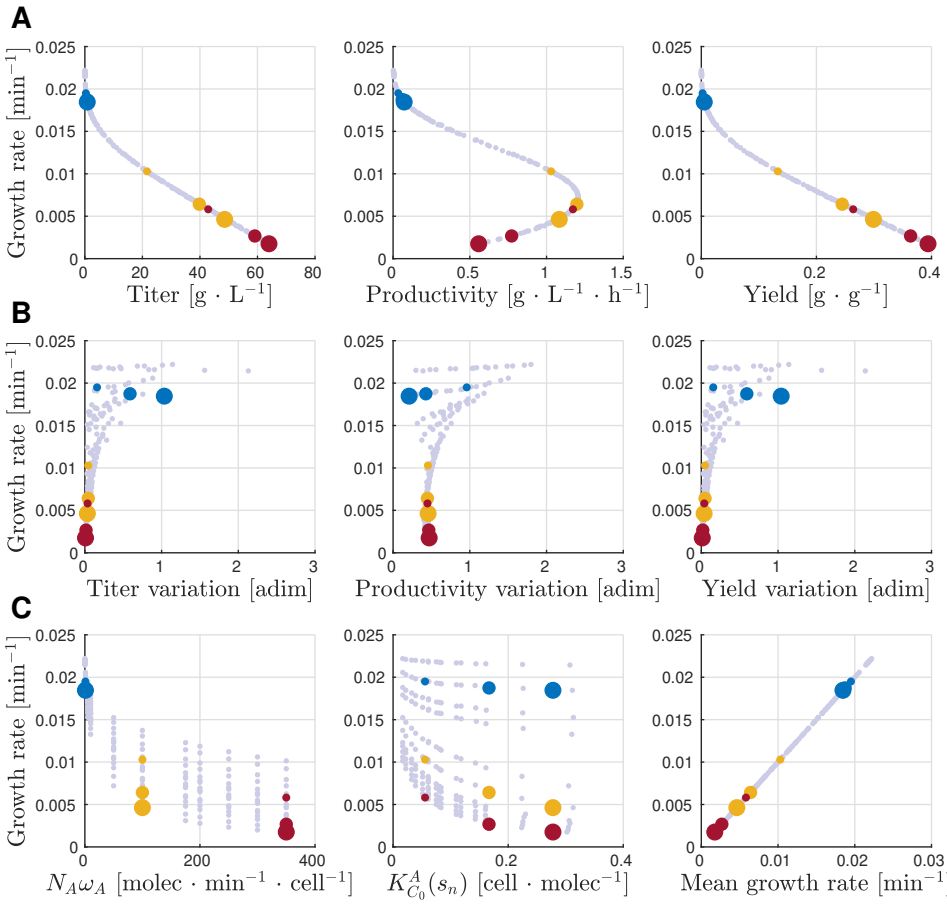

Figure 4.3: Titer, productivity rate and yield (TRY) values and relative variation indices of protein A for all the gene expression space in fed-batch operation obtained by simulation with the multi-scale model. All the plots share the growth rate as the y-axis. Each one dot corresponds to a different combination of transcription rate $N_A\omega_A$ and effective RBS $K^A_{C_0}(s_n)$ (note that we plot $K^A_{C_0}(s_n)$ at nominal substrate $s_n = 3.6\,\mathrm{g}\cdot\mathrm{L}^{-1}$, since $K^A_{C_0}(s)$ is not constant because s changes in simulation time). The colors blue, yellow and red represent low, medium and high $N_A\omega_A$ respectively, while the sizes of the markers are small, medium and big and show low, medium and high $K^A_{C_0}(s_n)$. The gray color represents the rest of $N_A\omega_A$ and $K^A_{C_0}(s_n)$ combinations. (**A**) TRY measurements: titer, productivity rate, and yield. (**B**) TRY relative variation indices for substrate variations. (**C**) Transcription rate $N_A\omega_A$ and the effective RBS strength $K^A_{C_0}(s_n)$.

4.4.3 TRY relative variation indices are fundamentally different at low and high cell burden

Different $N_A\omega_A$ and $K^A_{C_0}(s_n)$ configurations can result in the same value of $J_A(\mu, r, s)$ for a given substrate, and hence the same TRY values. Equation (4.9) suggests these different configurations could have different sensitivity to variations in s and μr. To test this, we used the sensitivity indices defined in (4.22) to analyze how much the TRY space varies as a function of variations in the substrate availability for different configurations in the expression space.

Figure 4.3B shows that, at low protein expression ($\mu \in [0.015, 0.025]$ min^{-1}), the relative variation indices $RV_{\mathcal{X}}$ span a wide range of values, from being close to zero (little or no variation in TRY measures due to variations in substrate) to values greater than 1.0 (large variation in TRY measures due to variations in substrate). This result confirms that, at low protein expression, different configurations of $N_A\omega_A$ and $K^A_{C_0}(s_n)$ have different sensitivities in variations of s.

However, Figure 4.3B also shows that increasing protein expression reduces the range of possible $RV_{\mathcal{X}}$ values. In particular, at high protein expression, $\mu \in [0, 0.005]$ min^{-1}, all $N_A\omega_A$ and $K^A_{C_0}(s_n)$ configurations converge to the same $RV_{\mathcal{X}}$ values. Specifically, RV_T and RV_Y approach to zero, while RV_R approaches to 0.5. This result shows that, at high protein expression, it is indifferent the $N_a\omega_a$ and $K^A_{C_0}(s_n)$ configuration; all $N_a\omega_a$ and $K^A_{C_0}(s_n)$ configurations tend to the same sensitivity with respect to substrate.

The TRY relative variation indices are fundamentally different at low and high cell burden. This is because in the case of high protein expression, μr becomes negligible with respect to $d^A_m/K^A_{C_0}(s)$, so that the $J_A(\mu, r, s)$ equation can be simplified into

$$J_A(s) = 0.62\frac{l^A_p}{l_e}\frac{\omega_A}{d^A_m}K^A_{C_0}(s)\,. \tag{4.23}$$

Then, for high protein expression, $J_A(s)$ depends only on the value of s, and it is independent of μr. This explains why at a high level of expression all the combinations of $N_A\omega_A$ and $K^A_{C_0}(s_n)$ tend to the same $RV_{\mathcal{X}}$ values; by simplifying Equation (4.9) into Equation (4.23), the effect of $N_A\omega_A$ and $K^A_{C_0}(s)$ in the sensitivity of $J_A(s)$ becomes similar.

4.4.4 There exists a trade-off in the relative variation indexes

In the previous sections, we have shown how $N_A\omega_A$ and $K^A_{C_0}(s_n)$ determine trade-offs in the measures of titer, productivity rate and yield; in this section we investigated whether there is also a trade-off in the relative variation indexes and how $N_A\omega_A$ and $K^A_{C_0}(s_n)$ affect it. Next, we show which conditions are necessary to minimize each of the relative variation indexes.

The titer and yield relative variation indexes are zero when the mass fraction between host cells and protein A is constant with changes in substrate. The titer and yield equations are independent of growth rate; therefore, for their relative variation indexes to be zero, it is sufficient for the cell to invest the same mass fraction as protein A regardless of substrate changes.

The productivity relative variation index is zero when the product of growth rate by the fraction of mass invested as protein A is constant with changes in substrate. The productivity equation depends on the growth rate; then, for its index to be zero, it is not sufficient to achieve a constant mass fraction invested as protein A (as was the case with titer and yield). To achieve a productivity relative variation index of zero, it is necessary that the product of mass fraction and growth rate remains constant.

Therefore, there exists a trade-off between the titer and yield indexes versus the productivity index since they require different conditions to minimize their values that are incompatible with each other. Figure 4.3B shows that none of the combinations of the colored dots is able to make all three indices zero at the same time (this also true for the gray dots, although it is not shown in the graph).

To take a deeper look into the relative variation index $RV_{\mathcal{X}}$ and their meaning, it is necessary to analyze the variation of indices for changing levels of substrate. Figure 4.4 shows that a change in the substrate can affect the TRY values by causing them to go down or up with respect to the TRY nominal value (it may even be the case that the TRY has one section where TRY goes up and another where TRY goes down). For example, Figure 4.4B shows that in the case of titer it is possible to select a value of k_u (solid blue line) that allows us to increase the titer when the substrate decreases, and we can also select a different value of k_u (dotted blue line) that decreases the titer when the substrate decreases.

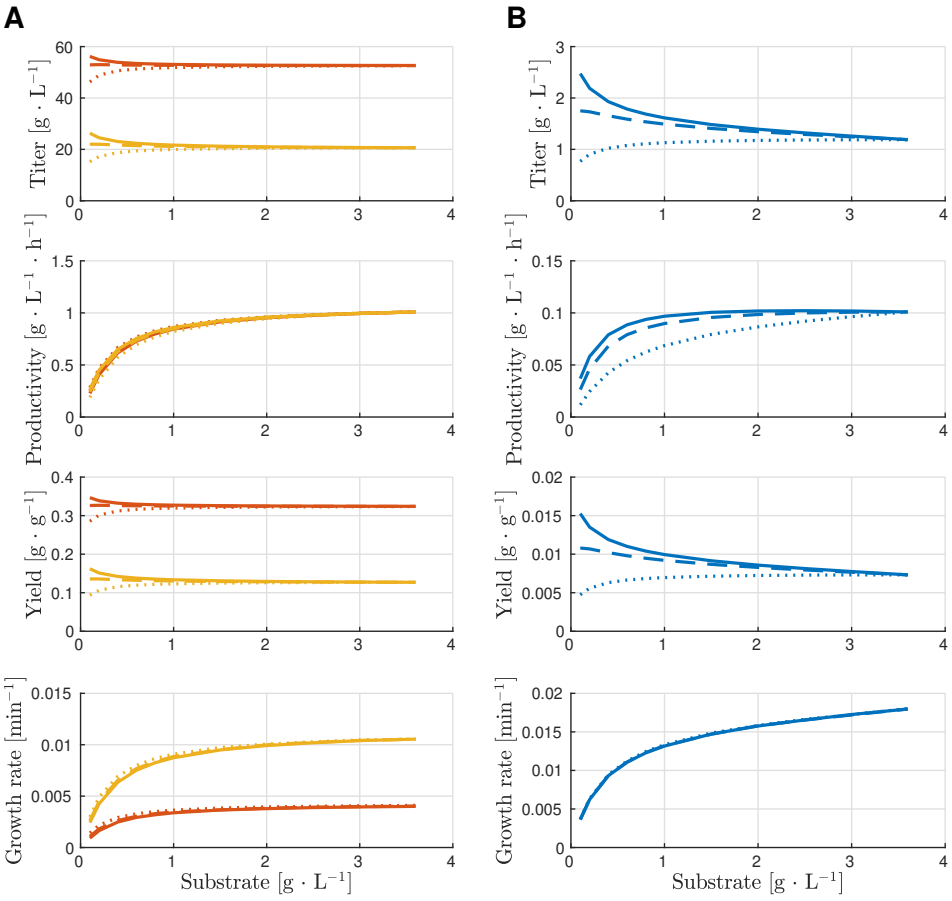

Figure 4.4: Titer, productivity rate and yield (TRY) at different nominal substrate concentrations for high and low protein expression levels for fedbatch operation. Orange and yellow (high and medium burden) lines correspond to different $N_A\omega_A$ and $K^A_{C_0}(s_n)$ combinations that achieved the same productivity value of $1\,\mathrm{g}\cdot\mathrm{L}^{-1}\cdot\mathrm{h}^{-1}$ at the nominal substrate $3.6\,\mathrm{g}\cdot\mathrm{L}^{-1}$, while blue (low burden) lines correspond to combinations that achieved $0.1\,\mathrm{g}\cdot\mathrm{L}^{-1}\cdot\mathrm{h}^{-1}$. We chose $k_b = 15$ and $k_u = \{6, 20, 135\}$ for all the combinations (red, yellow, and blue), and we tuned the value of $N_A\omega_A$ to achieve the desired productivity level. Low levels of k_u are drawn with dotted lines, mid levels with dashed lines and high levels with solid lines. Then, for these combinations, we reduced the nominal substrate up to $0.1\,\mathrm{g}\cdot\mathrm{L}^{-1}$ to see its effect on titer, productivity rate, yield and growth rate. (**A**) TRY for medium and high burden $N_A\omega_A$ and $K^A_{C_0}(s_n)$ combinations. (**B**) TRY for low burden $N_A\omega_A$ and $K^A_{C_0}(s_n)$ combinations.

From this, we can see that both the absolute variation and the direction of change in production are important. For industrial production purposes it is not the same for production to go down than for production to go up with respect to the nominal value. Then, there may be different combinations of $N_A\omega_A$ and $K^A_{C_0}(s_n)$ that achieve the same rate of change, but they may have different directionality, and then one combination can be more favorable than the other. This highlights the importance of the selection process for the values of $N_A\omega_A$ and $K^A_{C_0}(s_n)$.

4.5 Discussion and conclusions

In this work, we demonstrate the need for cell burden models as well as their utility. Our results show that, at low expression levels, gene transcription mainly defined TRY, and gene translation had a limited effect; whereas, at high expression levels, TRY depended on the product of both, in agreement with experiments in the literature [121]. Our model can be used to predict cell growth rate and cell mass distribution between enzymes of interest and host enzymes as a function of substrate uptake and the main lab-accessible gene expression-related characteristics: promoter strength, gene copy number and ribosome binding site strength. Multi-scale models, like the one presented here, incorporating the dynamics of (i) the cell population in the bioreactor, (ii) the substrate uptake and (iii) the interaction between the cell host and the expression of enzymes of interest, are useful to understand the differential roles of gene transcription and translation in shaping TRY trade-offs for a wide range of expression levels and the sensitivity of the TRY space to variations in substrate availability. Optimal gene expression is central for the development of both bacterial expression systems for heterologous protein production, and microbial cell factories for industrial metabolite production. With our approach it will be easier to fulfill industry-level overproduction demands optimally, as measured by the key performance metrics: titer, productivity rate and yield (TRY).

Chapter 5

Host-aware analysis of the antithetic controller

> "Because it has evolved to perform functions, biological circuitry is far from random or haphazard. It has a defined style, the style of systems that must function."
>
> —Uri Alon, *An Introduction to Systems Biology*

The contents of this chapter have been partially used in the following conference and journal publications:

- Santos-Navarro, F. N. et al. "Reference Conditioning Anti-windup for the Biomolecular Antithetic Controller". In: *IFAC-PapersOnLine* 52.26 (Jan. 2019), pp. 156–162. ISSN: 2405-8963. DOI: 10.1016/J.IFACOL.2019.12.251
- Boada, Y. et al. "Modeling and optimization of a molecular biocontroller for the regulation of complex metabolic pathways". In: *Frontiers in Molecular Biosciences* 9 (2022). DOI: 10.3389/fmolb.2022.801032

5.1 Abstract

This Chapter showcases the possibilities for analysis considering host-circuit and burden interactions for the antithetic controller using our host-aware model (see Chapter 2). We have used the *OneModel* implementation of the host-aware model as the base, and then we have extended the model to incorporate the antithetic controller. We show that increasing the antithetic controller's gain improves robustness and reduces position error (as expected), but there is burden-related limit gain at which the performance worsens. Counterintuitively, we show that reducing the dilution of the sigma and anti-sigma factors does not recover the integral action of the antithetic controller. We also show that the choice of RBS (ribosome binding site) strengths for expressing the antithetic controller has relevant effects on the robustness of the reference. This Chapter is an example of using our host-aware model and *OneModel* to analyze burden and host-circuit interactions in complex circuits. Improving host-aware models and their accessibility will allow us to design and implement complex synthetic genetic circuits successfully.

5.2 Introduction

The experimental implementation of synthetic genetic circuits depends, among other things, on (i) using robust circuits and (ii) taking into account host-circuit interactions. Robustness to parameter uncertainties and perturbations is a sought-after property. The antithetic controller, under ideal conditions, allows the robust expression of proteins of interest [16]. Models that consider competition for shared common cellular resources reveal the existence of host-circuit interactions that can lead to undesired interactions and compromise circuit performance and robustness [100, 125].

The antithetic controller behaves as an integral controller under ideal conditions. However, under realistic conditions (when the sigma and anti-sigma factors are diluted), the antithetic controller behaves as a proportional controller (or a leaky integrator), and its performance degrades [88]. Moreover, the burden introduced on the host by expressing the antithetic controller proteins is rarely considered, and it could also affect its performance. Analyzing the burden interactions requires incorporating the antithetic controller into a host-aware model. Most host-aware models are hard to extend, analyze, or are too simple to consider the details of a new added circuit. As a consequence, there is a gap in understanding the effects of cell burden on the performance of the antithetic controller.

This Chapter showcases the possibilities for analysis considering host-circuit and burden interactions for complex circuits using our host-aware model (see Chapter 2). We have used the *OneModel* implementation of the host-aware model as the base, and then we have extended the model to incorporate the antithetic controller. In particular, we analyse the effects of perturbations due to the expression of an exogenous protein B in the expression of a protein A regulated by an antithetic controller. This particular case exemplifies a common situation in a complex scenario and highlights the effects of the host-circuit interactions.

This Chapter should be understood as a computational experiment to emulate an antithetic controller under realistic conditions. In reality, the host cell activates and deactivates multiple genes in response to its external environment and internal cellular processes. This activation and deactivation cause the burden on the host cell to vary intermittently (independently of the operation of the antithetic controller). We emulated this gene activation and deactivation process in the host cell by expressing an additional protein B. This unrelated protein can be understood as an abstraction of then activation and deactivation of host genes and other synthetic genetic circuits (unrelated to the antithetic controller). The importance of this Chapter is not in the numerical results obtained, although we have used biological-feasible parameter values for the host cell and the synthetic circuit. This Chapter seeks to exemplify how protein expression in the host cell generates perturbations in the operation of the antithetic biomolecular controller and vice versa.

5.2.1 Ideal antithetic controller

This section is a brief introduction to the ideal antithetical controller. In the rest of this Chapter, we work with a more realistic version of it, but this ideal antithetic controller can be helpful to understanding the realistic version.

The antithetic controller is a synthetic gene system to robustly control the expression of a protein of interest [2]. This circuit is implemented using three genes coding three proteins: the sigma factor x_1, the anti-sigma factor x_2, and the protein of interest A. Normally, x_1 is constitutively expressed and induces the production of A. In turn, A activates the expression of x_2. Finally, x_1 and x_2 annihilate each other in a sequestration reaction forming an inert complex x_{12}, thus closing the loop.

The set of reactions of the ideal antithetic controller is as follow:

$$
\begin{aligned}
\emptyset &\xrightarrow{\theta_1} \mathrm{x}_1\,, \\
\mathrm{A} &\xrightarrow{\theta_2} \mathrm{A} + \mathrm{x}_2\,, \\
\mathrm{x}_1 &\xrightarrow{\theta_A} \mathrm{x}_1 + \mathrm{A}\,, \\
\mathrm{x}_1 + \mathrm{x}_2 &\xrightarrow{\gamma} \mathrm{x}_{12}\,, \\
\mathrm{A} &\xrightarrow{d_A} \emptyset\,,
\end{aligned}
\tag{5.1}
$$

where θ_1, θ_2, and θ_A are the translation rates of x_1, x_2, and A; γ is the antithetical sequestration rate constant; and d_A is the degradation rate of A (which may include active degradation and dilution due to cell growth).

From the set of reactions (5.1), applying mass-action kinetics, we obtain the following set of equations:

$$
\begin{aligned}
\dot{x}_1 &= \theta_1 - \gamma x_1 x_2\,, \\
\dot{x}_2 &= \theta_2 A - \gamma x_1 x_2\,, \\
\dot{x}_{12} &= \gamma x_1 x_2\,, \\
\dot{A} &= \theta_A x_1 - d_A A\,.
\end{aligned}
\tag{5.2}
$$

It is easy to show that A_{ss} (the value of A in steady-state) depends only on the rates of expressing the sigma and the anti-sigma factor:

$$
A_{ss} = \frac{\theta_1}{\theta_2}\,. \tag{5.3}
$$

Therefore, independent of the values of θ_A and d_A (which could be uncertain or time-varying), the protein of interest will be robustly expressed to the same value. This value of expression encoded in the antithetic controller is what we call the reference.

Note that the reactions and equations showed here are simple and do not consider the dilution of the sigma and anti-sigma factors, the saturation in the expression of the protein A, or the host-circuit interactions.

5.3 Methods

5.3.1 Model equations

This section shows the equations added to the host-aware model to model the expression of protein A, protein B, and the antithetic controller. Figure 5.1 shows a schematic representation of this circuit. The complete set of equations of the host-aware model is not listed here (see Chapter 2); we only show the extra equations we added to the host-aware model to perform the simulations in the Results section.

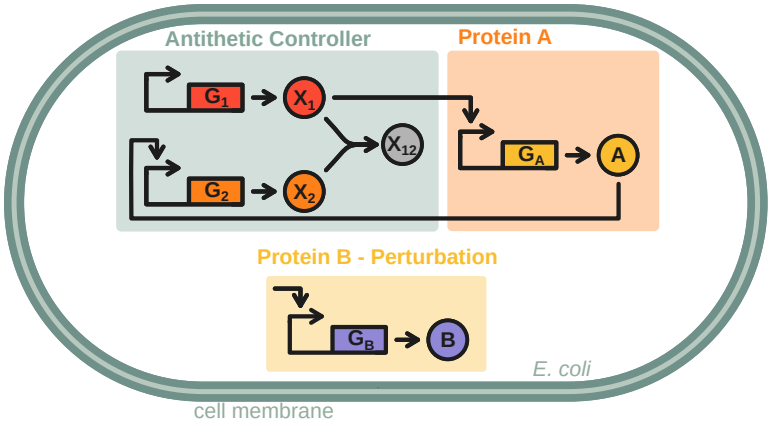

Figure 5.1: The schematic representation of the antithetic controller regulating the expression of protein A and, independently, the expression of protein B causing a perturbation in the host cell.

We must consider the dilution of the sigma and anti-sigma factors. This dilution can potentially be neglected for a specific set of parameters, but it is advisable to consider the dilution to model properly this type of synthetic circuits [10]. However, the host-aware model imposes us to consider the dilution of all proteins in the cell: not doing so would introduce errors between the growth equation and the mass equation that would break the model. Therefore, we have to consider the dilution of the sigma and anti-sigma factors of the antithetic controller, and thus the antithetic controller does not implement perfect integral action anymore.

Protein A is implemented as an induced expressed protein. The antithetic controller regulates protein A expression. The amount of sigma factor in the cell acts as a transcription factor for protein A expression with a Hill-like function.

Thus, there is a maximal transcription rate for expressing protein A when the protein A promoter is fully saturated with sigma factor.Yet, even if transcription saturates, the effective translation rate of protein A still depends on the available cell resources. Thus, it is affected by the varying cell burden introduced both by the antithetic controller itself and by other proteins being expressed in the cell, like protein B used in the analysis here.

Protein B is implemented as an induced expressed protein that is only active for a certain period. Protein B is used to produce a perturbation in the expression of protein A and the antithetic controller. However, protein B does not interact directly with protein A or with sigma or anti-sigma factors. It does it indirectly due to the burden it generates to the host.

The antithetic controller is implemented as a constitutively expressed protein for the sigma factor and an induced expressed protein for the anti-sigma factor. The amount of protein A induces the anti-sigma factor in the cell. For simplicity, we have considered no saturation in the transcription of the sigma and anti-sigma factors. We could consider the saturation, but that would lead to a more complex equation for the antithetic controller reference, and the results will be qualitatively equivalent to not considering saturation.

The sigma and anti-sigma factors form an inert bounded sigma::anti-sigma complex due to the annihilation reaction. This complex accumulates in the cell and is diluted by the growth rate.

The set of model equations is as follows:

$$\dot{A} = m_h(\mu)\frac{N_A J_A(\mu, r, s, x_1)}{N_r J_r(\mu, r, s) + N_{nr} J_{nr}(\mu, r, s)}\mu - A\mu\,, \tag{5.4}$$

$$\dot{B} = m_h(\mu)\frac{N_B J_B(\mu, r, s, t)}{N_r J_r(\mu, r, s) + N_{nr} J_{nr}(\mu, r, s)}\mu - B\mu\,, \tag{5.5}$$

$$\dot{x}_1 = m_h(\mu)\frac{N_{x_1} J_{x_1}(\mu, r, s)}{N_r J_r(\mu, r, s) + N_{nr} J_{nr}(\mu, r, s)}\mu - \gamma x_1 x_2 - x_1\mu\,, \tag{5.6}$$

$$\dot{x}_2 = m_h(\mu)\frac{N_{x_2} J_{x_2}(\mu, r, s, A)}{N_r J_r(\mu, r, s) + N_{nr} J_{nr}(\mu, r, s)}\mu - \gamma x_1 x_2 - x_2\mu\,, \tag{5.7}$$

$$\dot{x}_{12} = \gamma x_1 x_2 - x_{12}\mu\,, \tag{5.8}$$

where the equations for the resources recruitment strengths are:

$$J_A(\mu, r, s, x_1) = E_m(l_p^A, l_e)\frac{\omega_A}{d_m^A/K_{C_0}^A(s) + \mu r}\frac{x_1}{x_1 + h_A}\,, \tag{5.9}$$

$$J_B(\mu, r, s, t) = E_m(l_p^B, l_e)\frac{\omega_B}{d_m^A/K_{C_0}^B(s) + \mu r} f_B(t)\,, \tag{5.10}$$

$$J_{x_1}(\mu, r, s) = E_m(l_p^{x_1}, l_e)\frac{\omega_{x_1}}{d_m^{x_1}/K_{C_0}^{x_1}(s) + \mu r}\,, \tag{5.11}$$

$$J_{x_2}(\mu, r, s, A) = E_m(l_p^{x_2}, l_e)\frac{\omega_{x_2}}{d_m^{x_2}/K_{C_0}^{x_2}(s) + \mu r}\frac{A}{h_{x_2}} = J_{x_2}^{max}(\mu, r, s)\frac{A}{h_{x_2}}\,, \tag{5.12}$$

where $f_B(t)$ is a time dependent function which is equal to one during perturbation and zero the rest of the time. Recall the definition of the effective RBS strength:

$$K_{C_0}^i(s) = \frac{k_b^i}{k_u^i + k_e(s)}\,. \tag{5.13}$$

5.3.2 Antithetic controller reference

This section shows how to calculate the reference of the antithetic controller.

The antithetic controller reference equation is calculated under ideal conditions (when the dilution of the sigma and anti-sigma factors is neglected). From the equations (5.6) and (5.7), if we make zero the terms μx_1 and μx_2, in steady state, we obtain the general reference equation:

$$A_{ref} = \frac{J_{x_1}(\mu, r, s)}{J_{x_2}^{max}(\mu, r, s)} h_{x_2}\,, \tag{5.14}$$

where A_{ref} is the desired reference for A.

However, this general reference equation can be simplified, if we consider that $K_{C_0}^{x_1}(s)$ and $K_{C_0}^{x_2}(s)$ are equal (the RBS strength of the sigma and anti-sigma are equal, this is easy to achieve in the lab by using the same RBS for expressing sigma and anti-sigma), into:

$$A_{ref} = \frac{\omega_{x_1}}{\omega_{x_2}} h_{x_2}\,, \tag{5.15}$$

where the reference depends on the ratio of sigma to anti-sigma transcription rate and anti-sigma half-activation threshold. Note that the effects of substrate, growth rate, and free ribosome capacity cancel out in this case. The Equation (5.15) may be an oversimplification, but it is a straightforward way to know the reference value of the antithetic controller, and we can always use the general reference Equation (5.14).

5.3.3 States and parameters

This section shows the states (Table 5.1) we added to the host-aware model and the base parameters (Table 5.2) used in the simulations in the Results section. The tables do not show all the states or parameters of the host-aware model: the missing states and parameters can be found in Chapter 2.

We chose the parameter values for the antithetic controller, the protein A, and the protein B within the valid ranges of values discussed in Chapter 2. For simplicity, we choose the same promoter copy number, protein length, ribosome occupancy length, and RBS values for all the proteins of the circuit. Note that the units of the model states are the mass of each protein per cell (not molecules), so protein length is mainly irrelevant (only used to calculate $E_m(l_p^i, l_e)$) in this case. The half-activation threshold of the promoters indirectly captures the length of the proteins. Lastly, we chose the transcription rates of the antithetic controller not to burden the host in excess and to have a realistic reference. Protein A transcription was chosen to generate a low burden to the cell, and Protein B transcription generated a medium burden.

Table 5.1: States added to the host-aware model.

Name	Description	Units
A	Total mass of protein A in the cell.	$\text{fg} \cdot \text{cell}^{-1}$
B	Total mass of protein B proteins in the cell.	$\text{fg} \cdot \text{cell}^{-1}$
x_1	Total mass of sigma factor in the cell.	$\text{fg} \cdot \text{cell}^{-1}$
x_2	Total mass of anti-sigma factor in the cell.	$\text{fg} \cdot \text{cell}^{-1}$
x_{12}	Total mass of sigma::anti-sigma complex in the cell.	$\text{fg} \cdot \text{cell}^{-1}$

Table 5.2: Base model parameters used in simulations.

Name	Description	Value
N_i	Copy number of gene i.*	$1\,\text{copy}$
l_p^i	Protein length.*†	$195\,\text{aa}$
l_e	Ribosome occupancy length.*	$25\,\text{aa}$
d_m^i	mRNA degradation rate.*	$0.16\,\text{min}^{-1}$
k_b^i	Association rate RBS-ribosome of gene i mRNA.*	$4.7627\,\text{cell}\cdot\text{min}^{-1}\cdot\text{molec}^{-1}$
k_u^i	Dissociation rate RBS-ribosome of gene i mRNA.*	$119.7956\,\text{min}^{-1}$
ω_A	Transcription rate of protein A.	$20\,\text{min}^{-1}$
ω_B	Transcription rate of protein B.	$100\,\text{min}^{-1}$
ω_{x_1}	Transcription rate of sigma factor.	$15\,\text{min}^{-1}$
ω_{x_2}	Transcription rate of anti-sigma factor.	$3\,\text{min}^{-1}$
h_A	Half-activation threshold of protein A.	$2.5\,\text{fg}\cdot\text{cell}^{-1}$
h_{x_2}	Half-activation threshold of anti-sigma factor.	$1\,\text{fg}\cdot\text{cell}^{-1}$
γ	Antithetic sequestration affinity.	$10\,\text{fg}^{-1}\cdot\text{min}^{-1}$
s	Available substrate to the cell.	$3.6\,\text{g}\cdot\text{L}^{-1}$

The value of the parameters of this table are obtained from the values used in Chapter 2. * $i = \{A, B, x_1, x_2\}$, these parameters are used for protein A, protein B, sigma factor, and anti-sigma factor. † The model units are mass per cell (not molecules per cell), therefore the length of the protein is mainly irrelevant (only used to calculate $E_m(l_p^i, l_e)$).

5.3.4 Model implementation and simulation

The model was implemented using *OneModel*: the code can be found at `https://github.com/sb2cl/thesis_fernandonobel`. Code 5.18 shows a reduced version of the implementation of the model. We used *sbml2dae* to generate a *Matlab* implementation of the *OneModel* code, and then we used *Matlab* to do the simulations.

The initial condition of the simulations is the steady-state value without the perturbation; this way, we skip the dynamics of the antithetic controller, and we can focus on the effects of the perturbation. During the time interval 5–20 h, protein B is expressed to cause a perturbation in the host cell.

`antithetic_controller_reduced.onemodel`

```
import '../wild_type/protein_induced.onemodel'
import '../wild_type/wild_type.onemodel'

model AntitheticController (WildType)

  x1  = ProteinConstitutive()  # Sigma protein (Control action).
  x2  = ProteinInduced()       # Anti-sigma protein.
  x12 = ProteinConstitutive()  # Sigma::anti-sigma complex.
  A   = ProteinInduced()       # Protein of interest (Output).
  B   = ProteinInduced()       # Perturbation.
  species ref = 0              # Antithetic controller reference.

  parameter  # Add parameters of the antithetic controller.
    gamma = 10, x2.h = 2.5, A.h = 1, __perturbation = 0
    x1.omega = 15, x2.omega_max = 3, x12.omega = 0, A.omega_max = 20
  end

  reaction x1.m + x2.m -> x12.m ; gamma*x1.m*x2.m

  rule  # Set the equations of the antithetic controller.
    A.TF  := x1.m
    x2.TF := A.m
    A.omega_equation:  A.omega := A.omega_max * A.TF/(A.TF+A.h)
    B.TF := __perturbation
    ref := x1.J / x2.J * x2.h
  end

  rule  # Add x1, x2, x12, A and B to the model.
    x1.nu_t := nu_t
    x1.mu := mu
    ...  # For simplicity, we have ommited showing all the rules.
    B.J_host_sum := J_host_sum
  end

  rule  # Override equations (total cell burden and mass).
    WSum_equation:  WSum := p_r.W + p_nr.W + x1.W + x2.W + x12.W + A.W + B.W
    m_p_equation:   m_p  := p_r.m + p_nr.m + x1.m + x2.m + x12.m + A.m + B.m
  end
end
```

Code 5.18: *OneModel* implementation of the host-aware antithetic controller model. This figure only shows a reduced version of the code, the full implementation can be found at `https://github.com/sb2cl/thesis_fernandonobel` with the filename `antithetic_controller.onemodel`.

5.4 Results

5.4.1 The antithetic controller improves robustness

This section shows the effect of a perturbation (expressing an exogenous protein) on constitutive protein expression (open-loop) and the regulated expression of a protein by an antithetic controller (closed-loop). Recall that the perturbation does not interact directly with any open-loop or closed-loop species. The interaction is indirect: the perturbation introduces burden on the host cell, affecting the growth rate and the competition for shared host resources (ribosomes), affecting the expression of all the genes in the cell.

Figure 5.2 shows two simulations: the constitutive expression of protein A (open-loop) and the regulated expression of protein A by an antithetic controller (closed-loop). The equations of the open-loop are only (5.4) and (5.5) considering that $x_1/(x_1 + h_A)$ is saturated to one, the antithetic species are not expressed. The equations of the closed-loop are (5.4)–(5.8). Protein B is expressed from $t = 5\,\mathrm{h}$ to $t = 20\,\mathrm{h}$ generating a perturbation to the expression of protein A and the antithetic controller (in both simulations).

Protein B expression decreased protein A expression in open-loop and closed-loop cases. In the open-loop case (blue line), Figure 5.2A shows that protein A expression is around $11.7\,\mathrm{fg}\cdot\mathrm{cell}^{-1}$ when there is no perturbation, and $7.4\,\mathrm{fg}\cdot\mathrm{cell}^{-1}$ during perturbation: this is a drop in expression of 37 %. Whereas in the closed-loop case (red line), the expression of protein A is about $4\,\mathrm{fg}\cdot\mathrm{cell}^{-1}$, $3.2\,\mathrm{fg}\cdot\mathrm{cell}^{-1}$ during perturbation: a 20 % drop. The antithetic controller improves robustness to perturbations compared to constitutively expressed proteins.

The reference of the antithetic controller must be lower than the maximum value of protein expression reached by the open-loop case. Recall that we considered that there is saturation in the transcription of protein A: the open-loop case is when we are expressing protein A to the maximum possible value (the promoter of protein A is fully activated). Therefore, if we want the antithetic controller to be robust to perturbations, we must choose a reference below the maximum achievable expression value of protein A. If we choose a reference higher than the maximum achievable value, the antithetic controller will not follow the reference and will not improve the robustness since it will behave as the open-loop case.

The antithetic controller itself introduces burden on the host cell. Figure 5.2D shows the growth rate of the host-cell. During perturbation, the growth rate decreases due to the increase in cell burden produced by the expression of protein B. However, it should also be noted that the growth rate of the closed-loop case

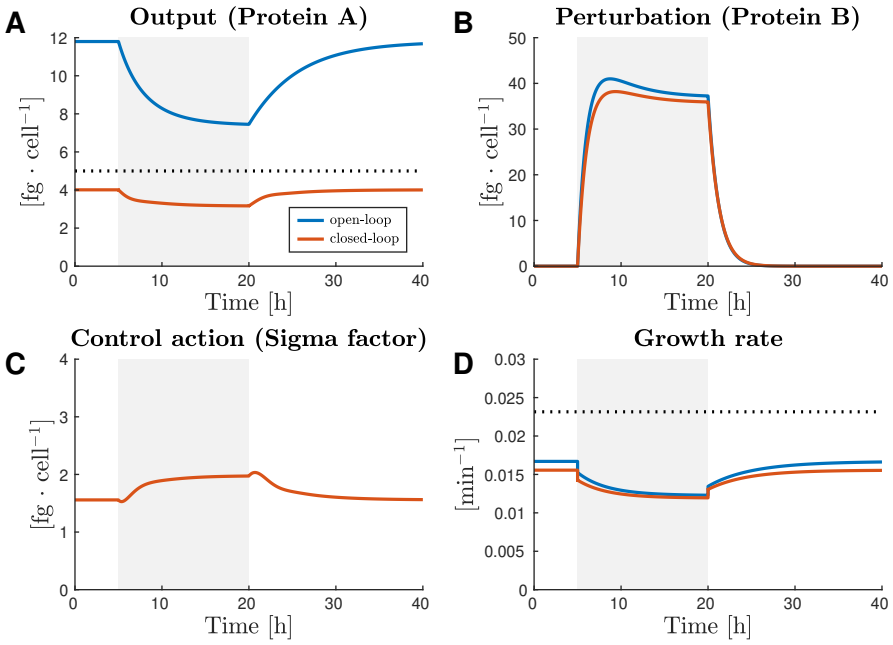

Figure 5.2: Host-aware simulations of (blue line) the constitutive expression of protein A and (red line) the antithetic controller regulating the expression of protein A. The shaded-gray area indicates the expression of protein B, which causes a perturbation. **A:** The solid lines are the output (protein A), and the dotted-black line is the desired reference. **B:** Amount of protein B in the host. **C:** The control action (sigma factor) of the antithetic controller. **D:** Solid lines are the growth rate of the host, and the dotted-black line is the wild-type growth without expressing any of the exogenous genes.

is lower than that of the open-loop case. The lower growth rate in the closed-loop case is due to the burden produced by expressing the antithetic controller species. Moreover, note that the growth rate is below the wild-type (the cell not expressing any exogenous protein) growth in both cases. The gain in robustness of the antithetic controller is not cost-free; the robustness is at the cost of burdening the cell.

The perturbation also affects the antithetic controller. Figure 5.2C shows the control action (the amount of sigma factor) of the antithetic controller. The control action increases when the disturbance is present to prevent the output from dropping. However, the increase is not sufficient to achieve perfect robustness to perturbations. Furthermore, Figure 5.2 shows that the burden generated by

the antithetic controller also affects the expression of protein B, as its expression is reduced compared to the open-loop case. Both perturbation and antithetic controller affects the expression of the other.

The position error of the antithetic controller is due to the dilution of the sigma and anti-sigma factors. In the ideal case, when we neglect the dilution of the sigma and anti-sigma factors, the antithetic controller behaves as an integral controller and can follow the reference without error. However, if we consider the dilution of the sigma and anti-sigma factors, the antithetic controller behaves like a leaky integrator. Figure 5.2A shows that the position error is 20 % and 36 % during the perturbation.

5.4.2 Increased gain leads to burden

This section analyzes the effect that increasing the gain in the antithetic controller has on the position error. The antithetic controller in real conditions (when considering the dilution of the sigma and anti-sigma factors) behaves as a proportional controller. Therefore, increasing the gain of the controller (by increasing the absolute value of the expression of the sigma and anti-sigma factors) should reduce the position error.

Figure 5.3 shows three simulations in which the gain of the antithetic controller is increased. The blue line (base gain) is the same closed-loop simulation as in Figure 5.2 and is shown for comparison purposes. The yellow line (high gain) is the same antithetic controller but with five times increased gain ($\omega_{x_1} = 75\,\mathrm{min}^{-1}$ and $\omega_{x_2} = 15\,\mathrm{min}^{-1}$), and the red line (very high gain) is fifty times increased gain ($\omega_{x_1} = 750\,\mathrm{min}^{-1}$ and $\omega_{x_2} = 150\,\mathrm{min}^{-1}$).

Increasing the gain reduces the position error of the antithetic controller. Figure 5.3A shows that the high gain antithetic controller reduces the position error to 15.6 % (and 23.2 % during perturbation). This reduction is an improvement over the base gain antithetic controller, but the host cell grows more slowly due to the increased burden. It is also noticeable how protein B expression is repressed more than in the base controller (recall that protein B is expressed in all cases with the same strength, its variations are due to cell burden).

Increasing the gain of the antithetic controller too much can lead to worse performance. Figure 5.3A shows that the very high gain antithetic controller, instead of further reducing the position error, increases the position error up to 42.8 %. This increase in the position error is caused because increasing the gain overloads the host cell and reduces the overall translation rate, indirectly repressing protein A

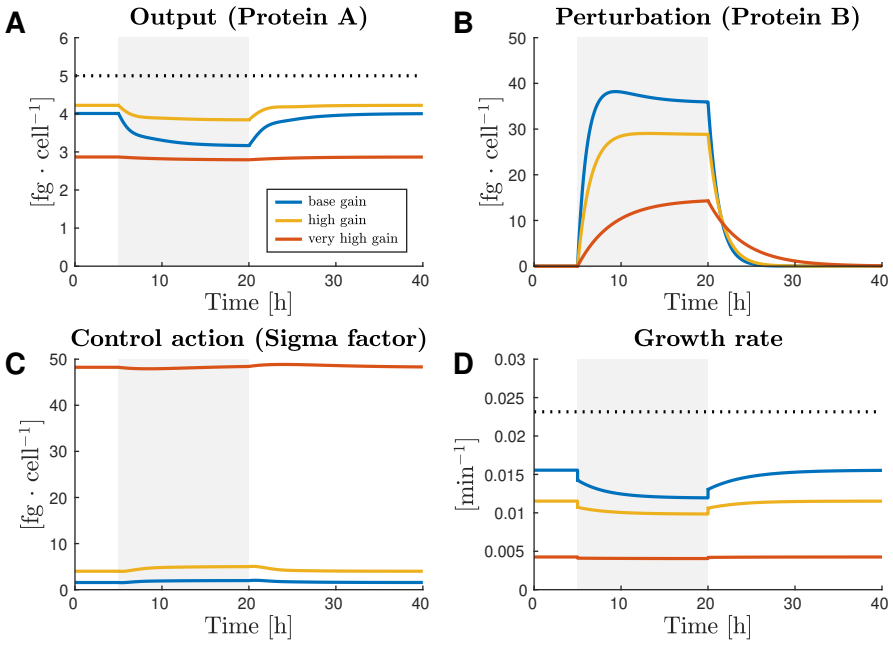

Figure 5.3: Host-aware simulations of the antithetic controller regulating the expression of protein A. The shaded-gray area indicates the expression of protein B, which causes a perturbation. Blue, yellow, and red lines correspond to the antithetic controller's base, high, and very-high gains. **A:** The solid lines are the output (protein A), and the dotted-black line is the desired reference. **B:** Amount of protein B in the host. **C:** The control action (sigma factor) of the antithetic controller. **D:** Solid lines are the growth rate of the host, and dotted-black line is the wild-type growth.

expression and thus increasing the position error. Note also the drastic decrease in growth rate.

In summary, we can improve the position error in the antithetic controller by increasing its gain, but the increase in cell burden must be considered because it can worsen the position error if the cell reaches an overburdened state.

5.4.3 Reducing dilution reduces gain

The previous section showed that increasing the gain to reduce position error is limited by host-cell burden interactions. This section investigates another possible way to reduce the antithetic controller's position error. The idea is as follows: if we reduce the dilution of the sigma and anti-sigma factors, the antithetic controller will again behave as an integral controller with zero position error.

The dilution of sigma and anti-sigma factors is mainly due to cell growth. In this work, we consider protein-based sigma and anti-sigma factors that are stable proteins (not subject to active degradation), and the only dilution is due to cell division through cell growth. Cell growth can be reduced by two means: (i) by reducing the substrate available to the cell and (ii) by increasing the cell burden. Both ways reduce the growth rate, but their effects are not identical (recall Chapter 4). In this section, we choose to reduce the available substrate to reduce the dilution of sigma and anti-sigma factors (recall we already used and analysed before the idea of increasing the cell burden by means of the perturbation generated by the expression of an extra protein).

Figure 5.4 shows two simulations: the blue line is the closed-loop antithetic controller from Figure 5.2 (shown here for comparison), and the red line is the same simulation but with the available substrate reduced to $0.72\,\mathrm{g}\cdot\mathrm{L}^{-1}$ (the base substrate is $3.6\,\mathrm{g}\cdot\mathrm{L}^{-1}$).

The position error increases as the dilution is reduced. We expected to see that the position error is reduced with the reduction in dilution rate, but Figure 5.4A shows the opposite: reducing dilution, by reducing the available substrate, makes the position error worse (this is also true for a reduction in growth rate due to cell loading). The explanation for this phenomenon (that the position error is worse when we reduce dilution) is that the controller's gain depends on the translation step (sigma and anti-sigma factors have to be translated using ribosomes), and the translation rate depends directly on the growth rate (see Chapter 2).

The integral action is not recovered by reducing the growth rate; the growth rate must be reduced while keeping the controller gain constant (or even increasing it).

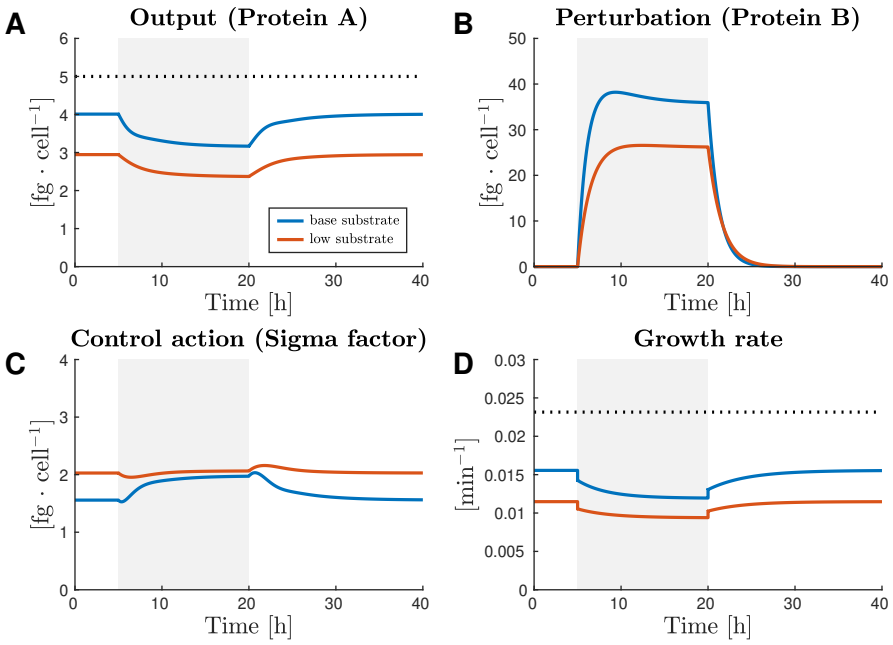

Figure 5.4: Host-aware simulations of the antithetic controller regulating the expression of protein A. The shaded-gray area indicates the expression of protein B, which causes a perturbation. Blue and red lines correspond to base and low substrate for the host cell. **A:** The solid lines are the output (protein A), and the dotted-black line is the desired reference. **B:** Amount of protein B in the host. **C:** The control action (sigma factor) of the antithetic controller. **D:** Solid lines are the growth rate of the host, and dotted-black line is the wild-type growth.

5.4.4 RBS effect on the reference

The previous sections considered that sigma and anti-sigma factors were expressed using the same RBS value. However, in Chapters 2 and 4, we showed that the effective value of different RBS changes differently with changes in substrate, growth rate, and free ribosome availability. This section investigates the effects of expressing sigma and anti-sigma factors using different RBS values.

The reference of the antithetic controller is robust to parameter uncertainties and perturbations (as we have seen in the previous sections). The reference value is calculated as the ratio of the expression of the sigma and anti-sigma factors, and therefore, the reference is robust to parameter uncertainties or perturbations that

equally affect the sigma and anti-sigma factors expression. Using the same RBS allows us to cancel the effect of parameter uncertainties and perturbations.

If we use different RBS for the sigma and anti-sigma factors, the effects of perturbations will not cancel. Figure 5.5 shows three simulations: the blue line is the closed-loop antithetic controller in Figure 5.2 that has equal RBS for sigma and anti-sigma, the yellow line uses a low RBS for sigma and a high RBS for anti-sigma, and the red line uses a high RBS for sigma and a low RBS for anti-sigma. The sigma transcription was adjusted to match the same reference as the base case in the yellow and red cases, the Table 5.3 shows the values for each case. Moreover, in all three cases, the dilution of the sigma and anti-sigma factors was neglected to showcase the effect on the reference more clearly. In the case of the real conditions (taking into account the dilution of the sigma and anti-sigma factors), the effect on the reference would be the same but could be potentially hidden due to the position error.

Table 5.3: Parameters for each RBS combination case.

Name	**Equal rbs**	**Low-high rbs**	**High-low rbs**
ω_{x_1} (min^{-1})	15	15	2.8846
ω_{x_2} (min^{-1})	3	0.5769	3
$k_b^{x_1}$ $(\text{cell} \cdot \text{min}^{-1} \cdot \text{molec}^{-1})$	4.7627	4.7627	12.44
$k_b^{x_2}$ $(\text{cell} \cdot \text{min}^{-1} \cdot \text{molec}^{-1})$	4.7627	12.44	4.7627
$k_u^{x_1}$ (min^{-1})	119.7956	119.7956	10.04
$k_u^{x_2}$ (min^{-1})	119.7956	10.04	119.7956

Figure 5.5A shows that when there is no perturbation, the three simulations have the same reference (they were adjusted for it). However, during perturbation, the reference of the three simulations acts differently in response to the change in the growth rate. The blue line is robust to the change in growth rate (as we have seen in the previous sections). The yellow line reduces the reference value, and the red line increases it. This phenomenon is similar to what we showed in Chapter 2 with ribosomal and non-ribosomal genes.

Using different RBS values to express sigma and anti-sigma factors makes the reference dependent on the growth rate, the substrate, and the availability of free ribosomes. Moreover, we can make the reference increase or decrease with changes in growth rate by choosing different combinations of RBS.

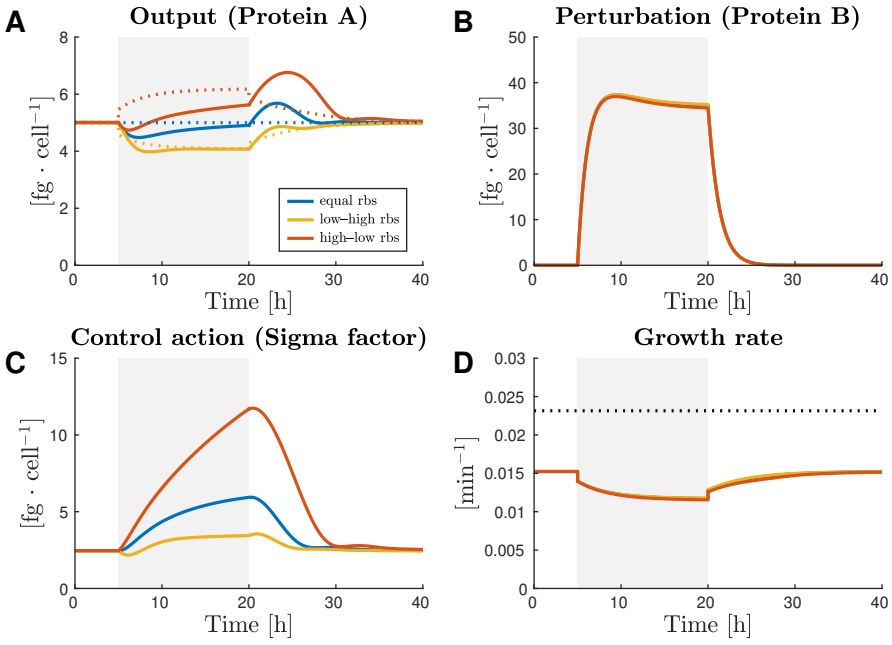

Figure 5.5: Host-aware simulations of the antithetic controller regulating the expression of protein A. The shaded-gray area indicates the expression of protein B, which causes a perturbation. Colored lines correspond to different RBS combinations for expressing sigma and anti-sigma factors: (blue) equal RBS; (yellow) low-RBS for sigma and high-RBS for anti-sigma; and (red) high-RBS for sigma and low-RBS for anti-sigma. The dilution of sigma and anti-sigma factors is neglected. **A:** The solid lines are the output (protein A), and the dotted lines are the reference for each case. **B:** Amount of protein B in the host. **C:** The control action (sigma factor) of the antithetic controller. **D:** Solid lines are the growth rate of the host, and dotted-black line is the wild-type growth.

5.5 Discussion and conclusions

We used our host-aware model (see Chapter 2) to analyze the antithetic controller's performance under burden perturbations and host-circuit interactions. This type of analysis would not have been possible without using a host-aware model, and *OneModel* streamlined using the host-aware model.

As expected, the antithetic controller improved (under realistic conditions) the robustness of protein expression compared to constitutive protein expression. We showed that increasing the gain increases the robustness and reduces the position error, but there is a limit to increasing the gain. If we exceed that limit, the cell will be overloaded expressing the antithetic controller, and the performance of the antithetic controller (robustness and position error) will worsen due to host-circuit interactions.

Counterintuitively, we have shown that reducing the dilution of the sigma and anti-sigma factors does not recover the integral action of the antithetic controller. The explanation for this phenomenon (that the error position is worse when we reduce the dilution) is that the antithetic controller's gain depends on the translation step (sigma and anti-sigma factors have to be translated using ribosomes), and the translation rate depends directly on the growth rate (the dilution). So neglecting dilution of sigma and anti-sigmas, in the case of low growth rate, is not a good approximation: we have to consider the ratio between the antithetic controller's gain and dilution. Therefore reducing the dilution (on its own) does not recover the integral action: we have to reduce the dilution and keep, or even increase, the gain.

We have shown that the choice of RBS values for expressing the antithetic controller has relevant effects in the reference. If we choose equal RBS values for the expression of sigma and anti-sigma factors, the reference will be robust. However, if we choose different RBS values, the reference will depend on the growth rate, the substrate, and the availability of free ribosomes. It is easy to use equal RBS values for expressing the antithetic controller in biological implementations, and we can still tune the reference value by changing the transcription rates. Nevertheless, we could take advantage of this phenomenon to code references depending on the growth rate. For example, we could tune an antithetic controller to drop its reference if the cell grows slowly due to a burden perturbation, freeing cellular resources and mitigating the perturbation effect. This phenomenon has similar effects to what we wanted to achieve with the reference conditioning scheme presented in [101]. Recall that even with different RBS values, the reference of the antithetic controller will remain robust to the uncertainties of the process parameters to be controlled.

We did not show it, but there is another way to increase the antithetic controller's gain. Reducing the half-activation threshold of expressing protein A (i.e., increasing the sensitivity to sigma factors) increases the gain without generating extra burden to the host cell: the antithetic controller will need to apply lower control actions to achieve the same process activation.

The antithetic controller's implementation we analyzed is protein-based: it depends on the translation step to express the sigma and anti-sigma factors. There are other implementations based on short mRNA interference [44], which skip the translation step and could produce less burden to the host cell. Therefore, increasing the gain of an mRNA-based antithetic controller should not have a limit as the protein-based has (if high transcription rates do affect the growth rate, similar phenomenons as the protein-based may arise). Interestingly, reducing the dilution (the growth rate) will still reduce the closed-loop gain of mRNA-based controllers because the gain of the process will still be reduced.

In this Chapter, we have done a brief analysis of the antithetic controller to showcase the host-aware model's usefulness. This analysis can be easily extended to many other synthetic genetic circuits. The host-aware model is simple enough to obtain analytical solutions that further explain the effects of burden, host-circuit, and circuit-circuit interactions. The prediction of the effect using different RBS values in the antithetic controller in the reference should be easy to test experimentally.

This Chapter is an example of using our host-aware model and *OneModel* to analyze burden and host-circuit interactions in complex circuits. Improving host-aware models and their accessibility will allow us to successfully design and implement complex synthetic genetic circuits.

Chapter 6

Control of turbidostats

> "Even though we may still be a long way from our goal, we are approaching it step by step. Everything is justifying our hopes. We must never, therefore, let ourselves fall into the way of thinking 'ignorabimus' ('We shall never know'), but must have every confidence that the day will dawn when even those processes of life which are still a puzzle today will cease to be inaccessible to us natural scientists."
>
> —Eduard Buchner, *Cell-free fermentation*

The contents of this chapter are based on the following journal publication:

- De Battista, H. et al. "Output Feedback Linearization of Turbidostats After Time Scaling". In: *IEEE Transactions on Control Systems Technology* (2018), pp. 1–9. ISSN: 10636536. DOI: 10.1109/TCST.2018.2834882

6.1 Abstract

Turbidostats are a class of bioreactors gaining interest due to the recent availability of micro- and small-scale devices for characterization and scaling-up of biotechnological systems relevant in the biotech and pharma industries. The goal is to keep cell density constant in continuous operation. Thus the control law, i.e. the substrate feeding strategy, must guarantee global or semi-global convergence to an equilibrium point. However, their control is difficult due to the uncertain,

time-varying and nonlinear nature of the processes involved. In this Chapter we propose an adaptive control law that globally stabilizes the desired biomass set-point. Further, in a certain region of the state space the controller linearizes the dynamic behavior after some time scaling. This way, the orbits of the closed loop system are imposed by the designer. The intrinsic integral action of the gain adaptation rejects parameter uncertainties. Moreover, the controller implementation only assumes biomass concentration to be measured. Both simulated and experimental results show the performance of the controller.

6.2 Introduction

Industrial biotechnology uses enhanced and/or genetically modified microorganisms to produce specialty metabolites (e.g. amino acids, vitamins, food additives, biofuels, ...) of importance for the health, chemical, food and energy sectors among others. Bioreactors are the workhorses where characterization, scaling-up, and production take place. Therefore, their feedback control has received much attention. However, control of bioreactors is difficult due to: (i) uncertainty on key variables of the system representing the physiological state of the culture, (ii) non-linear process dynamics, and (iii) large variability. In this context, model-based design of controllers has been addressed using simple models based on mass-balances [4] and developing generic and robust controllers based on the minimal model concept [23, 104, 72]. On the one hand, mass-balance based models concentrate the uncertainty in specific terms; the bioreaction kinetics, and the bioreaction yields. On the other, robust controllers based on the minimal model concept use the mass-balance structure and rough information on the kinetics structure and bounds.

In continuous bioreactions the volume of culture inside the bioreactor is kept constant by setting the inlet flow rate equal to the outlet one. The higher risk of contamination and cell mutation in continuous bioreactions have favored its use only for processes involving microorganisms with high mutation-stability. Yet continuous bioreactors have some advantages, like increased productivity for biomass and growth associated products, reduction of raw materials, waste production and maintenance requirements [91], or the possibility of analyzing cultures under sets of steady state conditions [105, 94]. Thus, industrial bioprocesses like biofuels, pharmaceutics and bioplastics production, etc. are increasingly being migrated to continuous bioreactors [37]. A few classes of continuous bioreactors are mostly used. In chemostats the goal is to keep a desired specific growth rate of the microorganism [37]. Nutristats are used to drive substrate concentration to a desired set-point, most often corresponding to optimal biomass or production rate [54,

72]. Several approaches have been used to stabilize the system at this optimum operating point, assuming measurement of full state [115, 48, 57], biomass [104], or reaction rate and substrate measurement or estimation [71, 69, 103].

Turbidostats regulate cell density at a prescribed value. Low cost micro- and small-scale turbidostats are increasingly becoming available for characterization and scaling-up of systems without nutrient limitation [58, 116, 122, 12, 75, 43]. Cell density is continuously monitored either using a spectrophotometer / turbidometer to measure optical density [90, 31], or using methods based on dielectric permiytivity [27, 55]. The control law, i.e. the substrate feeding strategy, must guarantee global or semi-global convergence of biomass to an equilibrium point.

Linear control theory was used in [28], and exact feedback linearization in [81, 86, 3, 115]. Feeding strategies that are proportional to the reaction rate avoid biomass washout and also avoid falling in batch operation. Variations adapting the controller gain [69, 83] or including error feedback [22, 23] have been proposed to achieve robustness against process uncertainties and variability, and to improve the transient response. In the first case, globally stable adaptive control laws that in the end are integral control ones are designed [1, 69], being the feedback gain dynamically adapted in such a way that control never saturates. Controllers using this idea can be applied to regulate biomass, substrate or product concentration in continuous bioprocesses. These controllers eliminate steady state errors but they were not designed to set the transient response by controller tuning. To speed up the transient, a nonlinear proportional control with adaptive gain leading to a class of nonlinear PI (proportional-integral) controller was proposed in [23] to control growth rate. Though exhibiting fast convergence properties and robustness, it is not clear how to impose the desired dynamics and care should be taken to avoid controller saturation. A saturated PI control is proposed in [104] with implicit anti-windup protection, achieving robust closed-loop stability. Biomass is measured and substrate is estimated using an observer. The gains of both observer and controller must meet some bounding conditions, and an iterative tuning procedure is proposed to set them.

In this Chapter we propose an adaptive control law that globally stabilizes the desired biomass concentration. Time scaling and Stability Preserving Maps are used as tools to simplify stability analysis and controller design [84]. The controller linearizes the dynamic behavior in a certain region of the state space after time scaling. This way, the closed loop orbits are imposed by the designer, resulting in very simple tuning rules. Neither a detailed model of the growth kinetics nor knowledge of the bioreaction yields are required. The controller assumes biomass concentration is measured and the reaction rate is estimated from it using high-gain or sliding observers [84, 120] whereby a practical principle of separation can

be assumed. The intrinsic integral action of the gain adaptation rejects parameter uncertainties. In case the reaction rate is indirectly measured or calculated with some error then the steady state output error is bounded.

The Chapter is organized as follows. In 6.3 the problem is formulated. The proposed control law and its analysis are considered in 6.4 and 6.5 respectively. Simulations highlighting the performance of the controller are shown in 6.6 and its experimental validation in 6.7. Conclusions are outlined in 6.8.

6.3 Problem statement

Consider a pure culture growing under a single substrate being continuously fed to the reactor [4, 28]:

$$\begin{aligned} \dot{x} &= \mu(x, s, \mathbf{q})x - D(t)x\,, \\ \dot{s} &= -y\mu(x, s, \mathbf{q})x + D(t)(s_i - s)\,, \\ \dot{\mathbf{q}} &= Q\left(\mathbf{q}, x, s, D(t)\right)\,, \end{aligned} \tag{6.1}$$

where $x \in \Re_+$ is the biomass concentration, $s \in \Re_+$ the substrate concentration, $s_i \in \Re_+$ the substrate concentration in the inlet flow, y the substrate-biomass conversion yield, $D(t)$ the dilution rate, i.e. the ratio between the flow rate and the culture volume that will be used as control input, and $\mu(x, s, \mathbf{q})$ the specific growth rate, where $\mathbf{q}$ gathers uncertain parameters and other variables (DO, temperature, pH, growth-linked products, etc.) affecting the reaction kinetics.

Model (6.1) encompasses a large number of practical cases for production of biomass and growth-linked products.

Assumption 6.3.1. *Assume the specific growth rate vanishes if there is no substrate* ($\mu(x, 0, \mathbf{q}) \equiv 0$)*, is strictly positive whenever substrate is available* ($\mu(x, s, \mathbf{q}) > 0 \quad \forall s > 0$)*, and is bounded* ($\mu(x, s, \mathbf{q}) \leq \mu_m \quad \forall x, s, q \geq 0$)*. For technical reasons, assume it can be modeled by a continuously differentiable function of* x *and* s ($\mu(x, s, \mathbf{q}) \in \mathcal{C}^1$)*.*

This assumption is fulfilled by all standard specific growth rate models [4].

Assumption 6.3.2. *Assume biomass concentration* x *is available for measurement.*

Definition 6.3.1. *A bioprocess operates in continuous mode when* $D(t) \geq d > 0\ \forall t \geq t_0$ *for some sufficiently small* d*.*

The goal is to design a control law $D(t)$ that globally stabilizes system (6.1) at the specified biomass set-point x^*.

6.4 Proposed control law

6.4.1 Growth rate proportional feeding law

Consider a controller of the form

$$D(t) = \gamma \mu(t) x \,. \tag{6.2}$$

In the following we use $\mu(t)$ to emphasize the time varying nature of the specific growth rate and that the feedback law is not constructed from the uncertain model $\mu(x, s, \mathbf{q})$ but from an observer-based estimation (we used a super-twisting based observer (6.26)).

Using (6.2), the closed loop dynamics for biomass and substrate can be expressed as:

$$\begin{aligned} \dot{x} &= D(t)(\gamma^{-1} - x)\,, \\ \dot{s} &= D(t)(s_i - y\gamma^{-1} - s)\,. \end{aligned} \tag{6.3}$$

Consider the simplest case, with $\gamma = \gamma^* \triangleq 1/x^*$. Then:

$$\begin{aligned} \dot{x} &= D(t)(x^* - x)\,, \\ \dot{s} &= D(t)(s^* - s)\,, \end{aligned} \tag{6.4}$$

where $s^* = s_i - yx^*$ is the substrate equilibrium point. It is clear that for a continuous bioreaction as stated in Definition 6.3.1, the controller (6.2) with $\gamma = \gamma^*$ locally stabilizes the equilibrium $P^* = (x^*, s^*)$ of (6.4).

6.4.2 Time scaling

The idea now is to semiglobally stabilize the biomass concentration preserving the control law structure (6.2), i.e. to feed the reactor in proportion to growth rate, but shaping the feedback gain γ so as to improve the transient dynamics. To this end, it is convenient to express the closed loop dynamics in a new time scale τ where the controller design problem becomes simpler. Consider the time scaling given by:

$$d\tau = D(t)dt\,, \tag{6.5}$$

where τ can be interpreted as the volume of substrate fed in $[t_0, t]$ relative to the bioreactor one. Now, defining the time derivative with respect to the new time variable as $(\cdot)' = \frac{d(\cdot)}{d\tau}$, the closed loop dynamics (6.3) become:

$$\begin{aligned} x' &= \gamma^{-1} - x\,, \\ s' &= s_i - y\gamma^{-1} - s\,. \end{aligned} \tag{6.6}$$

Notice the system in the transformed time scale has its eigenvalues at -1, and the parameter γ can be shaped to improve convergence of one variable, say x, relative to the other. Notice also that the dynamics of biomass and substrate are decoupled in the transformed time scale.

6.4.3 Adaptive shaping of γ

The goal now is to shape γ so as to improve convergence of x towards the desired set-point x^*. To this end, we propose the following adaptive control law in the τ-scale:

$$\gamma' = -\gamma^2 \left[(\gamma^{-1} - x)(1-a) - b(x - x^*)\right] f(\gamma)\,, \tag{6.7}$$

where $f(\gamma)$ is a saturation function satisfying:

$$\begin{aligned} &f \in \mathcal{C}^1\,, \\ &f(\gamma) > 0 \quad \forall \gamma \in (\underline{\gamma}, \bar{\gamma}), \quad \frac{y}{s_i} < \underline{\gamma} < \gamma^* < \bar{\gamma}\,, \\ &f(\gamma) = 1 \quad \forall \gamma \in \Gamma = [\gamma_m, \gamma_M] \subset (\underline{\gamma}, \bar{\gamma}), \quad \gamma^* \in \Gamma\,, \\ &f(\gamma) = 0 \quad \forall \gamma \notin (\underline{\gamma}, \bar{\gamma})\,, \end{aligned} \tag{6.8}$$

and $\gamma_0 = \gamma(\tau_0) \in (\underline{\gamma}, \bar{\gamma})$.

To prove global stability of the closed loop system corresponding to the biomass dynamics define the errors:

$$\begin{aligned} \tilde{x} &= x - x^* \,, \\ \tilde{\gamma} &= \gamma^{-1} - \gamma^{*-1} \,. \end{aligned} \tag{6.9}$$

Taking derivatives with respect to τ we obtain the error dynamics

$$\begin{aligned} \tilde{x}' &= \tilde{\gamma} - \tilde{x} \,, \\ \tilde{\gamma}' &= f(\gamma) \left[(\tilde{\gamma} - \tilde{x})(1-a) - b\tilde{x} \right] \,. \end{aligned} \tag{6.10}$$

Let us now define the following positive definite radially unbounded function:

$$\begin{aligned} W_1 &= \int_0^{\tilde{\gamma}} \frac{g}{f(\frac{\gamma^*}{1+g\gamma^*})} dg \,, \\ W_2 &= \frac{b-a+1}{2} \tilde{x}^2 \,, \\ W &= W_1 + W_2 \,, \end{aligned} \tag{6.11}$$

with $b-a+1 > 0$. Using the Leibnitz Integral Rule for W_1, the τ-time derivative of (6.11) along the state trajectory is:

$$\begin{aligned} W_1' &= \frac{\tilde{\gamma}\tilde{\gamma}'}{f(\gamma)} = -(a-1)\tilde{\gamma}^2 - (b-a+1)\tilde{\gamma}\tilde{x} \,, \\ W_2' &= -(b-a+1)\tilde{x}^2 + (b-a+1)\tilde{\gamma}\tilde{x} \,, \\ W' &= -(a-1)\tilde{\gamma}^2 - (b-a+1)\tilde{x}^2 \,, \end{aligned} \tag{6.12}$$

which is negative definite for $a > 1$. Therefore, the equilibrium $(\tilde{x}, \tilde{\gamma}) = (0, 0)$ of the error dynamics (6.10) is globally asymptotically stable.

Next we show that, after some finite time Tc the control law (6.2)-(6.7) linearizes the biomass dynamics in the $\tau-$scale within the set $\Gamma = [\gamma_m, \gamma_M]$. To this end, differentiating x twice with respect to τ in (6.6) one gets

$$\begin{aligned} x'' &= -\gamma^{-2}\gamma' - x' \,, \\ &= f(\gamma)\left[x'(1-a) - b(x-x^*)\right] - x' \,. \end{aligned} \tag{6.13}$$

Since $\tilde{\gamma}$ asymptotically converges to zero, there will exist some finite time T_c such that $\gamma \in \Gamma$ for all $\tau \geq T_c$ where, from (6.8), $f(\gamma) = 1$. Therefore, once γ enters the set Γ the system will behave according to the linearised dynamics:

$$x'' + ax' + b(x - x^*) = 0 \,. \tag{6.14}$$

Remark 6.4.1. *Notice that although in this linear region in the $\tau-$scale it suffices to choose $a, b > 0$ to ensure asymptotic stability, the Lyapunov function* (6.11) *we found is more restrictive.*

6.4.4 Stability in the original time scale

In the original time scale t, the control law becomes:

$$\begin{aligned} D(t) &= \gamma\mu(t)x \,, \\ \dot{\gamma} &= -\gamma^2\left[(\mu(t) - D(t))x(1-a) - D(t)b(x-x^*)\right] f(\gamma) \,, \\ \gamma_0 &\in (\underline{\gamma}, \bar{\gamma}) \,, \end{aligned} \tag{6.15}$$

with $b > (a-1) > 0$ and $f(\gamma)$ defined as in (6.8).

Stability of the system (6.3)-(6.15) is equivalent to that of (6.6)-(6.7) if the time scaling (6.5) defines a stability preserving map.

Theorem 6.4.1. *If $D(t)$ is bounded and strictly positive, the time scaling* (6.5) *defines a stability preserving map.*

Proof. If $D(t)$ is bounded and strictly positive, the time-scaling (6.5) defines a strictly increasing and onto function $\xi : t \longrightarrow \tau$. Thus, the homomorphism given by the identity transformation for the coordinates and the time-scaling defined by (6.5) preserves stability. See Th. 7 and Coroll. 8 in [84]. □ □

Now we must prove that the dilution rate $D(t)$ in (6.15) is bounded and strictly positive. That is, we must prove that the control law (6.15) induces continuous mode operation.

First we prove that $D(t)$ does not vanish. The saturation function (6.8) in the adaptive control law (6.15) ensures that

$$\gamma(t) \in (\underline{\gamma}, \bar{\gamma}) > 0 \quad \forall t \geq t_0 \,. \tag{6.16}$$

Therefore, it will suffice to prove that both biomass $x(t)$ and the specific growth rate $\mu(t)$ are bounded away from zero. That is, we must prove that the control law avoids both washout of biomass and batch operation with substrate depletion.

Notice $(x = 0, s, q)$ are stable fixed points of system (6.1) which correspond to biomass washout. We can assume initial conditions satisfying $x_0 > 0$, i.e. there is some initial biomass concentration in the bioreactor. To avoid washout the region $R_{\overline{\mathcal{X}}} = \{(x, s, q) | x \geq \underline{x}\}$ must be positively invariant for some sufficiently small $0 < \underline{x} < x^*$. By continuity $R_{\overline{\mathcal{X}}}$ will be locally non-attractive if the dilution $D(t)$ is smaller than the specific growth rate as the biomass approaches zero:

$$\lim_{x \to 0^+} D(t) = \lim_{x \to 0^+} \gamma(t)\mu(x, s, q)x < \mu(0, s, q) \,. \tag{6.17}$$

Taking into account (6.16), condition (6.17) is trivially fulfilled. Thus, the control law (6.15) avoids washout.

Now, given the properties for the specific growth rate in Assumption 6.3.1 and the result above, to prove that the specific growth rate $\mu(x, s, q)$ does not vanish it suffices to prove that the substrate concentration s so does not, i.e. the control law avoids batch operation with substrate depletion. From the second equation in (6.1), it will suffice to show that

$$\lim_{s \to 0^+} D(t) = \lim_{s \to 0^+} \gamma(t)\mu(x, s, q)x > yx \lim_{s \to 0^+} \frac{\mu(x, s, q)}{s_i - s} \,, \tag{6.18}$$

so that $\mathcal{S} = \{(x, s, q) | s = 0\}$ is locally non-attractive. Notice the control law (6.15) fulfills condition (6.18) provided $\underline{\gamma} > y/s_i$, as required in the definition (6.8) of the saturation function $f(\gamma)$. Therefore, the specific growth rate $\mu(x, s, q)$ is bounded away from zero.

Finally, we prove upper boundedness of the dilution rate $D(t)$. Assumption 6.3.1 and (6.16) ensure both the specific growth rate $\mu(t)$ and the adaptation gain γ are upper bounded. So, it only remains to show that biomass x is also upper bounded. From (6.3), positivity of $D(t)$, and boundedness of $\gamma(t)$ given in (6.16),

the biomass concentration x cannot grow unboundedly for its derivative will be negative whenever $x > \underline{\gamma}^{-1}$.

Thus, we have proved that the dilution rate $D(t)$ is both bounded from above and bounded away from zero. Therefore, the time scaling (6.5) is well defined.

Notice D vanishes on $\mathcal{S}$. Therefore, the controller is not able to start the process from $s = 0$. Anyway, the non-attractiveness condition (6.18) is always satisfied since $x^* < s_i/y$. Thus, it (semi)globally stabilizes the equilibrium.

6.5 Controller tuning

The key tuning parameters of the controller are the gains a and b. Some guidelines are given below.

6.5.1 Closed loop poles in the $\tau-$time scale

Recall that after some finite time, the biomass error in the transformed time scale τ will follow the linear dynamics (6.14). By choosing a, b in the control law (6.15), and using $b - a + 1 > 0$, we can place the corresponding eigenvalues λ_1, λ_2 of the closed-loop system with:

$$\begin{aligned} \omega_0 &= \sqrt{\lambda_1 \lambda_2} = \sqrt{b}\,, \\ \xi &= \frac{\lambda_1 + \lambda_2}{2\omega_0} = \frac{a}{2\sqrt{b}}\,, \\ \frac{1}{2}\omega_0^{-1} < \xi &< \frac{1}{2}(\omega_0 + \omega_0^{-1})\,, \end{aligned} \tag{6.19}$$

where ω_0 is the natural frequency, ξ the damping coefficient. Equivalently, the last inequalities can be rewritten in terms of ξ and the damping factor $\sigma = \xi\omega_0$ as

$$\sigma > \frac{1}{2}\,, \quad \xi < \frac{\sigma}{\sqrt{2\sigma - 1}}\,. \tag{6.20}$$

6.5.2 Effects of the growth rate observer

Any biased measurement in x will produce an error at steady state independently of the controller tuning. To analyze the controller performance we consider that estimation $\hat{\mu}$ of the specific growth rate μ introduces an error, so that $\Delta\mu = (\mu - \hat{\mu})/\mu$. Then (6.1) and (6.15) become:

$$\begin{aligned} \dot{x} &= \gamma\mu(t)x\left(\gamma^{-1} - \frac{\hat{\mu}(t)}{\mu(t)}x\right), \\ \dot{\gamma} &= -\gamma^2\hat{\mu}(t)x\left[(1-\gamma x)(1-a) - \gamma b(x - x^*)\right] f(\gamma), \end{aligned} \tag{6.21}$$

that evaluated at steady state result in:

$$x_{ss} = \frac{x^*}{1 - \frac{(a-1)}{b}\Delta\mu}. \tag{6.22}$$

The effect of error in estimation of the specific growth rate will depend on the term:

$$\Delta_{x/\mu} := \frac{(a-1)}{b}. \tag{6.23}$$

Recall that $b > a - 1 > 0$ and $a > 1$, and notice $\Delta_{x/\mu} \to 0$ as $a \to 1$. Therefore, there is a compromise between transient response (speed and overshoot) and steady state error. If a, b are designed to have a given damping coefficient ξ and damping factor $\sigma = a/2$ (thus a given settling time), then

$$\Delta_{x/\mu} = \frac{(2\sigma - 1)\xi}{\sigma^2}. \tag{6.24}$$

For a given ξ, the error has a single local maximum $\Delta_{x/\mu,\max} = \xi$ at $\sigma = 1$ ($a = 2$) and goes to zero as $a \to 1$ or $a \to \infty$. Also, the smaller ξ, the smaller the error will be. The steady state error of the classical algorithm (6.2) is very similar to the error in the estimation of μ: $\Delta_{x/\mu} \cong 1$.

Thus, if well designed, the proposed controller will significantly reduce the error. Anyway, to avoid this source of error one can estimate μ using a supertwisting observer that gives finite time convergence with zero steady state estimation error [84].

6.5.3 Design of $f(\gamma)$

The span of γ is constrained for the restriction $y/s_i > \gamma$ and by the maximum flow rate of the pumping system. The inequality $\underline{\gamma} > y/s_i$ implies that the desired set-point for biomass concentration must be bounded by $x^\star < s_i/y$, just expressing that the amount of substrate converted into biomass at equilibrium ($yx^\star$) must be smaller than the one in the input flow (s_i). The adaptation limits $\underline{\gamma}$ and $\bar{\gamma}$ must be set to fulfill these constraints. The limits of the linear region γ_m and γ_M can be close to $\underline{\gamma}$ and $\bar{\gamma}$ but considering that a smooth transition from 1 to 0 is required.

6.6 Simulated results

Intensive simulation analysis has been carried out to verify the main features of the proposed control law. Set-point changes, process start-up and effect of the initial conditions have been evaluated considering growth kinetics without and with substrate inhibition. Furthermore, the control strategy has been compared with previous proposals in the literature, and the advantages and disadvantages are discussed. The process dynamics (6.1) has been simulated using the specific growth kinetics plotted in Figure 6.1a and process parameters $y = 2.22$ and $s_i = 3.6$. The dilution rate is given by the dynamic feedback law (6.15), with bounding function $f(\gamma)$ as depicted in Figure 6.1b. The tuning constants a, b were set to different values. In this section, the growth rate is assumed to be known so as to avoid including observer dynamics in the controller assessment.

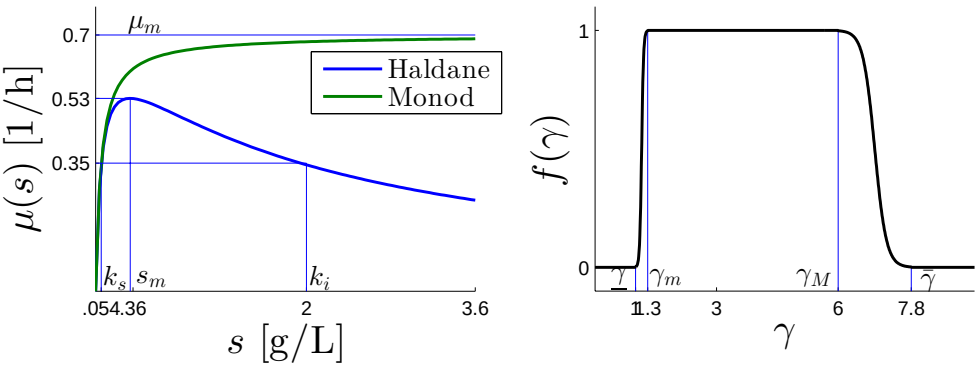

Figure 6.1: (a) Specific growth kinetics $\mu(s)$. (b) Bounding function $f(\gamma)$ of controller (6.15).

The controller performance has been evaluated in time scales t and τ since both lead to interesting conclusions. On the one hand, the $t-$scale measures the process time in hours having an obvious meaning. On the other, the time unit in the $\tau-$scale is the retention time, so that τ represents the times the volume of the bioreactor is entirely exchanged. That is, the duration of the transient response in the $\tau-$scale represents the amount of medium consumed before reaching the new set-point. A key advantage of the $\tau-$scale is that the transient is independent of the dilution rate, which in steady state equals the specific growth rate. On the other hand, processes with lower specific growth rates are slower in the $t-$scale, thus controllers performance is more difficult to compare. It is important to point out also that the time mapping $t(\tau)$ is monotonous and preserves the amplitude of a time response. Moreover, as a general rule, faster responses in the $\tau-$scale are typically faster in the $t-$scale also.

6.6.1 Set-point change

The response of the process to a set-point change for different controller settings is first presented. The initial operating point is $(x_0, s_0, \gamma_0) = (1/5, s_i - yx_0, x_0^{-1})$ and the final one is $(x^*, s^*, \gamma^*) = (1/3, s_i - yx^*, x^{*-1})$. The controller parameters a and b were chosen so that the damping coefficient is constant ($\xi = .5$) while ω_0 ranges from 2 to 12.

Figures 6.2 and 6.3 show the results for Monod and Haldane specific growth kinetics. Note that the overshoot in the $x-$response is the same for all ω_0 but $\omega_0 = 12$. The reason is that the controller temporarily leaves its linear region in this case (see the plot of γ), so the expected linear behavior (6.14) is temporarily lost. In case the controller always operates in its linear region, there is a direct mapping between (a, b) and the response in the $\tau-$scale. Because of the nature of the time scaling, the underdamped response designed in the τ domain is also observed in the $t-$scale. Moreover, as observed in the figures, the order is also preserved. The faster responses in the $\tau-$scale are also the faster ones in the $t-$scale. The reason is that, although the $t(\tau)$ mappings are not the same for all cases, they are very similar. Moreover, at the peak or settling time of a given response, the $t(\tau)$ mapping practically coincides with the ones corresponding to faster responses.

From Figures 6.2 and 6.3 note that in the $t-$scale the process with inhibition is slower than without it even if the responses in the $\tau-$scale are equal, being consistent with the fact that the dilution rate is lower. Note also that the shape of the $t(\tau)$ mappings are very similar for both growth kinetics.

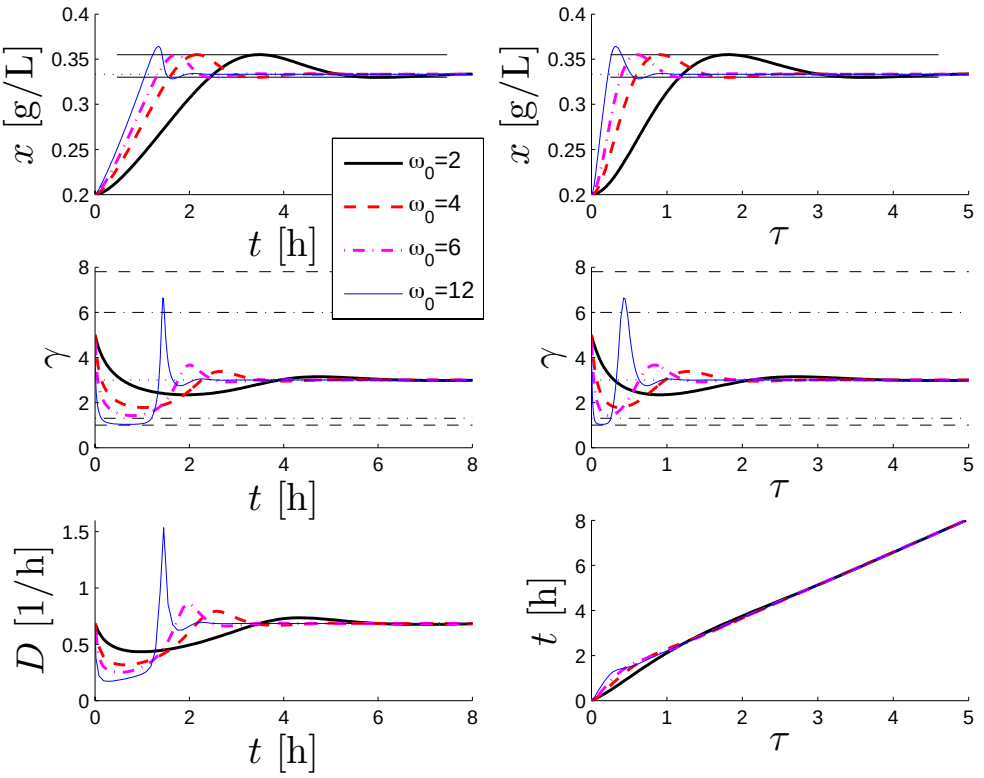

Figure 6.2: Set-point change for Monod kinetics. Former set-point: $(x_0, s_0, \gamma_0) = (.2, 3.16, 5)$, new set-point: $(x^*, s^*, \gamma^*) = (1/3, 2.86, 3)$.

6.6.2 Start-up

The process has been also simulated for the same set-point as before but starting from very low initial conditions $(x_0, s_0, \gamma_0) = (.05, .01, x^*)$. Note that the process is not initially at steady state. Figures 6.4 and 6.5 show the results for the Monod and Haldane growth kinetics and for different controller tunings. In particular, the response for three different damping coefficients ξ and two different damping factors $\sigma = \xi\omega_0$ are depicted. The responses in the $\tau-$scale present the classical linear second-order dynamics while γ keeps between (γ_m, γ_M). The responses in the $t-$scale are very similar, preserving amplitude and order. Comparing both figures, notice the x responses in the $\tau-$scale are identical, whereas the responses in the $t-$scale are very similar, although the process with Monod kinetics is a bit

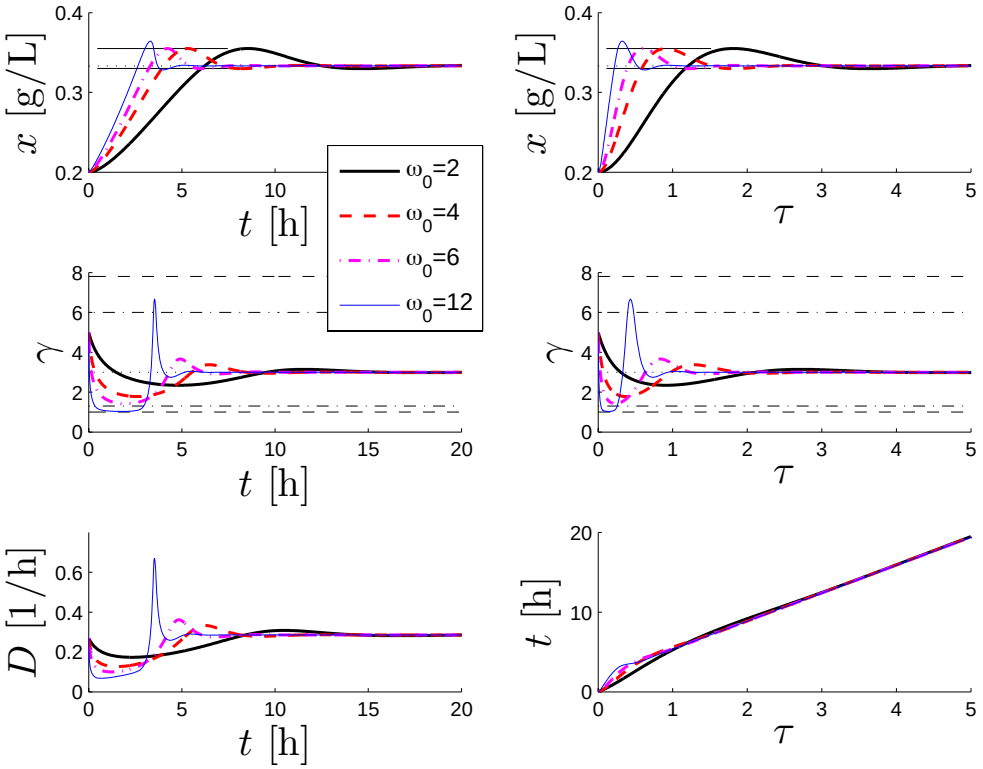

Figure 6.3: Set-point change for Haldane kinetics. Former set-point: $(x_0, s_0, \gamma_0) = (.2, 3.16, 5)$, new set-point: $(x^*, s^*, \gamma^*) = (1/3, 2.86, 3)$.

faster because the specific growth rate is higher. This shows the robustness of the controller with respect to the growth rate kinetics.

In addition, the controller performance is robust with respect to variability in the substrate initial conditions, as observed in Figure 6.6. This figure shows the start-up from different initial substrate concentrations for Monod kinetics and setting $\sigma = 1.5, \xi = .75$, evidencing that subsystem $x - \gamma$ is independent of the substrate initial condition in the $\tau-$scale. Therefore, only the dependence of the time-scaling on the substrate s when getting back into the original time scale t *–via* the specific growth rate $\mu(x, s)$– will induce differences in the $t-$scale. Anyway, the time responses of biomass preserve the shapes and amplitudes of the responses in the $\tau-$scale.

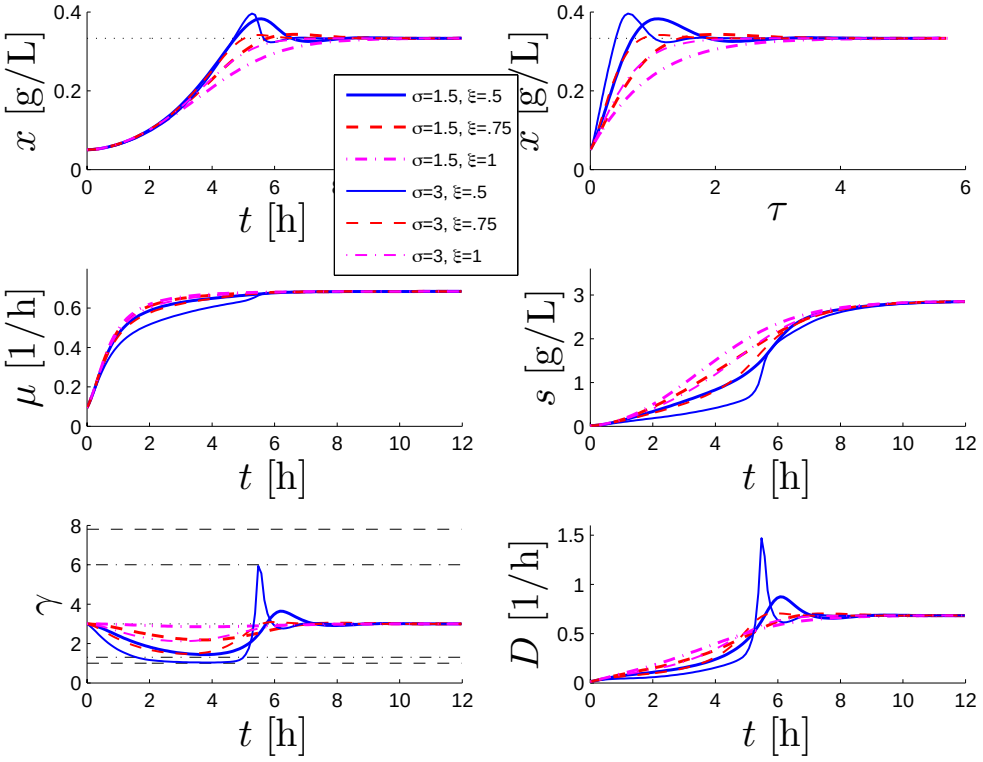

Figure 6.4: Process start-up for Monod kinetics. Initial condition: $(x_0, s_0, \gamma_0) = (.05, .01, 3)$, set-point: $(x^*, s^*, \gamma^*) = (1/3, 2.86, 3)$.

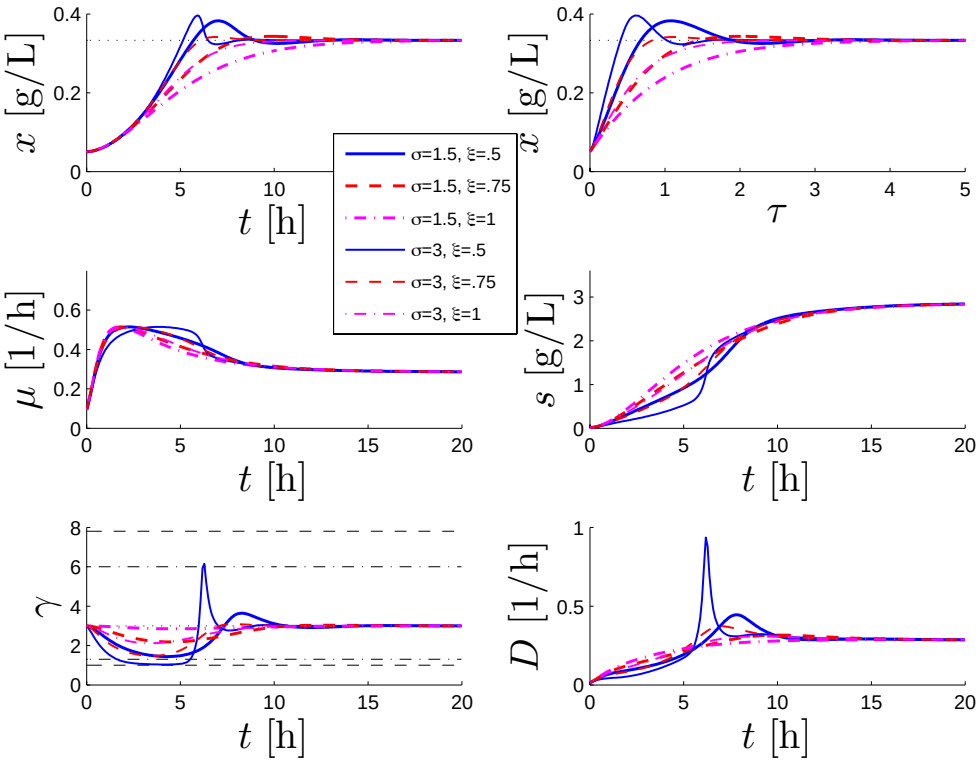

Figure 6.5: Process start-up for Haldane kinetics. Initial condition: $(x_0, s_0, \gamma_0) = (.05, .01, 3)$, set-point: $(x^*, s^*, \gamma^*) = (1/3, 2.86, 3)$.

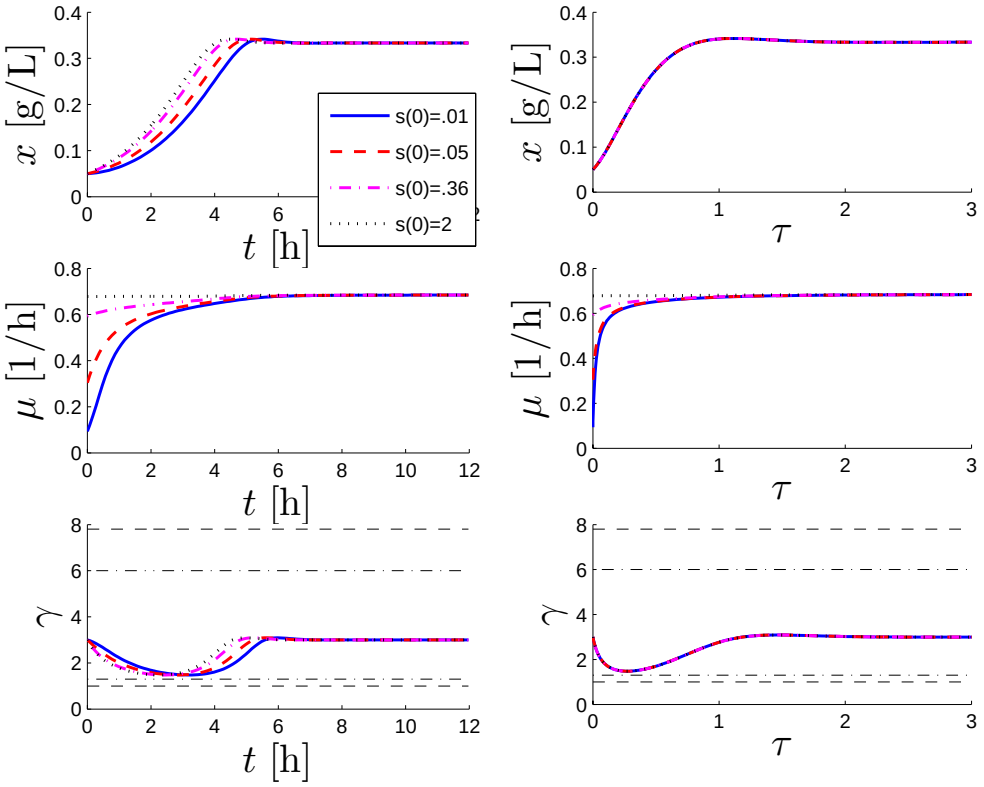

Figure 6.6: Process start-up from different initial substrate concentrations for Monod kinetics. Initial condition: $(x_0, s_0, \gamma_0) = (.05, s_0, 3)$, set-point: $(x^*, s^*, \gamma^*) = (1/3, 2.86, 3)$. Controller tuning: $(\xi, \sigma) = (.75, 3)$.

6.6.3 Comparison with other controllers

Next the proposed controller is compared with the baseline controller $D(t) = \gamma^* \mu(t) x$ and with the one proposed in [69]:

$$\begin{aligned} \dot{\gamma} &= K\mu(t)x(x - x^*)(\gamma - \underline{\gamma})(\bar{\gamma} - \gamma)\,, \quad \gamma_0 \in (\underline{\gamma}, \bar{\gamma})\,, \\ D(\mu, x) &= \gamma\mu(t)x\,. \end{aligned} \tag{6.25}$$

Notice controller (6.25) consists basically of the baseline controller where the gain γ is adapted, introducing an integral compensation. The main feature of this controller is its capability to reject disturbances at steady state. However, since it has only one tuning parameter and it does not include proportional action, its potential to improve the transient response is limited. In fact, the cost of faster responses is larger overshoots. This is probably the main shortcoming of this approach. On the other hand, the controller proposed in this Chapter allows improving the desired transient response and is very easy to tune. Yet, although significantly attenuated, some measurement errors cannot be completely rejected.

Figure 6.7 shows the time t evolution of biomass concentration and adaptation gain for different tunings of the proposed controller (left) and controller (6.25) (right) in presence of a -10% error in the estimation of $\mu(t)$. The response obtained with the baseline controller is plotted in both columns. The simulation scenario of start-up is repeated. Notice with the proposed controller one can achieve better settling time and can easily set the desired transient specifications. This is because there are two tuning parameters to set the linear second order dynamics (in $\tau-$scale). Furthermore, it significantly reduces the error in comparison with the baseline controller as predicted by (6.24) form 11% to 2.5%. On the other hand, controller (6.25) achieves zero error thanks to the integral action. Yet, in pure integral adaptive controllers as (6.25), a large integral action that drives the error to zero reasonably fast deteriorates significantly the transient response. Although controller (6.25) response in the large tends to the desired value, the transient is longer (12 hours) than the one obtained with our controller (5 hours). These results suggest exploring the possibility of combining or scheduling both controllers in order to improve the transient response and completely reject steady state errors.

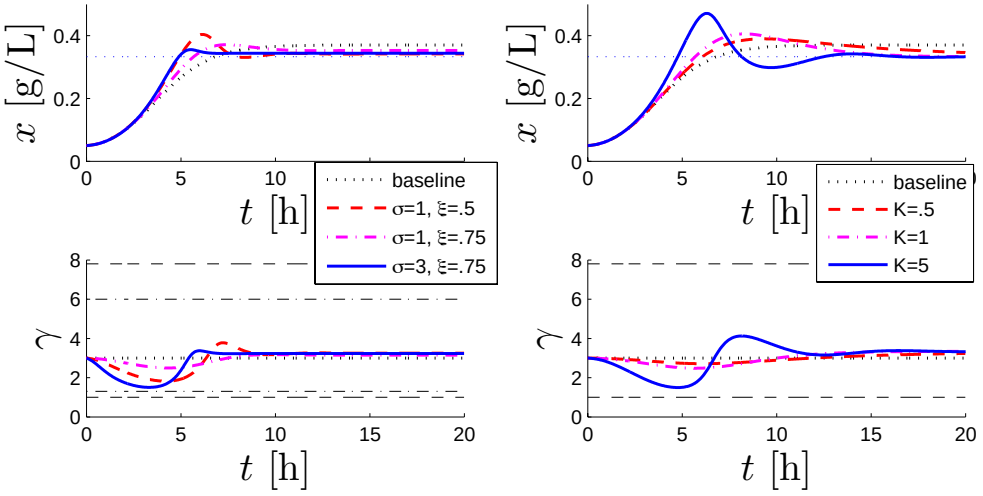

Figure 6.7: Effect of μ estimation error on controller performance using the proposed (left) and (6.25) (right) controllers. Initial condition: $(x_0, s_0, \gamma_0) = (.05, .01, 3)$, set-point: $(x^*, s^*, \gamma^*) = (1/3, 2.86, 3)$.

6.7 Experimental results

We used the experimental setup shown in Figure 6.8. It consists of a 16ml turbidostat adapted from [116] and fed by a syringe pump (NE-1000, New Era Pump Systems, Inc.). Volume was kept constant by injecting pressurized air using a small pump. Optical density at 650 nm (OD_{650}) was measured with an absorbance custom-made sensor using a photodiode converting light intensity to frequency (TSL235-LF, Farnell). In our working range there was a linear relationship between OD_{650} and biomass concentration, with $OD_{650} = 1$ approximately corresponding to $1.5\,\mathrm{g}\cdot\mathrm{L}^{-1}$ wet weight, so we controlled optical density. We grew *E. coli* transformed cells containing the blue-purple chromoprotein amilCP using SOB medium as substrate with 20mM glucose ($3.603\,\mathrm{g}\cdot\mathrm{L}^{-1}$). We expected a yield for *E. coli* grown on glucose around $y = 2.2\,\mathrm{g_{glucose}}\cdot\mathrm{g}^{-1}_{\mathrm{biomass}}$ [76] so that $y/s_i \approx 0.6$. Therefore we used the lower bound $\underline{\gamma} = 1$ with the saturation function (6.8) as in Figure 6.1b. This gives a good stability margin. The culture was grown overnight, and was diluted before each experiment so that the initial concentration was $\mathrm{OD}_{650} = 0.12$ in all cases. The sensor was calibrated at each experiment using a spectrophotometer (Zuzi 4140) to compensate for absorbance of the medium.

Figure 6.8: Experimental turbidostat setup.

To estimate the specific growth rate $\hat{\mu}$ from biomass measurements x we used the super-twisting based observer (6.26) presented in [25]:

$$\begin{aligned} \sigma &= p^{-1} \log\left(\frac{x}{z_1}\right), \\ \dot{z}_1 &= -D(t) z_1 + p z_1 z_2 + 2p\beta \left|\sigma\right|^{\frac{1}{2}} \operatorname{sign}(\sigma), \\ \dot{z}_2 &= \alpha \operatorname{sign}(\sigma), \\ \hat{\mu} &= p z_2 . \end{aligned} \tag{6.26}$$

The observer parameters were set to $\alpha = 1.625$, $\beta = 1.5$ and $p = 0.2$ using the tuning methodology in [84], and its initial conditions were set to $z_1(0) = 0.1, z_2(0) = 1.25$. These result in $\hat{\mu}(0) = 0.25$ for the estimated specific growth rate. We did not allow for an initial open-loop period for the observer to converge before closing the loop. This observer converges in finite time. In all cases it converged in less than one hour, as shown in Figure 6.10 (middle).

Figure 6.9 shows the experimental results fit well with the theoretical predictions considering the noisy, uncertain and time-varying context. We used the param-

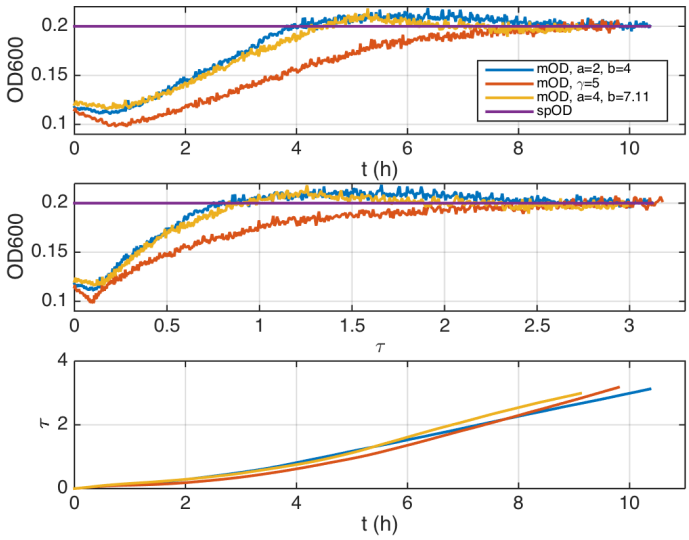

Figure 6.9: Biomass concentration x in the t time scale (top), in the transformed τ-time (middle), and relationship between t and τ(bottom). Initial conditions: $x_0 \approx 0.12(OD_{600})$. Set-point $x^* = 0.2$. Proposed controller: $(a, b) = (4, 7.11)$ (yellow), $(a, b) = (2, 4)$ (blue). Baseline controller (orange).

eters $(a, b) = (4, 7.11)$ corresponding, in the τ scale, to peak time $\tau_p = 1.8$, settling time $\tau_s = 2$ (98% criterium) and overshoot $\delta = 0.03$, and $(a, b) = (2, 4)$ for $\tau_p \approx 1.8$, $\tau_s = 4$ and $\delta = 0.16$. The proposed control law is compared with the baseline controller corresponding to (6.2) with $\gamma = 1/x^\star$. The comparison between the time responses in the $\tau-$ and $t-$ scales shows that order and magnitude are preserved. The experimental relationship between both time scales is shown in the bottom panel. As predicted by the simulations in section 6.6, although the $t(\tau)$ mappings are not the same for all cases they are very similar.

Figure 6.10 shows the experimental dilution D, estimated specific growth rate $\hat{\mu}$ and controller gain γ. Notice the controller is robust with respect to uncertain factors that affect the specific growth rate and may differ from one experiment to another (e.g. substrate initial concentration, oxygen diffusion, cells metabolic state, etc.). This is reflected in the slightly different steady state value of the specific growth rate reached for each experiment. Notice there were performed independently in different days, so the environmental conditions (e.g.temperature) were most probably different. The controller designed in τ to achieve longer

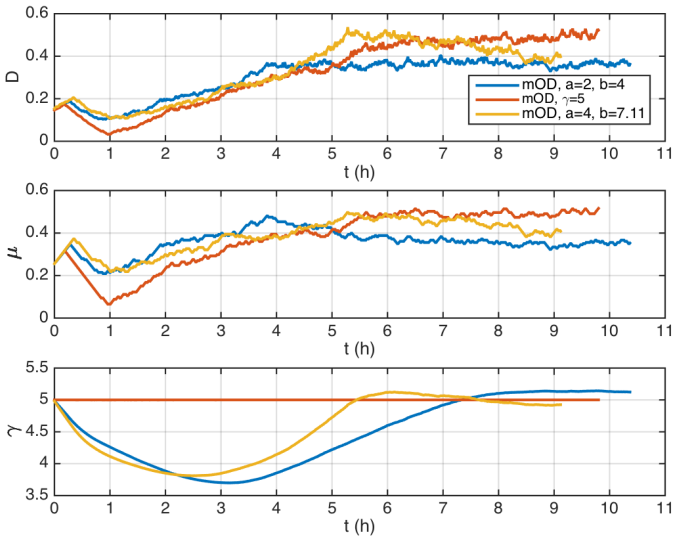

Figure 6.10: Dillution D (top), estimated specific growth rate μ (middle) and controller gain γ (bottom). Proposed controller: $(a, b) = (4, 7.11)$ (yellow), $(a, b) = (2, 4)$ (blue). Baseline controller (orange).

settling time ($(a, b) = (2, 4), \tau_s = 4$) indeed delivers lower values of dilution rate D and has lower values of the adaptation gain γ during the transient.

6.8 Discussion and conclusions

In this Chapter we have proposed an adaptive control law that globally stabilizes the desired biomass set-point in continuous bioreactions. Using time scaling we render the system linear in the transformed time scale, where analysis and tuning of the controller becomes extremely simple. Stability is preserved in the original time domain. Important time-response characteristics such as order and magnitude relationships are also preserved in practice. Furthermore, our controller only assumes biomass concentration is measured, and does not require a detailed model of the growth kinetics or knowledge of the bioreaction yields. The intrinsic integral action of the gain adaptation rejects parameter uncertainties. In case the reaction rate is indirectly measured or calculated and some error appears, then the error is bounded, and the proposed controller can be tuned so as to significantly reduce it.

The simulation and experimental results validate the easiness to tune the controller to achieve desired time response patterns, and its robustness in face of noisy uncertain bioreaction environments.

Chapter 7

General conclusions

> "Suddenly it was clear to me that all the beautiful complexity of life had simplicity at its core. This is the kind of thing mathematicians love."
>
> —Eric S. Lander, *Unraveling the Threads of Life*

The contributions of this Thesis were listed in Chapter 1, and particular conclusions can be found at the end of each main chapter. Here some general conclusions are drawn and discussed together besides some proposed lines for future work.

- **(Chapter 2)** We have presented a small-size model of gene expression dynamics accounting for host-circuit interactions. The predictions of our model agree with the experimental data of *E. coli*. In particular, the model accurately predicts—for different substrate concentrations—the growth rate and the mass fraction distribution between ribosomal and non-ribosomal mass.
- **(Chapter 2)** The good agreement between the predictions of our model and experimental data highlight the relevance of the cellular resources recruitment strength defined in our model as a key functional coefficient. Our resources recruitment strength coefficient allows us to explain the distribution of resources between the host and the genes of interest. This functional coefficient explicitly considers the interplay between the flux of available free resources and lab-accessible gene expression characteristics. In particular, the promoter and RBS strengths.
- **(Chapter 2)** Among other predictions, the model provides insights into how the differential role of promoter and RBS strengths in protein expression

may have evolved in *E. coli* and other micro-organisms to encode the mass distribution between ribosomal and non-ribosomal proteins as a function of cell growth rate.

- **(Chapter 2)** Our model may also be useful for design purposes in synthetic biology where it can be used to design the proper promoter-RBS strategy depending on the desired behavior of the genes expression as a function of growth rate.

- **(Chapter 3)** We developed *OneModel*, a new SBML-compliant tool for defining models focused on user accessibility, simplicity and modularity.

- **(Chapter 3)** Instead of developing monolithic files that contain all the equations and model parameters values, *OneModel* syntax allows the user to add new models to connect with the old ones. That is, splitting models into modules and re-programming them by making small changes that fulfill the new requirements, but always having the option to go back. *OneModel* reduces the modeling efforts by increasing modularity. The user can develop and test each module of a model separately, avoiding to start from scratch every time that the user implements a new model.

- **(Chapter 3)** We used *OneModel* for almost all of the simulations of this Thesis. The guided examples in this chapter showed the benefits of modular incremental implementations, and how it would be easy for a non-expert user to take advantage of previously defined models (such as our host-aware model).

- **(Chapter 4)** In this work, we demonstrate the need for cell burden models as well as their utility. Our results show that, at low expression levels, gene transcription mainly defined Titer, Rate (productivity) and Yield (TRY) at bioreactor level, and gene translation had a limited effect; whereas, at high expression levels, TRY depended on the product of both, in agreement with experiments in the literature [121].

- **(Chapter 4)** Multi-scale models, like the one presented here, incorporating the dynamics of (i) the cell population in the bioreactor, (ii) the substrate uptake and (iii) the interaction between the cell host and the expression of enzymes of interest, are useful to understand the differential roles of gene transcription and translation in shaping TRY trade-offs for a wide range of expression levels and the sensitivity of the TRY space to variations in substrate availability.

- **(Chapter 4)** Optimal gene expression is central for the development of both bacterial expression systems for heterologous protein production, and microbial cell factories for industrial metabolite production. With our approach it will be easier to fulfill industry-level overproduction demands optimally, as measured by the key performance metrics: titer, productivity rate and yield.

- **(Chapter 5)** As expected, the antithetic controller improved (under realistic conditions) the robustness of protein expression compared to constitutive protein expression. We showed that increasing the gain increases the robustness and reduces the position error, but there is a limit to increasing the gain. If we exceed that limit, the cell will be overloaded expressing the antithetic controller proteins, and the performance of the antithetic controller (robustness and position error) will worsen due to host-circuit interactions.

- **(Chapter 5)** Counterintuitively, we have shown that reducing the dilution of the sigma and anti-sigma factors does not recover the integral action of the antithetic controller. Reducing the dilution (i.e., the growth rate) in turn reduces the translation rate which lowers the antithetic controller's gain. Therefore reducing the dilution (on its own) does not recover the integral action: we have to reduce the dilution and keep, or even increase, the gain.

- **(Chapter 5)** The choice of RBS values for expressing the antithetic controller proteins has relevant effects in the reference. If we choose equal RBS values, the reference will be robust. However, if we choose different RBS values, the reference will depend on the growth rate, the substrate, and the availability of free ribosomes. We could take advantage of this phenomenon: for example, we could tune an antithetic controller to drop its reference if the cell grows slowly due to a burden perturbation, freeing cellular resources and mitigating the perturbation effect.

- **(Chapter 6)** We have proposed an adaptive control law that globally stabilizes the desired biomass set-point in continuous bioreactions. Using time scaling we render the system linear in the transformed time scale, where analysis and tuning of the controller becomes extremely simple. Stability is preserved in the original time domain. Important time-response characteristics such as order and magnitude relationships are also preserved in practice.

- **(Chapter 6)** Our controller only assumes biomass concentration is measured, and does not require a detailed model of the growth kinetics or knowledge of the bioreaction yields. The intrinsic integral action of the gain adap-

tation rejects parameter uncertainties. In case the reaction rate is indirectly measured or calculated and some error appears, then the error is bounded, and the proposed controller can be tuned so as to significantly reduce it.

- **(Chapter 6)** The simulation and experimental results validate the easiness to tune the controller to achieve desired time response patterns, and its robustness in face of noisy uncertain bioreaction environments.

The goal of this Thesis was to develop methods that are at the basis of the design, analysis, and implementation of synthetic genetic systems considering the scales of bioproduction (bioreactor, host, and circuit). The main conclusion is that the three scales of bioproduction are deeply interconnected, and we must take these interactions into account to design and analyze synthetic genetic systems successfully. This work provides guidelines for improving the practical implementation of synthetic circuits. In addition, many of the predictions of the host-aware model can be easily tested in the laboratory, the most important being the differential role of the RBS and promoter on the expression of exogenous proteins under different growth rates This thesis also provides insight into burden effects and the host-circuit and circuit-circuit interactions that arise.

This Thesis has helped pave the way for the design and analysis of multi-scale host-aware synthetic genetic systems.

Future work

There are many possibilities to use, expand, and improve the methods presented in this Thesis. This section shows chapter by chapter the main possibilities for future work.

- **(Chapter 2)** Though we only considered *E. coli* in our host-aware model, our findings can be extrapolated to other microorganisms, and the model can be easily fitted using a small amount of experimental data of the host cell. Further extensions of the model can be easily implemented. The model only requires as input a measure of the fraction of available substrate with respect to the saturated case, and predicts both the resulting cell specific cell growth rate and the mass and mass rates of the expressed proteins. This makes its integration with constraint-based models of metabolism rather straightforward. The possibility to consider expression systems using orthogonal ribosomes can also be implemented without much difficulties. All this makes the model useful in the context of model-based design of gene synthetic circuits and protein expression systems.

- **(Chapter 3)** *OneModel* requires *Python* version 3.8 and is installed as a package with PyPI. Although the installation process is simple, it can be challenging for non-expert users. In the future, an executable version and a web interface will avoid this step. Error feedback is often one of the weaknesses of domain specific languages, and is key to ensuring accessibility. Currently, *OneModel* provides simple error feedback, but our goal is to improve it. Object-oriented programming in *OneModel* is class-based, but we are experimenting with prototype-based programming which could simplify the internal implementation of *OneModel*. *Antimony*, *Little b*, *BioCRNpyler*, *OneModel*, and many other tools not listed here, paved the way to define more complex and larger models efficiently. However, these types of models are difficult to debug, test and maintain. Normally, researchers have performed these tasks manually, but this is inefficient for models of this size, or in sometimes even impossible. Therefore, there is an increasing need for tools to debug and test SBML models automatically.

- **(Chapter 4)** Our multi-scale model could take advantage of the information generated by constraint-based metabolic models. The integration of the macroscopic population dynamics at the bioreactor level, the prediction of metabolic fluxes from the external uptake of substrates to the precursors of metabolic pathways of interest, and the cost of producing the enzymes in the pathway of interest have clear interest for metabolic engineering. There already are works in the literature in this direction. Yet, current approaches are either very complex (like ME-models) or too simplistic. We could use our host-aware and population dynamics models in Chapters 2 and 4 along with well-established constraint-based metabolic models. The substrate definition in our multi-scale model is simple and does not capture many real effects (like multiple limiting substrates, oxygen concentrations, etc). Metabolic models handle these real effects. Thus, as a first step, a mapping between both representations of the substrates would be needed. Both classes of models make predictions about growth rate; thus, reconciling the two predictions could help integrate the models.

- **(Chapter 5)** The host-aware analysis can be easily extended to many other synthetic genetic circuits. The host-aware model is simple enough to obtain analytical solutions that further explain the effects of burden, host-circuit, and circuit-circuit interactions. The prediction of the effect using different RBS values in the antithetic controller in the reference should be easy to test experimentally.

- **(Chapter 6)** The proposed control law for bioreactors was validated both in simulation and in experiments. A family of slight variants of the control

law can be generated by changing the time-scaling. These new controllers could be faster or easier to tune. However, the bulk of future work lies in developing mini-bioreactors with such integrated controllers. In this Thesis, we designed and implemented the mini-bioreactors; the next step would be to assemble a final prototype, which would be easy to replicate and commercialize.

Bibliography

"Who can say if the thoughts you have in your mind as you read these words are the same thoughts I had in my mind as I typed them? We are different, you and I, and the qualia of our consciousnesses are as divergent as two stars at the ends of the universe.
And yet, whatever has been lost in translation in the long journey of my thoughts through the maze of civilization to your mind, I think you do understand me, and you think you do understand me. Our minds managed to touch, if but briefly and imperfectly.
Does that thought not make the universe seem just a bit kinder, a bit brighter, a bit warmer and more human?
We live for such miracles."

—Ken Liu, *The Paper Menagerie and Other Stories*

[1] Antonelli, R. and Astolfi, A. "Nonlinear controllers design for robust stabilization of continuous biological reactors". In: *Control Applications, 2000. Proceedings of the 2000 IEEE International Conference on*. IEEE. 2000, pp. 760–765 (cit. on p. 121).

[2] Aoki, S. K. et al. "A universal biomolecular integral feedback controller for robust perfect adaptation". In: *Nature* 570.7762 (2019), pp. 533–537. ISSN: 14764687. DOI: 10.1038/s41586-019-1321-1. URL: http://dx.doi.org/10.1038/s41586-019-1321-1 (cit. on pp. 55, 101).

[3] Bastin, G. and Van Impe, J. "Nonlinear and adaptive control in biotechnology: A tutorial." In: *European J. of Control* 1.1 (1995), pp. 37–53 (cit. on p. 121).

[4] Bastin, G. and Dochain, D. *On-line Estimation and Adaptive Control of Bioreactors*. Elsevier, 1990 (cit. on pp. 120, 122).

[5] Bernstein, J. A. et al. "Global analysis of mRNA decay and abundance in Escherichia coli at single-gene resolution using two-color fluorescent DNA microarrays". In: *Proceedings of the National Academy of Sciences* 99.15 (2002), pp. 9697–9702. DOI: 10.1073/pnas.112318199 (cit. on p. 200).

[6] Bienick, M. S. et al. "The interrelationship between promoter strength, gene expression, and growth rate". In: *PLoS ONE* 9.10 (2014) (cit. on pp. 11, 77).

[7] Blanchard, A. E., Liao, C., and Lu, T. "Circuit-Host Coupling Induces Multifaceted Behavioral Modulations of a Gene Switch". In: *Biophysical Journal* 114.3 (2018). ISSN: 15420086. DOI: 10.1016/j.bpj.2017.12.010 (cit. on p. 11).

[8] Boada, Y. et al. "Host-circuit interactions explain unexpected behavior of a gene circuit." In: *IFAC-PapersOnLine* 51.19 (2018), pp. 86–89. ISSN: 24058963. DOI: 10.1016/j.ifacol.2018.09.030 (cit. on p. 11).

[9] Boada, Y., Vignoni, A., and Pico, J. "Engineered Control of Genetic Variability Reveals Interplay among Quorum Sensing, Feedback Regulation, and Biochemical Noise". In: *ACS Synthetic Biology* 6.10 (2017), pp. 1903–1912. ISSN: None. DOI: 10.1021/acssynbio.7b00087 (cit. on p. 77).

[10] Boada, Y. et al. "Extended Metabolic Biosensor Design for Dynamic Pathway Regulation of Cell Factories". In: *iScience* 23.7 (2020). ISSN: 25890042. DOI: 10.1016/j.isci.2020.101305 (cit. on pp. 77, 103).

[11] Boada, Y. et al. "Modeling and optimization of a molecular biocontroller for the regulation of complex metabolic pathways". In: *Frontiers in Molecular Biosciences* 9 (2022). DOI: 10.3389/fmolb.2022.801032 (cit. on pp. 6, 99).

[12] Bolic, A. et al. "A flexible well-mixed milliliter-scale reactor with high oxygen transfer rate for microbial cultivations". In: *Chemical Engineering J.* 303 (2016), pp. 655–666 (cit. on p. 121).

[13] Boo, A., Ellis, T., and Stan, G.-B. "Host-aware synthetic biology". In: *Current Opinion in Systems Biology* 14 (2019), pp. 66–72. ISSN: 2452-3100. DOI: 10.1016/j.coisb.2019.03.001 (cit. on p. 77).

[14] Bosdriesz, E. et al. "How fast-growing bacteria robustly tune their ribosome concentration to approximate growth-rate maximization". In: *The FEBS Journal* 282.10 (2015), pp. 2029–2044. DOI: 10.1111/febs.13258 (cit. on pp. 11, 24, 77, 83, 179, 202).

[15] Bremer, H. and Dennis, P. P. "Modulation of chemical composition and other parameters of the cell at different exponential growth rates". In: *EcoSal Plus* 3.1 (2008) (cit. on pp. 17, 22, 24, 81, 83, 179, 190, 192, 195, 200, 202, 209).

[16] Briat, C., Gupta, A., and Khammash, M. "Antithetic Integral Feedback Ensures Robust Perfect Adaptation in Noisy Bimolecular Networks". In: *Cell Systems* 2.1 (2016), pp. 15–26. ISSN: 24054720. DOI: 10.1016/j.cels.2016.01.004. arXiv: 1410.6064. URL: http://dx.doi.org/10.1016/j.cels.2016.01.004 (cit. on pp. 3, 100).

[17] Carbonell, M. et al. "Dealing with the genetic load in bacterial synthetic biology circuits: covergences with the Ohm's law". In: *Nucleic Acids Research* 44.1 (2016), pp. 496–507 (cit. on p. 11).

[18] Cardinale, S. and Arkin, A. P. "Contextualizing context for synthetic biology - identifying causes of failure of synthetic biological systems". In: *Biotechnology Journal* 7.7 (2012), pp. 856–866. ISSN: 18606768. DOI: 10.1002/biot.201200085 (cit. on p. 10).

[19] Ceroni, F. et al. "Quantifying cellular capacity identifies gene expression designs with reduced burden". In: *Nature Methods* 12.5 (Apr. 2015), pp. 415–418. DOI: 10.1038/nmeth.3339 (cit. on p. 34).

[20] Chen, H. et al. " Genome-wide study of mRNA degradation and transcript elongation in E scherichia coli ". In: *Molecular Systems Biology* 11.1 (2015), p. 781. ISSN: 1744-4292 (cit. on p. 200).

[21] Dai, X. and Zhu, M. "Coupling of Ribosome Synthesis and Translational Capacity with Cell Growth". In: *Trends in Biochemical Sciences* 45.8 (2020), pp. 681–692 (cit. on p. 22).

[22] De Battista, H., Picó, J., and Picó-Marco, E. "Globally stabilizing control of fed-batch processes with Haldane kinetics using growth rate estimation feedback". In: *Journal of Process Control* 16.8 (2006), pp. 865–875. ISSN: 09591524. DOI: 10.1016/j.jprocont.2006.02.001 (cit. on p. 121).

[23] De Battista, H., Picó, J., and Picó-Marco, E. "Nonlinear PI control of fed-batch processes for growth rate regulation". In: *Journal of Process Control* 22.4 (2012), pp. 789–797. ISSN: 09591524. DOI: 10.1016/j.jprocont.2012.02.011 (cit. on pp. 120, 121).

[24] De Battista, H. et al. "Output Feedback Linearization of Turbidostats After Time Scaling". In: *IEEE Transactions on Control Systems Technology* (2018), pp. 1–9. ISSN: 10636536. DOI: 10.1109/TCST.2018.2834882 (cit. on pp. 6, 119).

[25] De Battista, H. et al. "Reaction rate reconstruction from biomass concentration measurement in bioreactors using modified second-order sliding mode algorithms." In: *Bioprocess Biosyst Eng* 35.9 (2012), pp. 1615–1625. DOI: 10.1007/s00449-012-0752-y (cit. on p. 139).

[26] De Jong, H. et al. "Mathematical modelling of microbes: metabolism, gene expression and growth". In: *Journal of the Royal Society Interface* 14.136 (2017), p. 20170502 (cit. on p. 77).

[27] Downey, B. et al. "A novel approach for using dielectric spectroscopy to predict viable cell volume (VCV) in early process development". In: *Biotechnology Progress* 30 (2014), pp. 479–487 (cit. on p. 121).

[28] Dunn, I. J. et al. *Biological Reaction Engineering: Dynamic Modelling Fundamentals with Simulation Examples, Second Edition*. Wiley-VCH Verlag, 2003 (cit. on pp. 121, 122).

[29] Egea, J. A. et al. "MEIGO: an open-source software suite based on metaheuristics for global optimization in systems biology and bioinformatics". In: *BMC Bioinformatics* 15.1 (2014), p. 136 (cit. on p. 210).

[30] Eriksen, M. et al. "Occlusion of the ribosome binding site connects the translational initiation frequency, mRNA stability and premature transcription termination". In: *Frontiers in Microbiology* 8.MAR (2017), pp. 1–7. ISSN: 1664302X. DOI: 10.3389/fmicb.2017.00362 (cit. on pp. 83, 200).

[31] Fan, R. et al. "An Innovative Optical Sensor for the Online Monitoring and Control of Biomass Concentration in a Membrane Bioreactor System for Lactic Acid Production". In: *Sensors* 16.3 (2016), pp. 411–423. DOI: doi:10.3390/s16030411 (cit. on p. 121).

[32] Fernandes, L. D., Moura, A. P., and Ciandrini, L. "Gene length as a regulator for ribosome recruitment and protein synthesis: Theoretical insights". In: *Scientific Reports* 7.1 (2017), pp. 1–11. ISSN: 20452322. DOI: 10.1038/s41598-017-17618-1. URL: http://dx.doi.org/10.1038/s41598-017-17618-1 (cit. on pp. 83, 200, 205).

[33] Fowler, M. *Domain Specific Languages*. 1st. Addison-Wesley Professional, 2010. ISBN: 0321712943 (cit. on p. 44).

[34] Frei, T. et al. "Characterization, modelling and mitigation of gene expression burden in mammalian cells". In: *bioRxiv* (2019). DOI: 10.1101/867549 (cit. on p. 11).

[35] Funahashi, A. et al. "CellDesigner 3.5: A versatile modeling tool for biochemical networks". In: *Proceedings of the IEEE* 96.8 (2008), pp. 1254–1265. ISSN: 00189219. DOI: 10.1109/JPROC.2008.925458 (cit. on p. 40).

[36] Giordano, N. et al. "Dynamical Allocation of Cellular Resources as an Optimal Control Problem: Novel Insights into Microbial Growth Strategies". In: *PLoS Computational Biology* 12.3 (2016), e1004802. ISSN: 15537358. DOI: 10.1371/journal.pcbi.1004802. URL: https://project.inria.fr/reset/ (cit. on p. 77).

[37] Gómez-Pérez, C. A. and Espinosa, J. "The analysis of continuous bioreactors in series with recirculation using Singular Value Decomposition". In: *Chemical Engineering Research and Design* 125 (2017), pp. 108–118 (cit. on p. 120).

[38] Gorochowski, T. E. et al. "A Minimal Model of Ribosome Allocation Dynamics Captures Trade-offs in Expression between Endogenous and Syn-

thetic Genes". In: *ACS Synthetic Biology* 5.7 (2016). ISSN: 21615063. DOI: 10.1021/acssynbio.6b00040 (cit. on pp. 11, 30, 35, 77, 78).

[39] Gorochowski, T. E. et al. "Absolute quantification of translational regulation and burden using combined sequencing approaches". In: *Molecular Systems Biology* 15.5 (May 2019) (cit. on p. 10).

[40] Großmann, P., Lück, A., and Kaleta, C. "Model-based genome-wide determination of RNA chain elongation rates in Escherichia coli". In: *Scientific Reports* 7 (2017), p. 17213. DOI: 10.1038/s41598-017-17408-9 (cit. on p. 200).

[41] Hartline, C. J. et al. "Dynamic control in metabolic engineering: Theories, tools, and applications". In: *Metabolic Engineering* July (2020). ISSN: 10967184. DOI: 10.1016/j.ymben.2020.08.015. URL: https://doi.org/10.1016/j.ymben.2020.08.015 (cit. on p. 77).

[42] Hausser, J. et al. "Central dogma rates and the trade-off between precision and economy in gene expression". In: *Nature Communications* 10.1 (2019). ISSN: 2041-1723. DOI: 10.1038/s41467-018-07391-8. URL: http://dx.doi.org/10.1038/s41467-018-07391-8 (cit. on pp. 15, 20, 21, 23, 30, 81, 83, 199–202, 205–207).

[43] Hoffmann, S. A. et al. "A user-friendly, low-cost turbidostat with versatile growth rate estimation based on an extended Kalman filter". In: *PLoS ONE* 12.7:e0181923 (2017), pp. 1–15 (cit. on p. 121).

[44] Huang, H.-H., Qian, Y., and Vecchio, D. D. "A quasi-integral controller for adaptation of genetic modules to variable ribosome demand". In: *bioRxiv* (2018). ISSN: 20411723. DOI: 10.1101/336271 (cit. on p. 118).

[45] Hucka, M. et al. "The Systems Biology Markup Language (SBML): Language Specification for Level 3 Version 2 Core Release 2". In: *Journal of integrative bioinformatics* 16.2 (2019). ISSN: 16134516. DOI: 10.1515/jib-2019-0021 (cit. on p. 40).

[46] Jabarivelisdeh, B. and Waldherr, S. "Improving Bioprocess Productivity Using Constraint-Based Models in a Dynamic Optimization Scheme". In: *IFAC-PapersOnLine* 49.26 (2016), pp. 245–251. ISSN: 24058963. DOI: 10.1016/j.ifacol.2016.12.133 (cit. on p. 77).

[47] Jabarivelisdeh, B. and Waldherr, S. "Optimization of bioprocess productivity based on metabolic-genetic network models with bilevel dynamic programming". In: *Biotechnology and Bioengineering* 115.7 (2018), pp. 1829–1841. ISSN: 10970290. DOI: 10.1002/bit.26599. URL: http://doi.wiley.com/10.1002/bit.26599 (cit. on p. 77).

[48] Jang, M. F., Chern, Y. J., and Chou, Y. S. "Robust adaptive controller for continuous bioreactors". In: *Biochemical Engineering Journal* 81 (2013), pp. 136–145 (cit. on p. 121).

[49] Jayanthi, S., Nilgiriwala, K. S., and Del Vecchio, D. "Retroactivity controls the temporal dynamics of gene transcription." In: *ACS Synth. Biol.* 2.8 (2013), pp. 431–441 (cit. on p. 11).

[50] Jeanne, G. et al. "Dynamical resource allocation models for bioreactor optimization." In: *IFAC PapersOnLine* 51.19 (2018), pp. 20–23 (cit. on pp. 11, 77).

[51] Kamp, A. von et al. "Use of CellNetAnalyzer in biotechnology and metabolic engineering". In: *Journal of Biotechnology* 261 (2017), pp. 221–228. ISSN: 18734863. DOI: 10.1016/j.jbiotec.2017.05.001. URL: https://pubmed.ncbi.nlm.nih.gov/28499817/ (cit. on p. 90).

[52] Kanehisa, M. and Goto, S. "KEGG: kyoto encyclopedi of genes and genomes". In: *Nucleic acids research* 28.1 (2000), pp. 27–30. DOI: 10.1093/nar/28.1.27 (cit. on p. 20).

[53] Kierzek, A. M., Zaim, J., and Zielenkiewicz, P. "The Effect of Transcription and Translation Initiation Frequencies on the Stochastic Fluctuations in Prokaryotic Gene Expression". In: *Journal of Biological Chemistry* 276.11 (2001), pp. 8165–8172. ISSN: 1467968X. DOI: 10.1111/1467-968X.12030 (cit. on pp. 84, 199, 200, 207).

[54] Kleman, G. et al. "Glucose-stat, a glucose-controlled continuous culture". In: *Applied and Environmental Microbiology* 57 (1991), pp. 918–923 (cit. on p. 120).

[55] Konakovsky, V. et al. "Universal Capacitance Model for Real-Time Biomass in Cell Culture". In: *Sensors* 15 (2015), pp. 22128–22150 (cit. on p. 121).

[56] Kubitschek, H. E. et al. “Independence of buoyant cell density and growth rate in Escherichia coli.” In: *J Bacteriol.* 158.1 (1984), pp. 296–299 (cit. on p. 187).

[57] Lara-Cisneros, G., Alvarez-Ramirez, J., and Femat, R. “Self-optimising control of a class of continuous bioreactor via variable-structure feedback”. In: *Int. J. of Systems Science* 47.6 (2016), pp. 1394–1406 (cit. on p. 121).

[58] Lee, K. S. et al. “Microfluidic chemostat and turbidostat with flow rate, oxygen, and temperature control for dynamic continuous culture.” In: *Lab Chip* 11.10 (2011), pp. 1730–1739. ISSN: 1473-0189. DOI: 10.1039/c1lc20019d (cit. on p. 121).

[59] Li, G. W. et al. “Quantifying absolute protein synthesis rates reveals principles underlying allocation of cellular resources”. In: *Cell* 157.3 (2014), pp. 624–635. ISSN: 10974172. DOI: 10.1016/j.cell.2014.02.033. URL: http://dx.doi.org/10.1016/j.cell.2014.02.033 (cit. on pp. 81, 196).

[60] Liu, D. et al. “Dynamic metabolic control: towards precision engineering of metabolism”. In: *Journal of Industrial Microbiology and Biotechnology* (2018). ISSN: 14765535. DOI: 10.1007/s10295-018-2013-9 (cit. on p. 77).

[61] Liu, Y. “Overview of some theoretical approaches for derivation of the Monod equation.” In: *Appl Microbiol Biotechnol* 73 (2007), pp. 1241–1250. DOI: 10.1007/s00253-006-0717-7 (cit. on p. 188).

[62] Llaneras, F. and Picó, J. “Stoichiometric modelling of cell metabolism”. In: *Journal of Bioscience and Bioengineering* 105.1 (2008), pp. 1–11. ISSN: 13891723. DOI: 10.1263/jbb.105.1 (cit. on p. 77).

[63] Llaneras, F., Sala, A., and Picó, J. “Dynamic estimations of metabolic fluxes with constraint-based models and possibility theory”. In: *Journal of Process Control* 22.10 (2012), pp. 1946–1955. ISSN: 09591524. DOI: 10.1016/j.jprocont.2012.09.001 (cit. on p. 77).

[64] Lloyd-Price, J. et al. “Dissecting the stochastic transcription initiation process in live Escherichia coli”. In: *DNA Research* 23.3 (2016), pp. 203–214. DOI: 10.1093/dnares/dsw009 (cit. on p. 200).

[65] Lorenzo, V. de. "Evolutionary tinkering vs. rational engineering in the times of synthetic biology." In: *Life Sci Soc Policy* 14.18 (2018), pp. 1263–1272. DOI: 10.1186/s40504-018-0086-x (cit. on p. 10).

[66] Macklin, D. N. et al. "Simultaneous cross-evaluation of heterogeneous E. coli datasets via mechanistic simulation". In: *Science* 369.6502 (2020). DOI: 10.1126/science.aav3751 (cit. on pp. 11, 77).

[67] Maggioli, F., Mancini, T., and Tronci, E. "SBML2Modelica: Integrating biochemical models within open-standard simulation ecosystems". In: *Bioinformatics* 36.7 (2020), pp. 2165–2172. ISSN: 14602059. DOI: 10.1093/bioinformatics/btz860 (cit. on pp. 41, 42).

[68] Mahadevan, R., Edwards, J. S., and Doyle, F. J. "Dynamic flux balance analysis of diauxic growth". In: *Biophysical Journal* 83.3 (2002), pp. 1331–1340 (cit. on p. 77).

[69] Mailleret, L., Bernard, O., and Steyer, J.-P. "Nonlinear adaptive control for bioreactors with unknown kinetics". In: *Automatica* 40.8 (2004), pp. 1379–1385 (cit. on pp. 121, 137).

[70] Mallavarapu, A. et al. "Programming with models: Modularity and abstraction provide powerful capabilities for systems biology". In: *Journal of the Royal Society Interface* 6.32 (2009), pp. 257–270. ISSN: 17425662. DOI: 10.1098/rsif.2008.0205 (cit. on p. 40).

[71] Marcos, N. I., Guay, M., and Dochain, D. "Output feedback adaptive extremum seeking control of a continuous stirred tank bioreactor with Monod's kinetics". In: *Journal of Process Control* 14 (2004), pp. 807–818 (cit. on p. 121).

[72] Mazenc, F., Harmand, J., and Malisoff, M. "Stabilization in a chemostat with sampled and delayed measurements and uncertain growth functions". In: *Automatica* 78 (2017), pp. 241–249 (cit. on pp. 120, 121).

[73] Medley, J. K. et al. "Tellurium notebooks—An environment for reproducible dynamical modeling in systems biology". In: *PLoS Computational Biology* 14.6 (2018), pp. 1–24. ISSN: 15537358. DOI: 10.1371/journal.pcbi.1006220 (cit. on p. 43).

[74] Metzl-Raz, E. et al. “Principles of cellular resource allocation revealed by condition-dependent proteome profiling”. In: *eLife* 6 (Aug. 2017). Ed. by P. J. Wittkopp, e28034 (cit. on p. 22).

[75] Milias-Argeitis, A. et al. “Automated optogenetic feedback control for precise and robust regulation of gene expression and cell growth”. In: *Nature Communications* (2016). ISSN: 20411723. DOI: 10.1038/ncomms12546 (cit. on p. 121).

[76] Milo, R., Phillips, R., and Orme, N. *Cell Biology by the Numbers*. Garland Science, 2016 (cit. on pp. 84, 138).

[77] Milo, R. et al. “BioNumbers–the database of key numbers in molecular and cell biology”. In: *Nucleic acids research* 38.Database issue (Jan. 2010), pp. D750–3. DOI: 10.1093/nar/gkp889 (cit. on pp. 20, 83, 185, 200, 205).

[78] Myers, C. J. et al. “iBioSim: A tool for the analysis and design of genetic circuits”. In: *Bioinformatics* 25.21 (2009), pp. 2848–2849. ISSN: 13674803. DOI: 10.1093/bioinformatics/btp457 (cit. on p. 40).

[79] Nikolados, E. M. et al. “Growth Defects and Loss-of-Function in Synthetic Gene Circuits”. In: *ACS Synthetic Biology* 8.6 (2019), pp. 1231–1240. ISSN: 21615063. DOI: 10.1021/acssynbio.8b00531 (cit. on p. 34).

[80] Oyarzún, D. A. and Stan, G. B. V. “Synthetic gene circuits for metabolic control: Design trade-offs and constraints”. In: *Journal of the Royal Society Interface* 10.78 (2013) (cit. on p. 30).

[81] Perrier, M. and Dochain, D. “Evaluation of control strategies for anaerobic digestion processes”. In: *International Journal of Adaptive Control and Signal Processing* 7.4 (1993), pp. 309–321 (cit. on p. 121).

[82] Picard, F. et al. “The significance of translation regulation in the stress response”. In: *BMC Genomics* 14.1 (2013). ISSN: 14712164. DOI: 10.1186/1471-2164-14-588 (cit. on pp. 83, 200, 205).

[83] Picó-Marco, E., Picó, J., and De Battista, H. “Sliding mode scheme for adaptive specific growth rate control in biotechnological fed-batch processes”. In: *Int. J. of Control* 78.2 (2005), pp. 128–141 (cit. on p. 121).

[84] Picó, J. et al. “Stability preserving maps for finite-time convergence: Super-twisting sliding-mode algorithm”. In: *Automatica* 49.2 (2013), pp. 534–539 (cit. on pp. 121, 126, 129, 139).

[85] Poole, W. et al. “BioCRNpyler: Compiling Chemical Reaction Networks from Biomolecular Parts in Diverse Contexts”. In: *bioRxiv* (2020). DOI: 10.1101/2020.08.02.233478. arXiv: 2020.08.02.233478. URL: https://doi.org/10.1101/2020.08.02.233478 (cit. on p. 40).

[86] Proll, T. and Karim, N. “Nonlinear control of a bioreactor model using exact and {I/O} linearization”. In: *International Journal of Control* 60.4 (1994), pp. 499–519 (cit. on p. 121).

[87] Purdy, H. M. and Reed, J. L. *Evaluating the capabilities of microbial chemical production using genome-scale metabolic models*. 2017. DOI: 10.1016/j.coisb.2017.01.008 (cit. on p. 77).

[88] Qian, Y. and Del Vecchio, D. “Realizing ‘integral control’ in living cells: how to overcome leaky integration due to dilution?” In: *J. R. Soc. Interface* 15.139 (2018) (cit. on p. 100).

[89] Qian, Y. et al. “Resource competition shapes the response of genetic circuits”. In: *ACS Synth. Biol.* 6.7 (2012), pp. 1263–1272 (cit. on pp. 10, 11, 33).

[90] Qiu, J., Arnold, M. A., and Murhammer, D. W. “On-line near infrared bioreactor monitoring of cell density and concentrations of glucose and lactate during insect cell cultivation”. In: *Journal of Biotechnology* 173 (2014), pp. 106–111 (cit. on p. 121).

[91] Raftery, J. P., DeSessa, M. R., and Karim, M. N. “Economic improvement of continuous pharmaceutical production via the optimal control of a multifeed bioreactor”. In: *Biotechnol. Prog.* 33.4 (2017), pp. 902–912 (cit. on p. 120).

[92] Rosano, G. L. and Ceccarelli, E. A. “Recombinant protein expression in Escherichia coli: advances and challenges”. In: *Frontiers in Microbiology* 5 (2014), p. 172 (cit. on p. 26).

[93] Sabi, R. and Tuller, T. “Modeling and measuring intracellular competition for finite resources during gene expression”. In: *J. R. Soc. Interface* 16 (2019), p. 20180887 (cit. on pp. 10, 77).

[94] Saldanha, A., Brauer, M., and Botstein. D. “Nutritional homeostasis in batch and steady-state culture of yeast.” In: *Molecular Biology of the Cell* 15 (2014), pp. 4089–4104 (cit. on p. 120).

[95] Salis, H. M. “Chapter two - The Ribosome Binding Site Calculator”. In: *Synthetic Biology, Part B*. Ed. by C. Voigt. Vol. 498. Methods in Enzymology. Academic Press, 2011, pp. 19–42. DOI: https://doi.org/10.1016/B978-0-12-385120-8.00002-4 (cit. on pp. 84, 200).

[96] Santos-Navarro, F. N., Navarro, J. L., and Picó, J. “SBModEns : A Modular Toolbox for Model Building , Reduction , Analysis and Simulation in System Biology”. In: *Proceedings of 12th IWBDA* (2020), pp. 49–50. URL: https://www.iwbdaconf.org/2020/docs/IWBDA2020Proceedings.pdf (cit. on p. 6).

[97] Santos-Navarro, F. N. and Picó, J. “Minimal model for protein expression accounting for metabolic burden”. In: *Proceedings of 12th IWBDA* (2020), pp. 41–42. URL: https://www.iwbdaconf.org/2020/docs/IWBDA2020Proceedings.pdf (cit. on p. 6).

[98] Santos-Navarro, F. N. et al. “Gene Expression Space Shapes the Bioprocess Trade-Offs among Titer, Yield and Productivity”. In: *Applied Sciences* 11.13 (2021), p. 5859. ISSN: 2076-3417. DOI: 10.3390/app11135859 (cit. on pp. 6, 75).

[99] Santos-Navarro, F. N. et al. “OneModel: an open-source SBML modeling tool focused on accessibility, simplicity, and modularity”. In: *DYCOPS* (2022), accepted for publication (cit. on pp. 6, 39).

[100] Santos-Navarro, F. N. et al. “RBS and Promoter Strengths Determine the Cell-Growth-Dependent Protein Mass Fractions and Their Optimal Synthesis Rates”. In: *ACS Synthetic Biology* 10.12 (2021), pp. 3290–3303. ISSN: 21615063. DOI: 10.1021/acssynbio.1c00131 (cit. on pp. 6, 7, 9, 58, 83–85, 90, 100, 167).

[101] Santos-Navarro, F. N. et al. "Reference Conditioning Anti-windup for the Biomolecular Antithetic Controller". In: *IFAC-PapersOnLine* 52.26 (Jan. 2019), pp. 156–162. ISSN: 2405-8963. DOI: 10.1016/J.IFACOL.2019.12.251 (cit. on pp. 6, 99, 117).

[102] Santos-Navarro, F. N. and Picó, J. "Resources allocation explains the differential roles of RBS and promoter strengths in cell mass distribution and optimal protein expression productivity". In: *bioRxiv* (2020). DOI: 10.1101/2020.11.19.390583. arXiv: 2020.11.19.390583 (cit. on p. 7).

[103] Savoglidis, G. and Kravaris, C. "Constant-yield control of continuous bioreactors". In: *Chemical Engineering Journal* 228 (2013), pp. 1234–1247 (cit. on p. 121).

[104] Schaum, A., Alvarez, J., and Lopez-Arenas, T. "Saturated PI control of continuous bioreactors with Haldane kinetics". In: *Chemical Engineering Science* 68 (2012), pp. 520–529 (cit. on pp. 120, 121).

[105] Scott, M. et al. "Interdependence of Cell Growth and Gene Expression: Origins and Consequences". In: *Science* 330.6007 (2010), pp. 1099–1102. DOI: 10.1126/science.1192588 (cit. on pp. 11, 12, 22, 77, 120, 200).

[106] Shaham, G. and Tuller, T. "Genome scale analysis of Escherichia coli with a comprehensive prokaryotic sequence-based biophysical model of translation initiation and elongation". In: *DNA Research* 25.2 (2017), pp. 195–205. DOI: 10.1093/dnares/dsx049 (cit. on p. 200).

[107] Shi, S., Ang, E. L., and Zhao, H. "In vivo biosensors: mechanisms, development, and applications". In: *Journal of Industrial Microbiology and Biotechnology* 45.7 (2018), pp. 491–516. ISSN: 14765535. DOI: 10.1007/s10295-018-2004-x. URL: https://link.springer.com/article/10.1007/s10295-018-2004-x (cit. on p. 76).

[108] Si, F. et al. "Invariance of Initiation Mass and Predictability of Cell Size in Escherichia coli". In: *Current Biology* 27.9 (2017), pp. 1278–1287 (cit. on pp. 17, 190, 191).

[109] Siwiak, M. and Zielenkiewicz, P. "Transimulation - Protein Biosynthesis Web Service". In: *PLoS ONE* 8.9 (2013). ISSN: 19326203. DOI: 10.1371/journal.pone.0073943 (cit. on pp. 83, 200).

[110] Smit, M. H. de and Duin, J. van. "Translational Standby Sites: How Ribosomes May Deal with the Rapid Folding Kinetics of mRNA." In: *Journal of Molecular Biology* 331.4 (2003), pp. 737–743. DOI: 10.1016/S0022-2836(03)00809-X (cit. on pp. 84, 200).

[111] Smith, L. P. et al. "Antimony: A modular model definition language". In: *Bioinformatics* 25.18 (2009), pp. 2452–2454. ISSN: 13674803. DOI: 10.1093/bioinformatics/btp401 (cit. on p. 40).

[112] St John, P. C., Crowley, M. F., and Bomble, Y. J. "Efficient estimation of the maximum metabolic productivity of batch systems". In: *Biotechnology for Biofuels* 10.1 (2017), p. 28. ISSN: 17546834. DOI: 10.1186/s13068-017-0709-0. arXiv: 1610.01114. URL: https://biotechnologyforbiofuels.biomedcentral.com/articles/10.1186/s13068-017-0709-0 (cit. on p. 90).

[113] Sundararaj, S. et al. "The CyberCell Database (CCDB): A comprehensive, self-updating, relational database to coordinate and facilitate in silico modeling of Escherichia coli". In: *Nucleic Acids Research* 32.DATABASE ISS. (2004), pp. 293–295. ISSN: 03051048. DOI: 10.1093/nar/gkh108 (cit. on p. 83).

[114] Sundararaj, S. et al. "The CyberCell Database (CCDB): a comprehensive, self-updating, relational database to coordinate and facilitate in silico modeling of Escherichia coli". In: *Nucleic Acids Research* 32.suppl1 (2004), pp. D293–D295. DOI: 10.1093/nar/gkh108 (cit. on p. 200).

[115] Szederkényi, G. et al. "Nonlinear analysis and control of a continuous fermentation process". In: *Computers and Chemical Engineering* 26 (2002), pp. 659–670 (cit. on p. 121).

[116] Takahashi, C. N. et al. "A low cost, customizable turbidostat for use in synthetic circuit characterization." In: *ACS Synth Biol* (2014). ISSN: 2161-5063. DOI: 10.1021/sb500165g (cit. on pp. 121, 138).

[117] Towbin, B. D. et al. "Optimality and sub-optimality in a bacterial growth law". In: *Nature Communications* 8 (2017), p. 14123 (cit. on p. 10).

[118] Trabelsi, H., Koch, M., and Faulon, J.-L. "Building a minimal and generalizable model of transcription factorised biosensors: Showcasing flavonoids".

In: *Biotechnology and Bioengineering* 115 (2018), pp. 2292–2304 (cit. on p. 176).

[119] Ugalde-Salas, P. et al. "Insights from Microbial Transition State Theory on Monod's Affinity Constant". In: *Scientific Reports* 10.1 (2020), p. 5323. DOI: 10.1038/s41598-020-62213-6 (cit. on p. 188).

[120] Vargas, A., Moreno, J. A., and Vande Wouwer, A. "A weighted variable gain super-twisting observer for the estimation of kinetic rates in biological systems". In: *Journal of Process Control* 24.6 (2014), pp. 957–965. ISSN: 09591524. DOI: 10.1016/j.jprocont.2014.04.018 (cit. on p. 121).

[121] Venayak, N. et al. "MoVE identifies metabolic valves to switch between phenotypic states". In: *Nature Communications* 9.1 (2018), pp. 1–9. ISSN: 20411723. DOI: 10.1038/s41467-018-07719-4. URL: https://doi.org/10.1038/s41467-018-07719-4 (cit. on pp. 77, 97, 144).

[122] Vree, J. H. de et al. "Turbidostat operation of outdoor pilot-scale photobioreactors". In: *Algal Research* 18 (2016), pp. 198–208 (cit. on p. 121).

[123] Waldherr, S., Oyarzún, D. A., and Bockmayr, A. "Dynamic optimization of metabolic networks coupled with gene expression". In: *Journal of Theoretical Biology* 365 (2015), pp. 469–485. ISSN: 10958541. DOI: 10.1016/j.jtbi.2014.10.035. arXiv: 1309.4936 (cit. on p. 77).

[124] Wehrs, M. et al. *Engineering Robust Production Microbes for Large-Scale Cultivation*. 2019. DOI: 10.1016/j.tim.2019.01.006 (cit. on p. 76).

[125] Weiße, A. Y. et al. "Mechanistic links between cellular trade-offs, gene expression, and growth". In: *Proceedings of the National Academy of Sciences* 112.9 (2015), E1038–E1047. DOI: 10.1073/pnas.1416533112 (cit. on pp. 11, 17, 30, 34, 37, 77, 100, 188, 189).

[126] Wu, S. G. et al. "Facilitate Collaborations among Synthetic Biology, Metabolic Engineering and Machine Learning". In: *ChemBioEng Reviews* 3.2 (2016), pp. 45–54. ISSN: 21969744. DOI: 10.1002/cben.201500024. URL: http://doi.wiley.com/10.1002/cben.201500024 (cit. on p. 77).

[127] Zeng, H. and Yang, A. "Bridging substrate intake kinetics and bacterial growth phenotypes with flux balance analysis incorporating proteome allo-

cation". In: *Scientific Reports* 10.1 (2020), p. 4283. DOI: 10.1038/s41598-020-61174-0 (cit. on p. 188).

[128] Zheng, H. et al. "General quantitative relations linking cell growth and the cell cycle in Escherichia coli". In: *Nature Microbiology* (2020) (cit. on pp. 17, 188, 190).

[129] Zhuang, K. et al. "Dynamic strain scanning optimization: An efficient strain design strategy for balanced yield, titer, and productivity. DySScO strategy for strain design". In: *BMC Biotechnology* 13 (2013). ISSN: 14726750. DOI: 10.1186/1472-6750-13-8 (cit. on pp. 83, 84, 86, 88).

Appendices

Appendix A

Derivation of the host-aware model

The contents of this chapter appeared in the following journal publication as the Supplementary Information:

- Santos-Navarro, F. N. et al. "RBS and Promoter Strengths Determine the Cell-Growth-Dependent Protein Mass Fractions and Their Optimal Synthesis Rates". In: *ACS Synthetic Biology* 10.12 (2021), pp. 3290–3303. ISSN: 21615063. DOI: 10.1021/acssynbio.1c00131

A.1 Modeling gene expression

In this section we obtain the dynamics for the polysomic expression of a protein-coding gene in a bacterial cell. We introduce the *resources recruitment strength* , a dimensionless function that expresses the capacity of a protein-coding gene to engage cellular resources to get expressed. The resources recruitment strengths serves as basis for all posterior analysis of the partitioning of the cell mass between ribosomal and non-ribosomal endogenous proteins, the effects caused by the expression of exogenous protein-coding genes, and the roles played by the RBS and promoter strengths.

In our model, we consider a set of basic assumptions:

1. Transcription dynamics is fast enough as compared to translation so it can be considered at quasi-steady state (QSS).
2. The main resources-dependent process in protein expression is translation. Therefore:
   (a) only ribosomes are considered as limiting shared resource required for protein expression. RNA polymerase is not considered explicitly.
   (b) the effective translation rate is assumed to depend on the availability of intracellular substrate. This is implicitly considering that building the polypeptide protein chain is the limiting energy-consuming process in the cell. We do not explicitly consider the catabolic conversion of substrate into amino acid building blocks.
3. Several ribosomes (a polysome) may translate a single messenger RNA (mRNA) simultaneously.
4. We identify a transcriptional unit by its promoter.

With these assumptions in mind, we modeled the expression of a given gene expressing the protein p_k by means of the set of pseudo-reactions (A.1).

$$
\begin{aligned}
\mathrm{g_k} &\xrightarrow{k(T_f)} \mathrm{g_k} + \mathrm{m_k} \\
\mathrm{m_k} + \mathrm{r} &\underset{K_u^k}{\overset{K_b^k}{\rightleftharpoons}} \mathrm{C_{k^0}} \\
\mathrm{C_{k^0}} &\xrightarrow{K_e(s_i)} \mathrm{C_{k^1}} + \mathrm{m_k} \\
\mathrm{C_{k^1}} &\xrightarrow{K_k(s_i)} \mathrm{p_k} + \mathrm{r} \\
\mathrm{C_{k^1}} &\xrightarrow{K_e(s_i)} \mathrm{C_{k^2}} \\
&\vdots \\
\mathrm{C_{k^{j-1}}} &\xrightarrow{K_k(s_i)} \mathrm{p_k} + \mathrm{r} \\
\mathrm{C_{k^{j-1}}} &\xrightarrow{K_e(s_i)} \mathrm{C_{k^j}} \\
&\vdots \\
\mathrm{C_{k^{n_r-1}}} &\xrightarrow{K_e(s_i)} \mathrm{C_{k^{n_r}}} \\
\mathrm{C_{k^{n_r}}} &\xrightarrow{K_k(s_i)} \mathrm{p_k} + \mathrm{r} \\
\mathrm{m_k} &\xrightarrow{d_{mk}+} \emptyset \\
\mathrm{p_k} &\xrightarrow{d_k+} \emptyset \\
\mathrm{C_{k^j\,j=0\ldots n_r}} &\longrightarrow \emptyset
\end{aligned}
\tag{A.1}
$$

where the species involved are:

g_k : free copy of the promoter, ie. of the gene transcriptional unit.

m_k : free ribosome binding site (RBS), ie. mRNA copy with its RBS free.

r : free ribosome.

C_{k^0} : complex formed by a ribosome bound to the RBS in a mRNA

C_{k^j} : j-th translating complex formed by j ribosomes simultaneously translating a mRNA copy with freed RBS. Each C_{k^j} species gives rise to a peptide chain which is in different stages of production.

p_k : protein abundance

s_i : intracellular substrate (molecules)

and the specific reaction rates stand for:

$\omega_k(T_f)$: transcription rate. It may be a function of one or several transcription factors (TF) and may include the gene copy number.

K_b^k : association rate constant between a free ribosome and the RBS.

K_u^k : dissociation rate constant between a free ribosome and the RBS.

$K_e(s_i)$: translation initiation rate. constant.

$K_k(s_i)$: translation elongation rate constant.

d_{mk} : mRNA degradation rate constant.

d_k : protein degradation rate constant.

μ : cell specific growth rate.

For a protein of length l_{pk} amino acids, we denote the mRNA length:

$$l_{mk} = l_e + l_{pk}\,, \tag{A.2}$$

where $1/l_e$ is the ribosomes density, with l_e expressed as equivalent number of codons. We consider the effective RBS length to be the same as l_e. The length of the protein is denoted as l_{pk}. Thus, up to n_r ribosomes can simultaneously be translating a single copy of mRNA, with $n_r = l_{pk}/l_e$, and an additional ribosome is bound to the RBS.

We consider the effective rates constants $K_e(s_i)$ and $K_k(s_i)$ at which the ribosome glides through the RBS and the remaining mRNA nucleotides respectively. Thus, we consider:

$$K_e(s_i) = \frac{\nu_t(s_i)}{l_e}\,, \quad K_k(s_i) = \frac{\nu_t(s_i)}{l_{xk}}\,, \tag{A.3}$$

with

$$\nu_t(s_i) = \frac{\nu s_i}{K_{sc} + s_i}\,, \tag{A.4}$$

where ν is the maximum attainable translation rate (peptide synthesis rate) and K_{sc} is a Michaelis-Menten parameter related to the cell substrate uptake and catabolic capacity. As a first approximation, we consider that ν is organism dependent but does not depend on the sequence of nucleotides, and K_{sc} is organism and substrate dependent but does not depend on the nucleotides sequence either.

The translating complexes $C_{k^{j-1}}$ are pseudo-species modeling the process of parallel translation. They can loosely be identified with each of the chains of amino acids under formation plus its associated ribosome. The first one, C_{k^0}, represents the ribosome bound to the RBS. Notice with rate $K_e(s_i)$ the ribosome bound to the RBS, forming C_{k^0}, advances to the next ribosome occupancy slot, generating the translating complex C_{k^1} and freeing the RBS so a new ribosome can enter in the queue. In turn, the displacement of the ribosomes in the queue by one occupancy generates the translating complexes C_{k^j} from the previous complex $C_{k^{j-1}}$. Thus, each C_{k^j} species gives rise to a peptide chain which is in different stages of production. Finally, each parallel translating complex C_{k^j} generates a protein with rate $K_k(s_i)$ and frees its bound ribosome. Recall the translating complexes can be identified with the ensemble formed by each of the chains of amino acids under formation and their respective associated ribosomes. This way, the queue dynamics of the ribosomes advancing along the mRNA is decoupled from the protein building ones, thus getting a continuous approximation of the polysomic process of translation. This continuous approximation results in an exponential distribution of the amounts of pseudo-complexes C_{k^j}.

Figure A.1 depicts the process. The amounts of the pseudo-complexes fulfill $C_{k^j} < C_{k^{j-1}}$ (see equation (A.9)). Thus, starting from C_{k^0}, at each time instant a fraction of it remains and a smaller fraction becomes C_{k^1} because of the continuous shuffling up. At the next differential time instant, on the one hand a fraction of C_{k^1} remains and a smaller fraction becomes C_{k^2} and, on the other hand, a fraction of the C_{k^0} which remained in the previous time step remains and a smaller fraction becomes C_{k^1}. This process of generation of the amounts of pseudo-complexes takes place at each differential time instant, eventually resulting in an exponential distribution of the amounts of pseudo-complexes C_{k^j}. In parallel (i.e. simultaneously) to the process above, each of the pseudo-complexes C_{k^j} generates a peptide chain through the pseudo-reactions $\mathrm{C_{k^j}} \xrightarrow{K_k(s_i)} \mathrm{p_k} + \mathrm{r}$. As a result the protein synthesis rate $K_k(s_i)\sum_{j=1}^{n_r} C_{k^j}$ (see equation (A.5)) will effectively be weighting the amount of proteins being synthesized at each of their synthesis stages. Recall again that the pseudocomplexes C_{k^j} can loosely be identified with each of the chains of amino acids under formation plus its associated ribosome. The less abundant ones at the later stages of the peptide chain synthesis (close or at $C_{k^{n_r}}$) are weighted much less than the more abundant shorter ones at earlier

stages. This way, the evaluation of the protein synthesis rate takes into account an approximation of the distribution profile of the lengths of the peptide chains being synthesized by the ribosomes along the transcript.

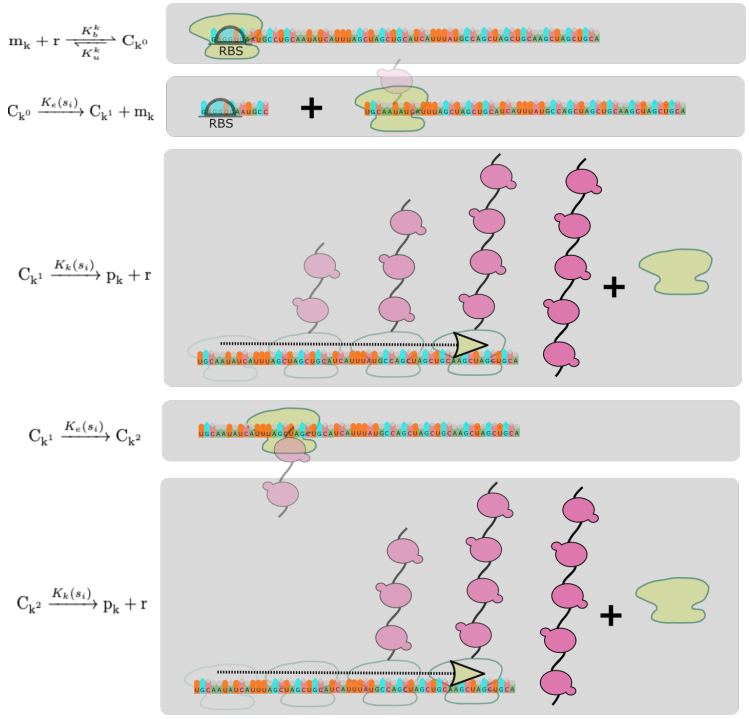

Figure A.1: Modeling polysomic translation. From top to bottom. **(1)** For a k-th protein-coding gene, a free ribosome binds to m_k, a mRNA copy with its RBS free, producing the initial pseudocomplex C_{k^0}: an avatar for a transcript with a ribosome bound to its RBS. **(2)** The ribosome shuffles up one space with rate constant equal to the translation initiation one $K_e(s_i)$ leaving the same mRNA copy with a free RBS, m_k. In addition, there will be a ribosome placed at the first slot of the transcript protein coding region This ensemble is represented by the pseudocomplex C_{k^1}. Thus, though physically there is only one transcript copy with a free RBS and a ribosome placed at the first slot of the transcript coding region, we model this situation as having two pseudospecies: m_k and C_{k^1}. The mRNA is again free to be bound by a new free ribosome, allowing new ribosomes to continually load into the system. Thus, ribosomes enter the translation process through the continuous "cycling" between the second and third pseudo-reactions in Equation (A.1) and their shuffling up through the chain pseudo-reactions $\mathrm{C_{k^{j-1}}} \xrightarrow{K_e(s_i)} \mathrm{C_{k^j}}$. **(3)** C_{k^1} becomes an independent pseudocomplex consisting of a ribosome placed at the first slot of the mRNA coding region and gliding along the transcript with translation elongation rate constant $K_k(s_i)$. This virtual representation allows to consider the pseudoreaction $\mathrm{C_{k^1}} \xrightarrow{K_k(s_i)} \mathrm{p_k + r}$. Eventually, one copy of the protein is synthesized and the ribosome is freed. **(4)** When the ribosome has initiated translation in C_{k^1} all other ribosomes will shuffle up one space in the physical mRNA. In our model we shuffle up the pseudocomplexes C_{kj}. Thus, C_{k^1} becomes C_{k^2} at rate $K_e(s_i)$ at which the entering free ribosome into the system "pushes" the queue to shuffle up. **(5)** A second parallel translation $C_{k^2} \rightarrow p_k + r$ takes place. Both C_{k^1} and C_{k^2} simultaneously synthesize a protein and eventually free their bound ribosomes (one per complex).

Next, we apply mass action kinetics to obtain the dynamic balances for the copy number of each species in the model. This way, we have:

$$\begin{aligned}
\dot{m}_k &= \omega_k(T_f) - K_b^k m_k r + K_u^k C_{k^0} + K_e(s_i) C_{k^0} - (d_{mk} + \mu) m_k \,, \\
\dot{C}_{k^0} &= K_b^k m_k r - K_u^k C_{k^0} - K_e(s_i) C_{k^0} - \mu C_{k^0} \,, \\
\dot{C}_{k^1} &= K_e(s_i) C_{k^0} - K_e(s_i) C_{k^1} - K_k(s_i) C_{k^1} - \mu C_{k^1} \,, \\
&\vdots \\
\dot{C}_{k^{j-1}} &= K_e(s_i) C_{k^{j-2}} - K_e(s_i) C_{k^{j-1}} - K_k(s_i) C_{k^{j-1}} - \mu C_{k^{j-1}} \,, \\
&\vdots \\
\dot{C}_{k^{n_r}} &= K_e(s_i) C_{k^{n_r-1}} - K_e(s_i) C_{k^{n_r}} - K_k(s_i) C_{k^{n_r}} - \mu C_{k^{n_r}} \,, \\
\dot{p}_k &= K_k(s_i) \sum_{j=1}^{n_r} C_{k^j} - (d_k + \mu) p_k \,.
\end{aligned} \tag{A.5}$$

Recall we assume that transcription dynamics is fast enough as compared to translation so it can be considered at quasi-steady state (QSS). We also assume the binding-unbinding dynamics to form the translation complexes C_{k^j} are fast enough so that we can also consider the number of each of the complexes quickly reaches steady state. Therefore, from $\dot{m}_k = 0$ and $\dot{C}_{k^j, j=1\ldots n_r} = 0$ we get:

$$C_{k^0} = \omega_k(T_f) \frac{1}{\frac{d_{mk}+\mu}{\frac{K_b^k}{K_u^k + K_e(s_i) + \mu}} + \mu r} r \tag{A.6}$$

and

$$C_{k^j} = \frac{K_e(s_i)}{K_e(s_i) + K_k(s_i) + \mu} C_{k^{j-1}} \,, \quad j = 1 \ldots n_r \,. \tag{A.7}$$

In practice, $d_{mk} \gg \mu$ and $k_u^k \gg \mu$ (see Table A.2). Therefore the magnitude of the specific growth rate μ can be neglected with respect to both the mRNA degradation rate constant d_{mk} and the sums $K_u^k + K_e(s_i)$ and $K_e(s_i) + K_k(s_i)$ respectively so that we can approximate:

$$C_{k^0} \approx \omega_k(T_f) \frac{1}{\frac{d_{mk}}{K_{C^0}^k} + \mu r} r \tag{A.8}$$

and

$$C_{k^j} \approx \frac{K_e(s_i)}{K_e(s_i) + K_k(s_i)} C_{k^{j-1}}\,, \quad j = 1 \ldots n_r\,, \tag{A.9}$$

where we have defined:

$$K^k_{C^0}(s_i) \triangleq \frac{K^k_b}{K^k_u + K_e(s_i)}\,. \tag{A.10}$$

Notice $K^k_{C^0}$ is directly related to the RBS strength.

Now, the dynamics for the abundance of the protein p_k can be obtained from the equations (A.5) and (A.9) as:

$$\begin{aligned} \dot{p}_k &= K_k(s_i) \sum_{j=1}^{n_r} C_{k^j} - (d_k + \mu) p_k\,, \\ &= K_k(s_i) a \left[1 + a + \ldots + a^{n_r - 1}\right] C_{k^0} - (d_k + \mu) p_k\,, \end{aligned} \tag{A.11}$$

with

$$a \triangleq \frac{K_e(s_i)}{K_e(s_i) + K_k(s_i)}\,. \tag{A.12}$$

Notice the geometric sum $1 + a + \ldots + a^{n_r - 1}$, with $a < 1$, gives:

$$\begin{aligned} K_k(s_i) a \left[1 + a + \ldots + a^{n_r - 1}\right] &= K_k(s_i) a \frac{1 - a^{n_r}}{1 - a} \\ &= K_e(s_i) \left[1 - \left(\frac{K_e(s_i)}{K_e(s_i) + K_k(s_i)}\right)^{n_r}\right]. \end{aligned} \tag{A.13}$$

Using the definitions (A.3), notice $a = l_{pk}/(l_e + l_{pk}) = l_{pk}/l_{mk}$ so we get the expression:

$$K_k(s_i) a \left[1 + a + \ldots + a^{n_r - 1}\right] = K_e(s_i) \left[1 - \left(\frac{l_{pk}}{l_{mk}}\right)^{\frac{l_{pk}}{l_e}}\right], \tag{A.14}$$

where we have taken into account that the maximum number of ribosomes bound to active translating complexes $C_{k^j, j=1\ldots n_r}$ simultaneously translating a mRNA molecule is $n_r = l_{pk}/l_e$, and we assume this maximum is always reached.

Recall we have assumed that transcription is not a limiting process, so we can express the effective transcription rate:

$$\omega_k(T_f) = \frac{\eta}{l_{mk}} F_k(T_f)\,, \tag{A.15}$$

where η (codons/min) is the maximum transcription speed and $F_k(T_f)$ is the transcription characteristic function that may depend on one or several transcription factors. By default, we assume the gene copy number c_{nk} is one. If this is not the case, the effective transcription rate $\omega_k(T_f)$ must be multiplied by c_{nk}. Notice in this case the transcription characteristic function $F_k(T_f)$ may depend on the gene copy number as described in [118].

As commented in our preliminary assumptions, we do not model competition for RNA polymerases, which would also affect the effective transcription rate. Yet, notice that, even if there are no cognate transcription factors associated, the term $F_k(T_f)$ can be used to account for competition for RNA polymerases preventing the effective transcription from proceeding at its maximum rate η/l_{mk}, sequence-dependent affinity of the promoter for the RNA polymerases (promoter strength) and the effect of nucleotides usage on the transcription speed. In summary, the term $F_k(T_f)$ can be used to accommodate aspects affecting transcription so that $\omega_k(T_f)$ is the effective transcription rate.

Thus, the dynamics for protein expression become:

$$\begin{aligned}
\dot{p}_k &= K_e(s_i)\left[1 - \left(\frac{l_{pk}}{l_{mk}}\right)^{\frac{l_{pk}}{l_e}}\right] C_{k^0} - (d_k + \mu)p_k\,, \\
&= K_e(s_i)\left[1 - \left(\frac{l_{pk}}{l_{mk}}\right)^{\frac{l_{pk}}{l_e}}\right] \frac{\eta}{l_{mk}} F_k(T_f) \frac{1}{\frac{d_{mk}}{K^k_{C^0}} + \mu r} r - (d_k + \mu)p_k\,, \\
&= \frac{\nu}{l_e}\frac{s_i}{K_{sc} + s_i}\left[1 - \left(\frac{l_{pk}}{l_{mk}}\right)^{\frac{l_{pk}}{l_e}}\right] \omega_k(T_f) \frac{1}{\frac{d_{mk}}{K^k_{C^0}} + \mu r} r - (d_k + \mu)p_k\,.
\end{aligned} \tag{A.16}$$

We now define the ribosomes density related term:

$$E_{mk}(l_{pk}, l_e) \triangleq \frac{l_{pk}}{l_e}\left[1 - \left(\frac{l_{pk}}{l_{mk}}\right)^{\frac{l_{pk}}{l_e}}\right] = \frac{l_{pk}}{l_e}\left[1 - \left(1 - \frac{l_e}{l_{mk}}\right)^{\left(\frac{l_{mk}}{l_e} - 1\right)}\right] . \quad \text{(A.17)}$$

Figure A.2 shows the values of $E_{mk}(l_{pk}, l_e)$ as a function of the protein length l_{pk} for different values of l_e. As seen, $E_{mk}(l_{pk}, l_e)$ can be accurately approximated as the linear function of l_{pk}/l_e:

$$E_{mk}(l_{pk}, l_e) \approx 0.62\frac{l_{pk}}{l_e} . \quad \text{(A.18)}$$

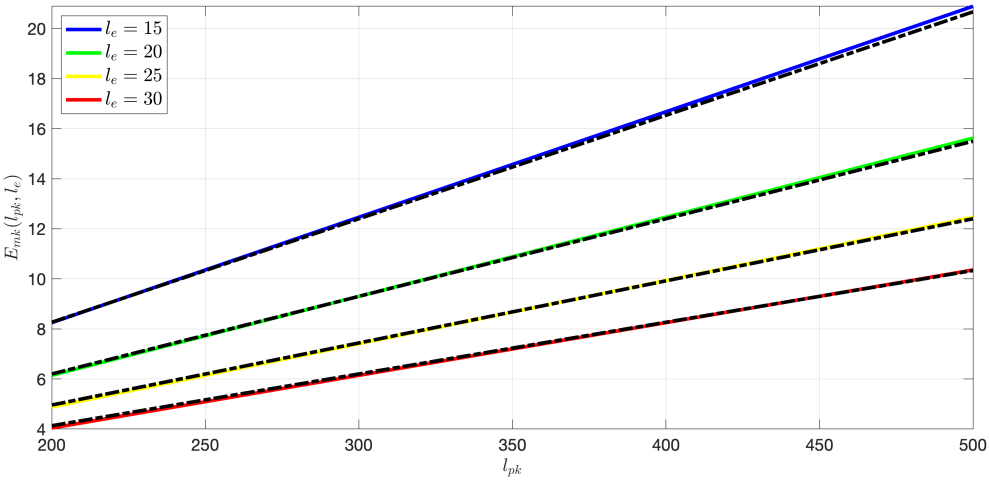

Figure A.2: Function $E_{mk}(l_{pk}, l_e)$ as a function of the protein length l_{pk} for different values of l_e and their corresponding linear approximations $E_{mk}(l_{pk}, l_e) = 0.62\frac{l_{pk}}{l_e}$.

We further define the *resources recruitment strength* functional parameter $J_k(\mu, r)$:

$$J_k(\mu, r) \triangleq E_{mk}(l_{pk}, l_e)\omega_k(T_f)\frac{1}{\frac{d_{mk}}{K^k_{C^0}} + \mu r} . \quad \text{(A.19)}$$

Notice the *resources recruitment strength* is a dimensionless function that expresses the capacity of the k-th protein-coding gene to engage cellular resources to get expressed. It explicitly depends on:

- the gene expression characteristics:

  - mRNA transcription rate $\omega_k(T_f)$ and degradation rate constant d_{mk}
  - RBS strength-related parameter $K^k_{C^0}(s_i)$

- and the availability of cell resources:
  - flux of free ribosomes μr
  - ribosomes density $\frac{l_{pk}}{l_e}$ (via $E_{mk}(l_{pk}, l_e)$)

Using these definitions in equation (A.16) we get the expressions for the abundance dynamics of the k-th protein:

$$\begin{aligned}\dot{p}_k &= 0.62\frac{\nu_t(s_i)}{l_e}\frac{r}{\frac{d_{mk}}{K^k_{C^0}(s_i)} + \mu r}\omega_k(T_f) - (d_k + \mu)p_k\,, \\ &= \frac{\nu_t(s_i)}{l_{pk}}J_k(\mu, r)r - (d_k + \mu)p_k\,.\end{aligned} \tag{A.20}$$

A.2 Dynamics of the number of mature available ribosomes

Next, using the results of the previous section, we obtain the dynamics of the number of mature available ribosomes. Ribosomes are large complexes formed by both ribosomal RNA molecules and a variety of ribosomal proteins, adding up to 55 different protein species in *E. coli*. Recall we consider that translation is the main energy and resources limiting process we model. Notice the total number of ribosomes in the cell at any one time instant r_T is the sum of the mature (r_a) and immature (r_i) ribosomes. The mature ribosomes r_a available for protein translation comprise the free ribosomes r and the ones bound to translating complexes. The bound ribosomes comprise the ones bound to translating complexes building the ribosomes themselves (r^r_b), the ones bound to the translating complexes of endogenous non-ribosomal proteins (r^{nr}_b) and the ones associated to the expression of exogenous genes (r^{exo}_b). In turn, the bound ribosomes may be either bound to the RBSs or actively elongating along the transcripts to synthesize proteins. We denote the last subset as r^r_t, r^{nr}_t and r^{exo}_t respectively. We also denote $r^h_b = r^r_b + r^{nr}_b$ as the set of *host* ribosomes bound both to ribosomal and non-ribosomal endogenous complexes. In case we are interested in a particular *strain* hosting exogenous genes, we refer to the set of all bound ribosomes as $r^s_b = r^h_b + r^{\text{exo}}_b$. In summary, we have:

$$\begin{aligned} r_T &= r_a + r_i \,, \\ r_a &= r + r_b^s \,, \\ r_b^s &= r_b^h + r_b^{\text{exo}} \,, \\ r_b^h &= r_b^r + r_b^{nr} \,. \end{aligned} \tag{A.21}$$

An analogous notation is used for the actively elongating bound ribosomes, with $r_t^h = r_t^r, + r_t^{nr}$ and $r_t^s = r_t^h + r_t^{\text{exo}}$.

The number of available mature ribosomes is a fraction of the total number of ribosomes, so that $r_a = \Phi_m r_T$. The fraction of active elongating ribosomes varies little in time, with an average value 0.8 [14, 15]. Therefore, we can expect that the fraction Φ_m also varies little, so that the dynamics of the total number of ribosomes and that of the number of available ribosomes are the same but for a scale factor. Next we consider the dynamics of the total copy number of ribosomes in the cell, r_T.

To get the dynamics of r_T we first consider an analogous expression to (A.20) for each of the proteins forming up a ribosome. If we consider the average ribosomal protein p_r, and an average ribosome composed of N_r proteins (e.g. $N_r = 55$ for *E. coli*) we define the total number of ribosomal proteins as $p_{\Sigma r} = N_r p_r$. For the average ribosomal protein p_r we have use (A.20) to obtain dynamics:

$$\begin{aligned} \dot{p}_r &= \frac{\nu}{l_{p_r}} \frac{s_i}{K_{sc} + s_i} E_{m,r} \omega_r \frac{1}{\frac{d_{mr}}{K_{C^0}^r} + \mu r} r - \mu p_r \,, \\ &= \frac{\nu}{l_{p_r}} \frac{s_i}{K_{sc} + s_i} J_r(\mu, r) r - \mu p_r \,, \end{aligned} \tag{A.22}$$

where we have assumed that ribosomal proteins are only subject to dilution caused by cell growth.

Then, the dynamics for the total number of ribosomal proteins can be approximated as:

$$\begin{aligned} \dot{p}_{\Sigma r} &= N_r \dot{p}_r \,, \\ &= \frac{\nu}{l_{p_r}} \frac{s_i}{K_{sc} + s_i} N_r J_r(\mu, r) r - \mu p_{\Sigma r} \,. \end{aligned} \tag{A.23}$$

Since all N_r protein species are needed to form up an individual ribosome, and considering the average ribosomal protein p_r, the total number of ribosomes is $r_T = p_{\Sigma r}/N_r$. Therefore, the dynamics of the total abundance of ribosomes r_T will be the same as those of p_r. That is:

$$\begin{aligned}\dot{r}_T &= \frac{\nu}{l_{p_r}} \frac{s_i}{K_{sc} + s_i} E_{m,r} \omega_r \frac{1}{\frac{d_{mr}}{K_{C^0}^r} + \mu r} r - \mu r_T \,, \\ &= \frac{\nu}{l_{p_r}} \frac{s_i}{K_{sc} + s_i} J_r(\mu, r) r - \mu r_T \,.\end{aligned} \tag{A.24}$$

Therefore, the dynamics of the number of mature available ribosomes is:

$$\dot{r}_a = \frac{\nu}{l_{p_r}} \frac{s_i}{K_{sc} + s_i} J_r(\mu, r) \Phi_m r - \mu r_a \,. \tag{A.25}$$

A.3 Relating free and available ribosomes

The number of free ribosomes is a measure of the cell burden, and plays a central role in the synthesis rate of proteins. Next we relate the number of free ribosomes with that of the synthesized available ones. That is, the available mature ones in the cell. This relationship will later allow to express the dynamics of the total number (alternatively, mass) of cell ribosomes and non-ribosomal proteins as a function of their interaction

Recall we had $r_a = r + r_b^s$. For each protein p_k, and using the previous results, the number of ribosomes bound to complexes involved in its translation at each time instant is given by:

$$\begin{aligned} r_b^{p_k} = C_{k^0} + \sum_{j=1}^{n_r} C_{k^j} &= \left[1 + \frac{a}{1-a}(1 - a^{n_r})\right] C_{k^0} \,, \\ &= \left[1 + E_{mk}(l_{pk}, l_e)\right] C_{k^0} \,, \\ &= \left[1 + E_{mk}(l_{pk}, l_e)\right] \frac{1}{E_{mk}(l_{pk}, l_e)} J_k(\mu, r) r \,, \\ &= \left[1 + \frac{1}{E_{mk}(l_{pk}, l_e)}\right] J_k(\mu, r) r \,, \end{aligned} \tag{A.26}$$

where notice that C_{k^0} ribosomes are bound to the RBSs of the k-th protein-coding gene transcripts and $r_t^{p_k} = E_{mk}(l_{pk}, l_e)C_{k^0} = J_k(\mu, r)r$ are actively involved in translation.

An analogous expression can be obtained for the number r_r of ribosomes bound to complexes involved in translation of ribosomes themselves:

$$r_b^r = N_r \left(C_{r^0} + \sum_{j=1}^{n_r} C_{r^j} \right) = N_r \left[1 + \frac{1}{E_{mr}(l_{p_r}, l_e)} \right] J_r(\mu, r)r \,, \tag{A.27}$$

where we have taken into account that it requires N_r protein species to build-up a ribosome and we consider an average ribosomal protein.

Therefore, the number of mature available ribosomes r_a can be obtained from:

$$\begin{aligned} r_a &= r + N_r \left(C_{r^0} + \sum_{j=1}^{n_r} C_{r^j} \right) + \sum_{k=1}^{N_{nr}} \left[C_{k^0} + \sum_{j=1}^{n_r} C_{k^j} \right] + \sum_{k=1}^{N_{\text{exo}}} \left[C_{k^0} + \sum_{j=1}^{n_r} C_{k^j} \right], \\ &= r + N_r \left[1 + \frac{1}{E_{mr}(l_{p_r}, l_e)} \right] J_r(\mu, r)r + \sum_{k=1}^{N_{nr}+N_{\text{exo}}} \left[1 + \frac{1}{E_{mk}(l_{pk}, l_e)} \right] J_k(\mu, r)r \,, \end{aligned} \tag{A.28}$$

where N_{nr} is the number endogenous non-ribosomal host protein-coding genes and N_{exo} that of exogenous ones.

Notice for a ribosomal density $l_e = 25$ and average ribosomal and non-ribosomal protein lengths (see Table A.2) the average values $1/E_{mr} = 0.21$ and $1/E_{mp} = 0.12$ are small, accounting for the percentage of ribosomes bound to the RBS.

From equation (A.28) we get the number of free ribosomes r as a function of the mature available ones r_a:

$$r = \frac{r_a}{1 + N_r \left[1 + \frac{1}{E_{mr}(l_{p_r}, l_e)} \right] J_r(\mu, r) + \sum_{k=1}^{N_{nr}+N_{\text{exo}}} \left[1 + \frac{1}{E_{mk}(l_{pk}, l_e)} \right] J_k(\mu, r)} \,. \tag{A.29}$$

A.4 Fractions of bound and actively translating ribosomes with respect to the available mature ones

Notice from (A.26) and (A.27) that the number of bound and bound-actively-translating ribosomes is:

$$\begin{aligned}
r_b^r &= N_r \left[1 + \frac{1}{E_{mr}(l_{p_r}, l_e)}\right] J_r(\mu, r) r\,, \quad r_t^r = N_r J_r(\mu, r) r\,, \\
r_b^{nr} &= \sum_{k=1}^{N_{nr}} \left[1 + \frac{1}{E_{mk}(l_{pk}, l_e)}\right] J_k(\mu, r) r\,, \quad r_t^{nr} = \sum_{k=1}^{N_{nr}} J_k(\mu, r) r\,, \\
r_b^{\text{exo}} &= \sum_{k=1}^{N_{\text{exo}}} \left[1 + \frac{1}{E_{mk}(l_{pk}, l_e)}\right] J_k(\mu, r) r\,, \quad r_t^{\text{exo}} = \sum_{k=1}^{N_{\text{exo}}} J_k(\mu, r) r\,.
\end{aligned} \tag{A.30}$$

Using (A.29) and:

$$\begin{aligned}
\text{WSum}(\mu, r) \triangleq\; & N_r \left[1 + \frac{1}{E_{mr}(l_{p_r}, l_e)}\right] J_r(\mu, r) \\
& + \sum_{k=1}^{N_{nr}+N_{\text{exo}}} \left[1 + \frac{1}{E_{mk}(l_{pk}, l_e)}\right] J_k(\mu, r)\,,
\end{aligned} \tag{A.31}$$

we can define the fractions of ribosomes bound to complexes and those actively involved in translation relative to the mature available ones for ribosomal, non-ribosomal and exogenous protein-coding genes:

$$
\begin{aligned}
\Phi_t^r &\triangleq \frac{N_r J_r(\mu, r)}{1 + \mathrm{WSum}(\mu, r)} = \frac{r_t^r}{r_a}\,, \\
\Phi_b^r &\triangleq \frac{N_r \left[1 + \frac{1}{E_{mr}(l_{pr}, l_e)}\right] J_r(\mu, r)}{1 + \mathrm{WSum}(\mu, r)} = \frac{r_b^r}{r_a}\,, \\
\Phi_t^{nr} &\triangleq \frac{\sum_{j=N_{nr}} J_j(\mu, r)}{1 + \mathrm{WSum}(\mu, r)} = \frac{r_t^{nr}}{r_a}\,, \\
\Phi_b^{nr} &\triangleq \frac{\sum_{k=N_{nr}} \left[1 + \frac{1}{E_{mk}(l_{pk}, l_e)}\right] J_k(\mu, r)}{1 + \mathrm{WSum}(\mu, r)} = \frac{r_b^{nr}}{r_a}\,, \\
\Phi_t^{exo} &\triangleq \frac{\sum_{j=\mathrm{exo}} J_j(\mu, r)}{1 + \mathrm{WSum}(\mu, r)} = \frac{r_t^{nr}}{r_a}\,, \\
\Phi_b^{exo} &\triangleq \frac{\sum_{k=N_{\mathrm{exo}}} \left[1 + \frac{1}{E_{mk}(l_{pk}, l_e)}\right] J_k(\mu, r)}{1 + \mathrm{WSum}(\mu, r)} = \frac{r_b^{nr}}{r_a}\,,
\end{aligned}
\tag{A.32}
$$

where $\sum_{k=N_{nr}}, \sum_{k=N_{\mathrm{exo}}}$ stand for the sum over the ensemble of all genes coding for endogenous and exogenous non-ribosomal proteins respectively.

Thus, for the native host cell we can consider:

$$
\Phi_t^h \triangleq \Phi_t^r + \Phi_t^{nr}\,, \quad \Phi_b^h \triangleq \Phi_b^r + \Phi_b^{nr}\,. \tag{A.33}
$$

and the analogous $\Phi_t^s = \Phi_t^h + \Phi_t^{exo}$ and $\Phi_b^s = \Phi_b^h + \Phi_b^{exo}$ for the strain hosting exogenous protein-coding genes.

Using (A.31) and (A.32), the fraction of ribosomes bound to translating complexes (including both translating complexes for endogenous and exogenous proteins and those bound to RBSs) relative to the available mature ribosomes is:

$$
\Phi_b = \frac{\mathrm{WSum}(\mu, r)}{1 + \mathrm{WSum}(\mu, r)}\,. \tag{A.34}
$$

Notice each term

$$
\frac{\left[1 + \frac{1}{E_{mj}}\right] J_j(\mu, r)}{1 + \sum_{k=\{r,nr,exo\}} \left[1 + \frac{1}{E_{mk}}\right] J_k(\mu, r)} \tag{A.35}
$$

can be understood as the share of cell resources required to express the $j-$th protein-coding gene. Thus, the magnitude of the dimensionless coefficient $J_j(\mu, r)$ is a measure of the resources recruited to express the $j-$th protein-coding gene.

A.5 Obtaining the cell specific growth rate

Cell growth can essentially be explained as the time variation of the protein fraction of the total cell mass. Yet, not all protein mass contributes to cell growth. On the one hand, there are proteins which may be undergoing active degradation. On the other, exogenous proteins will in general not have any active role in the cell that contributes to its growth. Therefore, next we consider only the endogenous ribosomal and non-ribosomal proteins to obtain the cell specific growth rate from the time variation of the endogenous protein fraction of the total cell mass.

To deal with protein degradation, we take into account that the protein fraction of cell mass is the sum of the mass of functional and non-functional proteins (ie. proteins undergoing degradation). Though non-functional proteins do not contribute to cell growth, they do to cell mass. Thus, for a k-th protein species we can consider the fraction quantity of functional molecules of the protein, p_k, and the one of non-functional ones p_k^{nf} so that the total number of protein copies of the k-th species is $p_k^T = p_k + p_k^{nf}$. Then, considering the dynamics (A.20) of a generic protein, we have:

$$\begin{aligned} \dot{p}_k &= \frac{\nu_t(s_i)}{l_{pk}} J_k(\mu, r) r - (d_k + \mu) p_k \,, \\ \dot{p}_k^{nf} &= d_k p_k - \mu p_k^{nf} \,, \end{aligned} \tag{A.36}$$

where we have taken into account that the non-functional fraction p_k^{nf} only undergoes dilution due to cell growth and there is a conversion from functional to non-functional fraction caused by protein degradation.

If we consider the average mass of an amino acid m_{aa}, the mass weight of a protein of length l_{pk} can be approximated as $m_{aa} l_{pk}$. Thus, for p_k^T molecules of the k-th protein, their total mass weight is $m_k = m_{aa} l_{pk} p_k^T$. Then, the mass of the N_{nr} non-ribosomal endogenous proteins in the cell proteins can be approximated as:

$$m_{nr} = \sum_{k=1}^{N_{nr}} m_{aa} l_{pk} \left(p_k + p_k^{nf} \right) \,. \tag{A.37}$$

Therefore, the times variation of the protein mass explained by this set of N_{nr} proteins is:

$$\begin{aligned}\dot{m}_{nr} &= \sum_{k=1}^{N_{nr}} m_{aa} l_{pk} \left(\dot{p}_k + \dot{p}_k^{nf}\right) , \\ &= m_{aa} \sum_{k=1}^{N_{nr}} l_{pk} \left[\frac{\nu_t(s_i)}{l_{pk}} J_k(\mu, r) r - (d_k + \mu) p_k\right] + m_{aa} \sum_{k=1}^{N_{nr}} l_{pk} \left(d_k p_k - \mu p_k^{nf}\right) , \\ &= m_{aa} \frac{\nu s_i}{K_{sc} + s_i} \sum_{k=1}^{N_{nr}} J_k(\mu, r) r - \mu m_{nr} , \\ &= m_{aa} \nu_t(s_i) \Phi_t^{nr} r_a - \mu m_{nr} ,\end{aligned} \tag{A.38}$$

where recall $J_k(\mu, r)r$ is the number of ribosomes bound to complexes involved in the translation of each k-th protein and we have used the definitions (A.32) in the last step. Notice, in addition, that the degradation of proteins does not play a role when we consider the dynamics of the protein mass. It indeed plays a role when we consider the number of active proteins.

As for the protein cell mass variation explained by the time variation in the total number of ribosomes r_T we will have the analogous expression:

$$\dot{m}_{r_T} = m_{aa} \nu_t(s_i) \Phi_t^r r_a - \mu m_{r_T} , \tag{A.39}$$

where we have used the fact that N_r ribosomal proteins are required to form up one ribosome. Notice that we only consider the weight of the protein fraction of the ribosomes. This accounts only for approximately one third of the ribosomes mass [77].

Denoting the *host* cell protein weight $m_h = m_{nr} + m_{r_T}$, we reach the expression:

$$\dot{m}_h = m_{aa} \nu_t(s_i) \left(\Phi_t^r + \Phi_t^{nr}\right) r_a - \mu m_h , \tag{A.40}$$

which, using (A.33), can be expressed as:

$$\dot{m}_h = m_{aa} \nu_t(s_i) \Phi_t^h r_a - \mu m_h . \tag{A.41}$$

Recall that the specific growth rate μ is a continuous approximation of the discrete event process of cell duplication. Here we consider that the total biomass dry weight variation (ie. that of the whole population of cells) is mainly caused by cell duplication (i.e. population growth), and the dynamics of cell mass accumulation are much faster than those of cell duplication. Under this assumption, we may consider the protein mass for each cell quickly reaches steady state ($\dot{m}_h \approx 0$). Thus, from equation (A.41) we get the expression for the cell specific growth rate:

$$\mu = \frac{m_{aa}}{m_h}\nu_t(s_i)\Phi_t^h r_a \,. \tag{A.42}$$

Notice $\Phi_t^h r_a$ is the number of ribosomes actively translating endogenous proteins (both ribosomal and non-ribosomal) at a given time instant. Equation (A.42) allows to predict this number given a specific growth rate, assuming saturation of intracellular substrate (eg. considering a batch experiment and the exponential growth phase) and considering the average values for the amino acids mass and that of the protein fraction of the cell.

Notice also that (A.42) can be expressed as a function of the total number of ribosomes r_T as:

$$\mu = \frac{m_{aa}}{m_h}\nu_t(s_i)\Phi_t^h\Phi_m r_T \,. \tag{A.43}$$

A.6 Cell specific growth rate and population dynamics

Our model considers the intracellular substrate s_i as source of building blocks to synthesize proteins. Using the model in a multi-scale framework considering the macroscopic dynamics of the population of cells and the uptake of extracellular substrate requires relating the cell population growth rate with the individual cell one. To this end, we next relate the expression for the cell specific growth rate μ obtained in (A.42) with the classical Monod-like expressions for the growth of a population of cells.

Recall if we have a population of N cells and we consider the average cell dry mass m_{cDW}, the total biomass dry-weight will be $M_b = Nm_{cDW}$. By taking derivative with respect to time, we get:

$$\dot{M}_b = \dot{N}m_{cDW} + N\dot{m}_{cDW} \,. \tag{A.44}$$

Now, consider the continuous approximation of cell duplication:

$$\dot{N} = \mu N \,, \tag{A.45}$$

where $\mu = \log 2/t_d$, with t_d the cell population doubling time, is the specific growth rate. Then:

$$\dot{M}_b = \mu M_b + N\dot{m}_{cDW} \,, \tag{A.46}$$

from which we get:

$$\dot{m}_{cDW} = \frac{\dot{M}_b}{N} - \mu m_{cDW} \,, \tag{A.47}$$

where $\frac{\dot{M}_b}{N}$ is the mean value per cell of the population mass growth.

As done in section A.5, we assume the cell mass quickly reaches steady state as compared to the population dynamics. Thus, from equation (A.47) and assuming $\dot{m}_{cDW} \approx 0$ we get:

$$\mu = \frac{\dot{M}_b}{N m_{cDW}} = \frac{\dot{M}_b}{M_b} \,. \tag{A.48}$$

Experimental evidence suggests that the cell densityρ varies little throughout the adult cell life [56], so:

$$\mu = \frac{\dot{M}_b}{M_b} = \frac{\rho\dot{M}_b}{\rho M_b} = \frac{\dot{V}_b}{V_b} \,. \tag{A.49}$$

The specific growth rate under a limiting substrate obtained from the experimental macroscopic analysis of a culture of cells can be expressed using the classical empirical Monod relationship:

$$\mu(s) = \frac{\dot{M}_b}{M_b} = \frac{\mu_m s}{K_s + s} \,, \tag{A.50}$$

where μ_m is the maximum specific growth rate, K_s is the Monod affinity constant, and s is the concentration of the limiting substrate in the culture medium.

The identities (A.49) allow to relate the specific growth rate obtained in (A.42) with the one obtained from population-scale macroscopic experiments under the condition of steady-state growth. Under this condition, the rate of total cell-mass growth is identical to the rate of cell number growth [128]. Thus:

$$\mu = \frac{\mu_m s}{K_s + s} = \frac{m_{aa}}{m_h} \frac{\nu s_i}{K_{sc} + s_i} \Phi_t^h r_a \,. \tag{A.51}$$

To relate the intracellular and extracellular substrate, we resorted to the theoretical approaches to derive the Monod equation. Several alternatives exist [61, 127, 119]. We followed a reasoning derived from the model developed in [125], where the quantity of intracellular substrate s_i is related to the one of extracellular substrate s through the dynamics of nutrient import and catabolism:

$$\dot{s}_i = e_t \frac{\nu_t s}{\frac{k_t}{V_m} + s} - e_m \frac{\nu_m s_i}{k_m + s_i} - \mu s_i \,, \tag{A.52}$$

where V_m is the harvest volume [119], e_t and e_m are transport and catabolism enzymes, and Michaelis-Menten kinetics are assumed (see [125]).

If we assume that nutrient import quickly balances nutrient catabolism and we neglect the dilution term:

$$\dot{s}_i \approx 0 \rightsquigarrow s_i = \frac{\frac{k_m}{c-1} s}{\frac{k_t}{V_m} \frac{c}{c-1} + s} \,, \tag{A.53}$$

where $c = e_m \nu_m / (e_t \nu_t)$. Notice if the maximum import and catabolism fluxes are balanced, ie. $c \approx 1$, then there is a linear relationship between the intracellular amount of substrate and its extracellular concentration. Otherwise, if catabolism is more efficient than transport ($c > 1$) the intracellular amount of substrate s_i saturates with increasing values of s.

Recall in our model the specific growth rate (A.51) is a function of s_i. Using (A.53) we obtain the Monod-like expression:

$$\frac{s_i}{K_{sc} + s_i} = \frac{\frac{k_m}{K_{sc}(c-1) + k_m} s}{\frac{K_{sc} k_t c}{V_m [K_{sc}(c-1) + k_m]} + s} \,. \tag{A.54}$$

If we assume that the Michaelis-Menten constant for substrate catabolism is the same as the constant we defined in (A.3), that is, $K_{sc} = k_m$, we have:

$$\frac{s_i}{K_{sc} + s_i} = \frac{\frac{1}{c}s}{\frac{k_t}{V_m} + s}\,. \tag{A.55}$$

Notice that the hypothesis $K_{sc} = k_m$ implicitly implies that the Michaelis-Menten constants for substrate catabolism and transport have similar values ($k_t = k_m$), in agreement with the assumptions in [125].

Also notice the term $1/c = e_t\nu_t/(e_m\nu_m)$ can be interpreted as the maximum flux yield between nutrient import and its catabolism. If $c \approx 1$, that is, under the hypothesis that the efficiency of nutrient import and catabolism are balanced so the maximum import and catabolism fluxes are similar:

$$\frac{s_i}{K_{sc} + s_i} = \frac{s}{\frac{k_t}{V_m} + s}\,. \tag{A.56}$$

In case catabolism is more efficient than transport ($c > 1$) we will need an increase in the concentration of the substrate in the extracellular medium (s) to achieve the same value of $s_i/(K_{sc}+s_i)$ in (A.55) as compared to the balanced case $c = 1$. Finally, in case transport is more efficient than catabolism ($c < 1$) we will need lower concentrations of the extracellular substrate.

Then, from (A.51), (A.50) and (A.55) we get:

$$\mu(s) = \frac{\frac{1}{c}\frac{m_{aa}}{m_h}\nu\Phi_t^h r_a s}{\frac{k_t}{V_m} + s} = \frac{\mu_m s}{K_s + s}\,, \tag{A.57}$$

so we can identify:

$$\begin{aligned} \mu_m &= \frac{1}{c}\frac{m_{aa}}{m_h}\nu\Phi_t^h r_a\,, \\ K_s &= \frac{k_t}{V_m}\,. \end{aligned} \tag{A.58}$$

Note: The relationship (A.55) is valid but in the extreme cases non relevant cases $c = 0$ (there is transport into the cell but nutrients are not metabolised)

and $c = \infty$. In these cases dilution cannot be neglected in equation (A.52) and the equilibria are different.

A.7 Relationship between growth rate and cell mass

Our model accounts for the protein mass distribution but does not obtain the total protein cell mass. A simple approach to solve this would be considering a constant average cell protein mass. Yet, the cell mass varies with growth rate. In this section we consider the relationship between growth rate and the total cell protein mass, and obtain an empirical model relating the host protein mass m_h with the specific growth rate μ.

Several phenomenological models have been proposed in the literature accounting for the relationship between growth rate and cell dry weight, like the recent ones [108, 128]. In [128] a Monod-like relation between the chromosome replication-segregation period $C + D$ and the cell specific growth rate μ is considered:

$$C + D = \frac{\alpha + \beta\mu}{\mu}\,. \tag{A.59}$$

We estimated the parameters α and β in (A.59) using the data in [15]. Figure A.3(left) shows the good fit obtained.

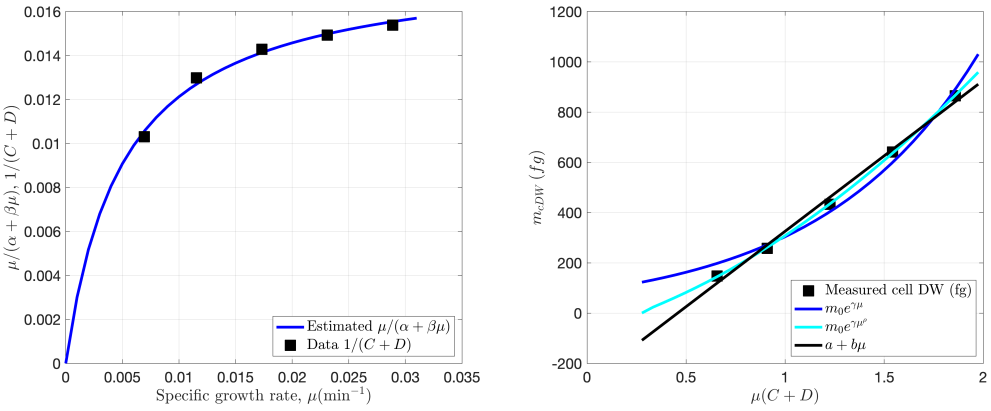

Figure A.3: (Left:) Best fitting results for the identity (A.59) using the data in [15]. (Right:) Alternative fittings between the cell dry weight m_c and the product $\mu\,(C + D)$.

In addition the authors in [128] propose a linear relation between the cell dry weight and the product of $C + D$ and μ:

$$m_c = m_0\mu(C + D)\,. \tag{A.60}$$

Therefore, according to this model, there is an affine relationship between the cell dry weight and specific growth rate:

$$m_c = m_0(\alpha + \beta\mu)\,. \tag{A.61}$$

As shown in Figure A.3(right), for the data we used, the relationship (A.60) gives a very rough approximation. Indeed, as shown in the same figure, and affine relationship gives much better fit.

As an alternative, in [108] an exponential relationship is proposed between the cell volume S_c and the product of $\mu(C + D)$:

$$S_c = S_0 e^{\mu(C+D)}\,. \tag{A.62}$$

Assuming constant cell density, equation (A.62) can be expressed as a function of the cell dry weight. Figure A.3(right) shows the fit assuming constant cell density, so that equation (A.62) can be expressed as a function of the cell dry weight. Notice that a better fit can be obtained considering the modified relationship:

$$m_c = m_0 e^{\gamma\mu(C+D)}\,. \tag{A.63}$$

Now, if we consider expressions (A.59) and (A.63), we get:

$$m_c = m_0 e^{\gamma(\alpha+\beta\mu)} = \bar{m}_0 e^{\bar{\gamma}\mu}\,, \tag{A.64}$$

where $\bar{m}_0 = m_0 e^{\gamma\alpha}$ and $\bar{\gamma} = \gamma\beta$.

Notice equation (A.64) is the solution of the differential equation:

$$\frac{dm_c}{d\mu} = \bar{\gamma} m_c\,, \tag{A.65}$$

which expresses that the variation of cell mass (dry weight) relative to the variation of the specific growth rate is proportional to the cell mass. Figure A.4 (left)

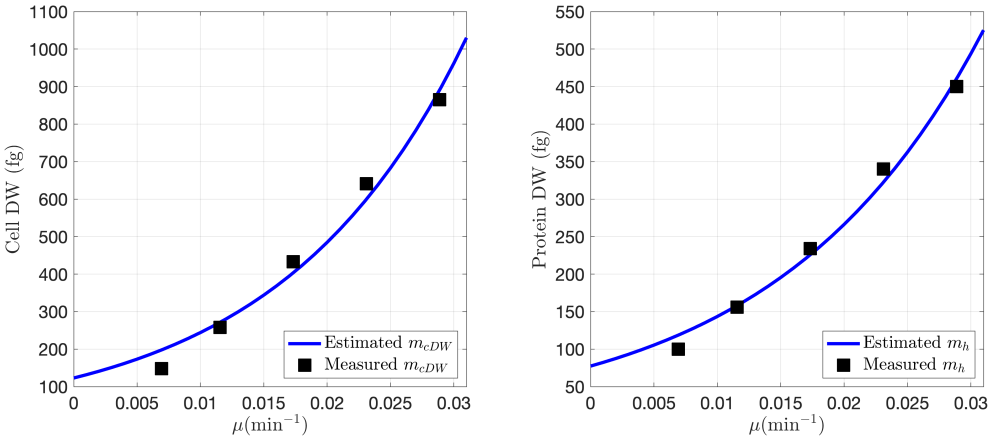

Figure A.4: Fit between the cell dry weight (left) and the protein content dry weight (right) as a function of the specific growth rate using the expressions (A.65) and (A.66) respectively.

shows the results obtained for (A.64) and Table A.1 lists the best fitted parameters. Better fits can be obtained with alternative phenomenological expressions to (A.64) at the cost of losing the simple interpretation provided by expression (A.65).

For the relationship between the cell content of endogenous proteins and the specific growth rate we postulate a relationship analogous to (A.65). Thus, we consider:

$$\frac{dm_h}{d\mu} = \beta m_h \,. \tag{A.66}$$

Figure A.4 (right) shows the results obtained and Table A.1 lists the parameters corresponding to the best fit.

Table A.1: Phenomenological models relating cell and protein dry mass and specific growth rate and their best fit parameters for the data in [15].

Model	**Model parameters**	
$m_c = \bar{m}_{c0} e^{\bar{\gamma}\mu}$	$\bar{m}_{c0} = 123.022 \times 10^{-15}\,\text{g}$	$\bar{\gamma} = 68.554\,\text{min}$
$m_h = \bar{m}_{h0} e^{\beta\mu}$	$\bar{m}_{h0} = 77.374 \times 10^{-15}\,\text{g}$	$\beta = 61.781\,\text{min}$

Notice that alternative phenomenological relationships can be used instead, with possibly better fit to the experimental data. Thus, for instance, the power law model $m_h = \bar{m}_0 e^{\bar{\gamma}\mu^\nu}$ with $\bar{m}_0 = 1.2918\,10^{-14}$, $\bar{\gamma} = 14.1089$ and $\nu = 0.389$ is an alternative.

A.8 Average host dynamics and steady state balanced growth.

We were interested in having an average model for the host dynamics and its steady state that can be used as base for analyzing host-circuit interactions. To this end we considered, on the one hand, the dynamics (A.39) of the ribosomal protein mass content of host, m_{r_T} and, on the other, the dynamics (A.38) of the mass of the ensemble of non-ribosomal endogenous proteins m_{nr}.

For the non-ribosomal endogenous proteins we consider them as a lumped species with a single average resources recruitment strength $J_{nr}(\mu, r)$ and average E_{mnr} such that:

$$N_{nr}\left[1 + \frac{1}{E_{mnr}}\right] J_{nr}(\mu, r) = \sum_{k=1}^{N_{nr}} \left[1 + \frac{1}{E_{mk}(l_{pk}, l_e)}\right] J_k(\mu, r)\,. \tag{A.67}$$

We also considered average values for the endogenous protein lengths l_p^r and l_p^{nr} and a common average amino acid mass m_{aa} so that the masses are related to the number of proteins as:

$$m_{r_T} = m_{aa} N_r l_p^r r_T = m_{rib} r_T\,, \tag{A.68a}$$
$$m_{nr} = m_{aa} N_{nr} l_p^{nr} p_{nr}\,, \tag{A.68b}$$

where m_{rib} is the average mass of the protein content of ribosomes and p_{nr} includes both functional and inactive endogenous proteins (see A.5).

Then, using (A.39), (A.38), the expression for the growth (A.42) and the definitions (A.32), the dynamics for the host endogenous ribosomal and non-ribosomal protein mass can be expressed as:

$$\dot{m}_{r_T} = \mu \left[m_h(\mu) \frac{N_r J_r(\mu, r)}{N_r J_r(\mu, r) + N_{nr} J_{nr}(\mu, r)} - m_{r_T} \right], \tag{A.69a}$$

$$\dot{m}_{nr} = \mu \left[m_h(\mu) \frac{N_{nr} J_{nr}(\mu, r)}{N_r J_r(\mu, r) + N_{nr} J_{nr}(\mu, r)} - m_{nr} \right], \tag{A.69b}$$

$$J_r(\mu, r) = 0.62 \frac{l_p^r}{l_e} \omega_r \frac{1}{\frac{d_m^r}{K_{C^0}^r(s_i)} + \mu r}, \tag{A.69c}$$

$$J_{nr}(\mu, r) = 0.62 \frac{l_p^{nr}}{l_e} \omega_{nr} \frac{1}{\frac{d_m^{nr}}{K_{C^0}^{nr}(s_i)} + \mu r}, \tag{A.69d}$$

with the average effective RBS strengths

$$K_{C^0}^k(s_i) = \frac{K_b^k}{K_u^k + \frac{\nu_t(s_i)}{l_e}} \tag{A.70}$$

for $k = \{r, nr\}$.

The number of free ribosomes is obtained using the averaged (A.29):

$$r = \frac{\Phi_m r_T}{1 + \left(1 + \frac{1}{E_{mr}}\right) N_r J_r(\mu, r) + \left(1 + \frac{1}{E_{mnr}}\right) N_{nr} J_{nr}(\mu, r)} \tag{A.71}$$

with r_T obtained from (A.68a), and the specific growth rate is obtained from (A.42) using the fraction

$$\Phi_t^h = \frac{N_r J_r(\mu, r) + N_{nr} J_{nr}}{1 + \left(1 + \frac{1}{E_{mr}}\right) N_r J_r(\mu, r) + \left(1 + \frac{1}{E_{mnr}}\right) N_{nr} J_{nr}(\mu, r)}. \tag{A.72}$$

Notice from (A.69a)–(A.69d) that the steady state will reached either when the growth rate stalls ($\mu = 0$) or when there exists exponential balanced growth with:

$$\frac{m_{r_t}}{m_h(\mu)} = \frac{N_r J_r(\mu, r)}{N_r J_r(\mu, r) + N_{nr} J_{nr}(\mu, r)}\,, \tag{A.73a}$$

$$\frac{m_{nr}}{m_h(\mu)} = \frac{N_{nr} J_{nr}(\mu, r)}{N_r J_r(\mu, r) + N_{nr} J_{nr}(\mu, r)}\,, \tag{A.73b}$$

and the growth rate (A.43):

$$\mu = \frac{m_{aa}}{m_p} \nu_t(s_i) \Phi_t^h \Phi_m r_T\,.$$

Thus, at steady state balanced growth the relative resources recruitment strength provides the relative mass fractions at steady state. It is easy to see that this holds also for individual proteins. The relative resources recruitment strength of a given protein equals its relative mass in the cell.

Notice from (A.69a), (A.71) and (A.43) that at steady state:

$$\mu_{ss}(s_i) = \frac{m_{aa}}{m_{rib}} \nu_t(s_i) \Phi_m \Phi_t^r\,, \tag{A.74}$$

that is, the growth rate at steady state depends linearly on the fraction $\Phi_m \Phi_t^r$ of bound ribosomes actively being used to build up ribosomes (ie. actively translating the transcripts) relative to the total number of ribosomes.

At steady state, the flux of free resources for a given intracellular substrate s_i, defined as the number of free ribosomes times the cell growth rate, can be obtained as:

$$\mu_{ss} r_{ss}(s_i) = m_{aa} \nu_t(s_i) \left(\frac{\Phi_m \Phi_t^r}{m_{rib}} \right)^2 \frac{1 - \Phi_b^s}{\Phi_t^h} m_h(\mu_{ss}) \tag{A.75}$$

showing a linear relationship with the cell protein weight m_p.

The host cell protein weight m_h depends of the growth rate. To calculate it, we used both the data available in [15] and the phenomenological relationships obtained in section A.7.

A.9 Synthesis rate of exogenous proteins and interaction with the host cell

Product titer and productivity rate are important measures of performance in biotechnological applications. In this section we consider the expression of an exogenous protein-coding gene and we analyze how the number of produced proteins at steady state and its specific mass productivity rate (synthesis rate) vary as a function of its RBS and promoter strengths and the interaction with the host cell. The expressions obtained can easily be adapted for the analysis of the synthesis rate of an endogenous protein as a function of its RBS and promoter strengths and interaction with the remaining host endogenous genes.

We define the synthesis rate of a protein A, Π_{nA}, as the number molecules of A at steady-state balanced growth produced per cell and generation as in [59]. Notice this is equivalent to the productivity rate of the protein A. Thus, if we consider a population of N cells and the continuous approximation of cell duplication (A.45), the total quantity of molecules of the protein A, P_A^N, will increase as the population of cells does as:

$$\dot{P}_A^N = p_{A,\mathrm{ss}}\dot{N} = p_{A,\mathrm{ss}}\mu N\,, \tag{A.76}$$

where $p_{A,\mathrm{ss}}$ is the number of molecules of the protein A at steady state in a single cell. Therefore:

$$\Pi_{nA} = \frac{\dot{P}_A^N}{N} = p_{A,\mathrm{ss}}\mu\,. \tag{A.77}$$

Analogously, we can defined the mass synthesis rate as:

$$\Pi_A = m_{A,\mathrm{ss}}\mu\,. \tag{A.78}$$

For the host endogenous dynamics we considered the average model as described in section A.8. We extend the model by adding the dynamics of the exogenous protein of interest A as:

$$\dot{m}_A = \mu \left[m_h(\mu) \frac{N_A J_A(\mu, r)}{N_r J_r(\mu, r) + N_{nr} J_{nr}(\mu, r)} - m_A \right] , \tag{A.79a}$$

$$J_A(\mu, r) = 0.62 \frac{l_p^A}{l_e} \omega_A \frac{1}{\frac{d_m^A}{K_{C^0}^A} + \mu r} , \tag{A.79b}$$

where notice the denominator in the fraction of the resources recruitment strengths only includes the host protein-coding genes and the mass $m_h(\mu)$ is still that of the native host cell and not the mass of the strain $m_s = m_h(\mu) + m_A$.

The expressions (A.69a)–(A.69d) for the host endogenous dynamics remain the same. Yet, now the number of free ribosomes r is obtained using (A.29):

$$r = \frac{\Phi_m r_T}{1 + \mathrm{WSum}(\mu, r)} \tag{A.80}$$

with

$$\begin{aligned} \mathrm{WSum}(\mu, r) = &\left(1 + \frac{1}{E_{mr}}\right) N_r J_r(\mu, r) \\ &+ \left(1 + \frac{1}{E_{mnr}}\right) N_{nr} J_{nr}(\mu, r) \\ &+ \left(1 + \frac{1}{E_{mA}}\right) N_A J_A(\mu, r) \end{aligned} \tag{A.81}$$

and the specific growth rate is obtained from (A.42) using the fraction

$$\Phi_t^h = \frac{N_r J_r(\mu, r) + N_{nr} J_{nr}(\mu, r)}{1 + \mathrm{WSum}(\mu, r)} . \tag{A.82}$$

The exogenous proteins do not contribute to cell growth, but contribute to cell mass. Thus, the fraction of fraction ϕ_A of synthesized exogenous protein(s) must take into account their mass contribution. To this end, we take into account the specific growth rate is obtained from (A.42):

$$\mu = \frac{m_{aa}}{m_h}\nu_t(s_i)\Phi_t^h r_a$$

and the mass dynamics for the exogenous protein:

$$\dot{m}_A = m_{aa}\nu_t(s_i)\Phi_t^{nr} r_a - \mu m_A \,. \tag{A.83}$$

From (A.83) evaluated at steady state, we get

$$\mu = \frac{m_{aa}}{m_A}\nu_t(s_i)\Phi_t^A r_a \,. \tag{A.84}$$

Therefore, we obtain the relationship:

$$\frac{m_A}{m_h} = \frac{\Phi_t^A}{\Phi_t^h} = \frac{N_A J_A(\mu, r)}{N_r J_r(\mu, r) + N_{nr} J_{nr}(\mu, r)} \,. \tag{A.85}$$

Now, the protein mass content of the strain is

$$m_s(\mu) = m_h(\mu) + m_A(\mu) = \left(1 + \frac{\Phi_t^A}{\Phi_t^h}\right) m_h(\mu) = \frac{\Phi_t^s}{\Phi_t^h} m_h(\mu) \,. \tag{A.86}$$

Therefore, the mass fraction of protein A is:

$$\phi_A(\mu) \triangleq \frac{m_A(\mu)}{m_s(\mu)} = \frac{\Phi_t^A}{\Phi_t^s} = \frac{N_A J_A(\mu, r)}{N_r J_r(\mu, r) + N_{nr} J_{nr}(\mu, r) + N_A J_A(\mu, r)} \,. \tag{A.87}$$

Using (A.79a) evaluated at steady state and the results above, the synthesis rate (A.78) becomes:

$$\begin{aligned} \Pi_A &= m_{A,\mathrm{ss}}\mu \\ &= m_h(\mu)\frac{N_A J_A(\mu, r)}{N_r J_r(\mu, r) + N_{nr} J_{nr}(\mu, r)}\mu \\ &= m_s(\mu)\frac{N_A J_A(\mu, r)}{N_r J_r(\mu, r) + N_{nr} J_{nr}(\mu, r) + N_A J_A(\mu, r)}\mu \,. \end{aligned} \tag{A.88}$$

A.10 Model parameters

Table A.2 shows the set of parameters used in the model.

A.11 Estimation of the fractions Φ_t^h and Φ_b^h

Next, for the host native *E. coli* cell, we evaluate the fractions Φ_b^h and Φ_t^h of bound ribosomes and bound ribosomes being actively used in translating complexes relative to the mature available ones as a function of the number of free ribosomes r (see the definitions (A.33) in Section A.4). We expect a very low number of free ribosomes. If this was not the case, there would be no real competition to recruit them. To set an initial upper limit, we use the estimation $r = \mathbf{N}(350, 35)$ in [53]. In addition, having too many free ribosomes in excess would imply a superfluous us of energy for the cell. Considering this hypothesis, we evaluated equation (A.34) as a function of the mature available ribosomes r_a and the free ones r. Figure A.5(left) shows the values estimated. Notice the values of Φ_b^h close to $\Phi_b^h = 1$ indicating that the cell is always at the edge of its maximum capacity for using the available resources.

From the estimation of the fraction Φ_b^h and using the definitions in section A.4 we can obtain that of the dimensionless sum $\sum_{k=\{r,nr\}} \left[1 + \frac{1}{E_{mk}}\right] J_k(\mu, r)$ reflecting the resources recruitment burden generated by the whole set of ribosomal and non-ribosomal endogenous proteins being expressed at a given moment in the cell (see Figure A.5(right)).

To validate the estimations above, we evaluated both the sum $\sum_{k=\{r,nr\}} J_k(\mu, r)$ and the weighted sum $\sum_{k=\{r,nr\}} \left[1 + \frac{1}{E_{mk}}\right] J_k(\mu, r)$ for all protein-coding genes reported in [42] for *E. coli*.

The dynamic model for the expression of protein p in [42] considers:

$$\dot{p} = \frac{\beta_p \beta_m}{d_m} - \mu p \,, \tag{A.89}$$

where β_m (mRNA/t) is the transcription rate, β_p (protein/(mRNA·t)) the translation one and d_m the mRNA degradation rate constant. For those transcripts without information for d_m in [42] we used the value shown in Table A.2. Then, we used the equation (A.20) to derive the relationship:

Table A.2: Model parameters for *E. Coli*.

Parameter	Description	Value	Reference
t_d	duplication time	20–60 min	[77]
μ	specific growth rate	0.01–0.035 min^{-1}	$\mu = \log 2/t_d$
$y_{x/s}$	yield biomass on glucose	0.5 gDW $\cdot$ gGlucose^{-1}	
$\bar{l}_p$	average non-ribosomal protein length	333 aa $\cdot$ molec^-1	calculated from data in [42]
$\bar{l}_r$	average ribosomal protein length	195 aa $\cdot$ molec^{-1}	calculated from data in [42]
l_e	ribosome occupancy length	20–30 aa $\cdot$ molec^{-1}	estimated ([109, 32, 30, 82])
$l_{e,\text{min}}$	ribosome occlusion length	11 aa $\cdot$ molec^{-1}	[106]
ν	maximum translation speed per ribosome	1260 aa $\cdot$ min^{-1}	[77, 105]
k_u^k	dissociation rate constant RBS-ribosome	6–135 min^{-1}	[110, 53]
k_b^k	association rate constant RBS-ribosome	3–15 $\text{molec}^{-1} \cdot \text{min}^{-1}$	[110, 95, 53]
$t_{\frac{1}{2},\text{mRNA}}$	average mRNA half-life	2–6 min	[5, 20, 42, 40]
$d_{m,\text{ss}}$	median mRNA degradation rate constant (steady phase)	0.17 min^{-1}	[5]
d_m	median mRNA degradation rate constant (exponential growth)	0.28 min^{-1}	[20]
η	maximum transcription speed	1500–3000 aa $\cdot$ min^{-1}	[77, 64]
m_{aa}	average amino acid mass	110 Da	[114]
m_{aa}	average amino acid mass	182.6×10^{-24} g	[114]
m_p	geometric average of cell protein weight	224×10^{-15} gDW	[77, 15]

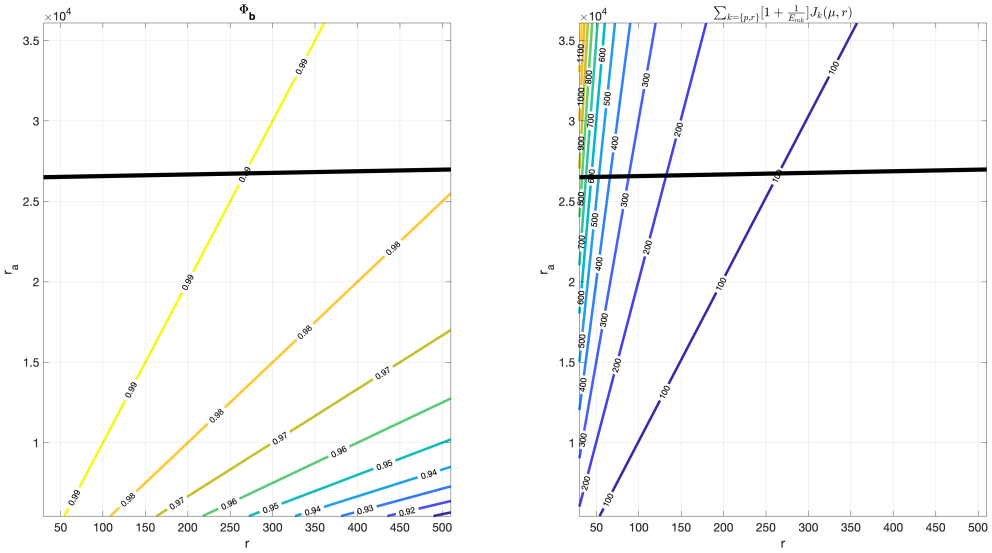

Figure A.5: (Left:) Fraction Φ_b^h of ribosomes bound to translating complexes relative to the mature available ones r_a as a function of r_a and the number of free ribosomes r. (Right:) Corresponding estimation of the weighted sum $\sum_{k=\{r,nr\}} \left[1 + \frac{1}{E_{mk}}\right] J_k(\mu, r)$. The black line in both subplots depicts the value of mature available ribosomes r_a as a function of the free ones r using the number of bound ribosomes obtained from the data in [42].

$$J_k = \frac{l_{pk}}{\nu_t(s_i)} \frac{\beta_p \beta_m}{d_m} \frac{1}{r} \,. \tag{A.90}$$

We took into account that the data was obtained for fast growing cells, with doubling time $t_d = 21$ minutes. Under this condition, it is sensible to consider that the intracellular substrate will be saturated and the cells are growing at its maximum growth rate., so that $\nu_t(s_i) = \nu$. We evaluated (A.90) considering the set of all non-ribosomal proteins, the ribosomal ones, and the full set of proteins, and obtained the corresponding fractions Φ_t^h and Φ_b^h.

Figure A.6(left) shows the experimental values of Φ_t^h and Φ_b^h obtained as a function of the number of free ribosomes r. As expected, and in agreement with the estimations shown in Figure A.5(left), the values of Φ_b^h kept very close to 1 for a wide range of values of the number of free ribosomes. Recall the data in [42] was obtained for fast growing cells for which it is sensible to consider saturated intracellular substrate so that the cells are growing at its maximum growth rate and the number of free ribosomes is very small. This is consistent with the result

shown in Figure A.5(left, black line) where the estimated values of Φ_b are plotted along with the value of mature available ribosomes r_a as a function of the free ones r using the number of bound ribosomes obtained from the data in [42]. In order to keep the experimental values of Φ_b^h over 0.98, the number of free ribosomes must keep below the limit of a few hundreds. Notice also the experimental constant ratio $\Phi_t^h/\Phi_b^h \approx 0.83$. in agreement with [14, 15]. That is, around 17% of the total number of mature available ribosomes are, in average, located at the RBSs.

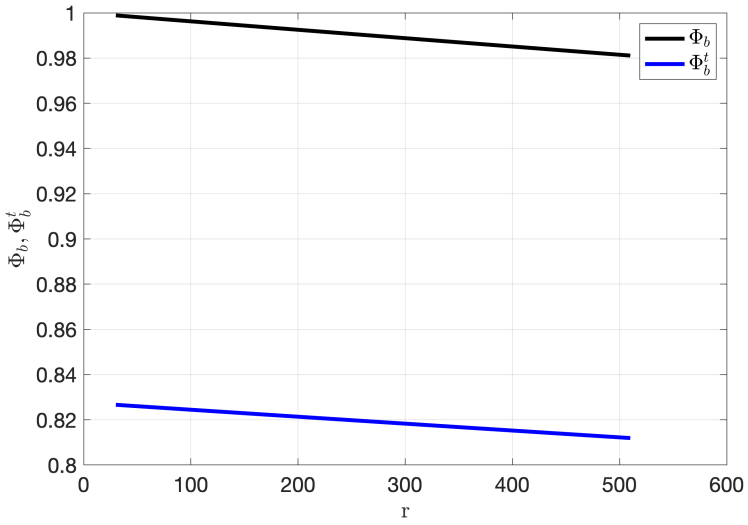

Figure A.6: Experimental values of Φ_b^h (black) and Φ_t^h (blue) as a function of the number of free ribosomes r obtained using the data in [42]. The ratio $\Phi_t^h/\Phi_b^h = 0.828$ is constant.

A.12 Evaluation of the maximum resources recruitment strength.

In this section we evaluate, on the one hand, the order of magnitude of the resources recruitment strength for the protein-coding genes in *E. coli*. This is useful to evaluate the maximum burden (measured as the sum of the resources recruitment strengths) in the native*E. coli* host cell and estimate how many genes are active.

We used the data in [42] and expression (A.90) to calculate the individual values of maximum resources recruitment strength J_k evaluated at $r = 1$ for the set of non-ribosomal and ribosomal protein-coding genes and sorted them by magnitude

and as the ratio between the sorted values of J_k and the corresponding protein lengths. The results are shown in Figure A.7.

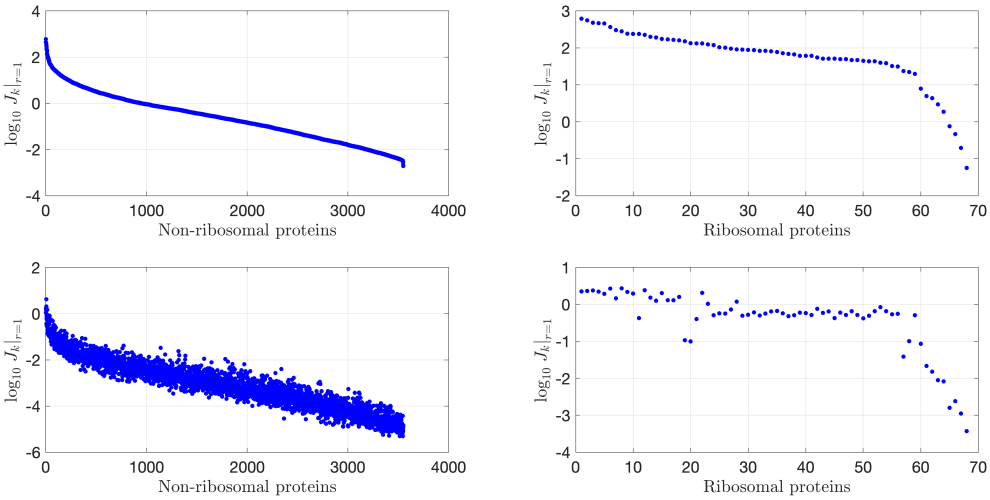

Figure A.7: (Top:) Maximum resources recruitment strength $J_k(r = 1)$ for the set of non-ribosomal (left) and ribosomal protein-coding genes (right) in *E. coli* sorted by magnitude in logarithmic scale. (Bottom:) Ratio between the sorted maximum resources recruitment strength $J_k(r = 1)$ and the corresponding protein length (aa).

As a proxy to estimate how many genes are active at a given time we calculated the cumulative sum of the maximum resources recruitment strengths and obtained how many genes being expressed are required to explain both 95% and 99% of the total cumulative sum. We did this independently for both ribosomal and non-ribosomal proteins. Figure A.8 shows the results obtained. Notice the same results will be obtained if using the weighted resources recruitment strengths since the ratio Φ_t^h/Φ_b^h is constant.

Our results show that out of the 68 ribosomal genes, 49 of them (72%) explain 95% of the cumulative sum of the maximum resources recruitment strength of the ribosomal genes. To explain 99% we need 57 ribosomal genes (84% of them). On the other hand, for non-ribosomal genes we need 875 out of 3551 genes (25%) to explain 95% of the cumulative sum and 1735 (49%) to explain the 99%.

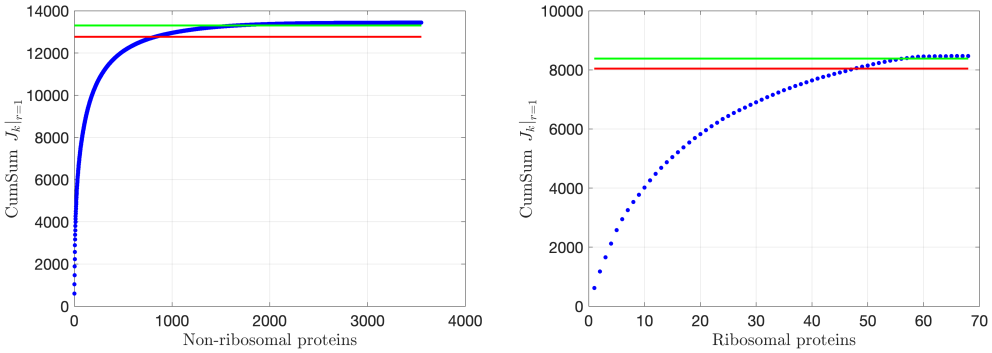

Figure A.8: Cumulative maximum resources recruitment strength for the set of non-ribosomal (left) and ribosomal protein-coding genes (right) in *E. coli*. The horizontal lines correspond to the 95% (red) and 99% (green) levels respectively.

A.13 Estimation of the number of free ribosomes in the cell

Estimation of the number of free ribosomes in the cell, r, is key for assessing the competition among the cell circuits for cellular resources. The results in the previous sections suggest that an extremely low number of free ribosomes, with order of magnitude in the range 10^1, gives a high sensitivity of the total cumulative sum of the maximum resources recruitment strength with respect to variations in the amount of free ribosomes (see Figures A.5 and A.6) while an order of magnitude in the range 10^2 is enough to keep a robust value of the total amount of recruited resources with respect to variations in the number of free ribosomes. That is, by not expressing superfluous resources, the cell forces a competition for them that induces a high sensitivity of the total amount of recruited resources with respect to variations in the number of free ribosomes, while a small surplus of superfluous resources induces robustness.

To evaluate the range of expected values of r we used experimental data of the translation efficiency per mRNA. Notice from the dynamics (A.20) for a protein p_k we can define:

$$Y_{p/\text{mRNA}} \triangleq \frac{\nu}{l_{pk}} \frac{J_k r}{\omega_k} = 0.62 \frac{\nu}{l_e} \frac{r}{\frac{d_{mk}}{K^k_{C^0}(s_i)} + \mu r} \frac{s_i}{K_{sc} + s_i}, \tag{A.91}$$

where s_i is the copy number of molecules of intracellular substrate, d_{mk} is the mRNA degradation rate constant, recall $K^k_{C^0}(s_i)$ is a substrate dependent parameter essentially related to the RBS strength and we consider that substrate

availability will only affect translation and not transcription (see Section A.1). Notice $Y_{p/\mathrm{mRNA}}$ is the number of protein copies produced per transcript.

We used the data from [42] to estimate an upper bound for the number of free ribosomes r using (A.91) and the values for $Y_{p/\mathrm{mRNA}}$ obtained using (A.89):

$$Y_{p/\mathrm{mRNA}} = \frac{\beta_{pk}}{d_{mk}}\,. \tag{A.92}$$

Then, the relationship between the RBS strength-related term $K^k_{C^0}$ and the free ribosomes r becomes:

$$0.62\frac{\nu}{l_e}\frac{r}{\frac{d_{mk}}{K^k_{C^0}(s_i)} + \mu_m f(s_i) r} f(s_i) = \frac{\beta_{pk}}{d_{mk}}\,, \tag{A.93}$$

where $f(s_i) = s_i/(K_{sc}+s_i)$ and we have used equations (A.56), (A.57) and (A.58) relating the specific growth rate μ with the maximum one μ_m and the availability of intracellular substrate. Notice that, for any given protein and intracellular substrate availability, the number of free ribosomes will determine the required value of the RBS strength-related term $K^k_{C^0}(s_i)$ to attain the experimental value of the translation efficiency per mRNA $Y_{p/\mathrm{mRNA}}$.

The translation efficiency given by expression (A.93) depends on the ribosomes density $1/l_e$. An average ribosomes density around 4.2 ribosomes per 100 codons in optimal growth conditions has been reported in the literature for the prokaryote *L. lactis* [82]. This value is the same we obtained from the data in (A.90) by considering the total number of available active ribosomes $\Phi_b r_a$ obtained in Section A.12 (see Figure A.6) and dividing it by the sum of the lengths of all proteins weighted by a factor 0.5 to account for the estimation that 50% of the genes are active (explain 99% of the cumulative sum of the maximum resources recruitment strength). Similar values are found for other organisms [32]. For *E. coli* a value of 3.5 is given in [77]. The ribosomes density is inversely log-linearly related to the length of the coding sequence, with a slope quite consistent for a variety of organisms [32]. To account for this, we approximated a power law consistent with the findings in [32] and resulting in 4 ribosomes per 100 codons for an average protein length of 330 codons. We obtained the relationship:

$$\frac{1}{l_e} = \frac{0.0703}{l_{pk}^{0.097}}\,. \tag{A.94}$$

This gives a range $l_e \subset [18, 31]$, with a value $l_e = 25$ for the average protein length. The minimum value is consistent with the shortest protein length (18 codons) in the database we used.

Figure A.9 shows the results obtained for the set of all ribosomal and non-ribosomal proteins and their average values.

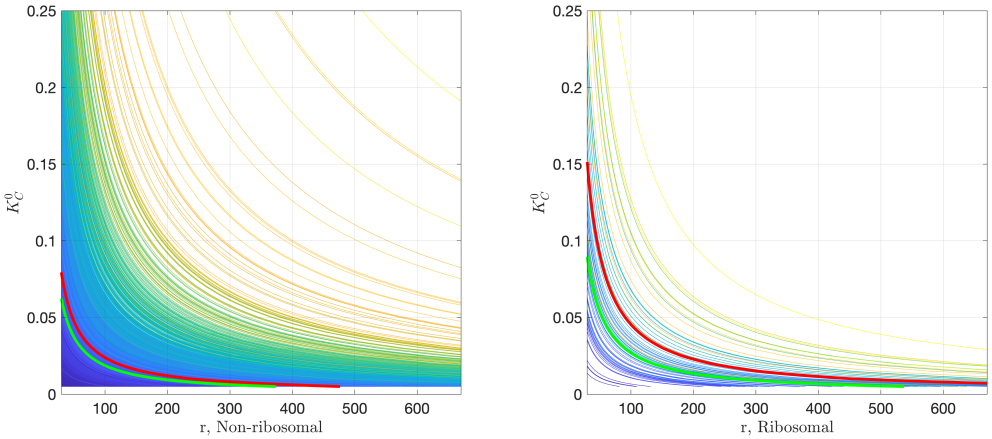

Figure A.9: Relationship between the RBS strength-related term $K_{C^0}^k$ and the number of free ribosomes r obtained using the experimental data in [42] for non-ribosomal (left) and ribosomal (right) proteins in *E. coli*. Thin lines correspond to the experimental value of the translation efficiency per mRNA $Y_{p/\mathrm{mRNA}}$ for each protein. The red thick line corresponds to the mean for all proteins in the corresponding non-ribosomal and ribosomal sets. The green thick line corresponds to the approximated mean when the term associated to the maximum specific cell growth rate is neglected in the expression (A.93).

From the results shown in Figure A.9, notice that the number of free ribosomes r required to explain the experimental value of the translation efficiency per mRNA $Y_{p/\mathrm{mRNA}}$ for each protein increases as the value of the RBS strength-related term $K_{C^0}^k$ decreases. Indeed, the more free ribosomes are available, the less competition for shared resources. The number of free ribosomes r is an indicator of the level of competition for resources. Thus, expression (A.93) implies that a gene producing short-living transcripts will require, for the same level of competition, a stronger RBS to achieve the same translation efficiency per mRNA $Y_{p/\mathrm{mRNA}}$ as one with long-living transcripts.

To estimate an upper limit for the copy number of free ribosomes required to achieve the experimental translation rates per mRNA, we considered an upper bound for the RBS strength-related term $K_{C^0}^k$.

Recall from equation (A.10) that $K^k_{C^0}$ term is a function of the intracellular substrate availability. Since the data we used from [42] was obtained for fast growing cells, we can consider intracellular substrate saturation. Under this condition we get:

$$K^k_{C^0} = \frac{K^k_b}{K^k_u + \frac{\nu}{l_e}} \,. \tag{A.95}$$

Using the value of ν in Table A.2 and the range of l_e given above, we estimate $\nu/l_e \subset [40,\, 70]$ (molec$^{-1}\cdot$min^{-1}). On the other hand, the values of the association and dissociation rates of the ribosome to the RBS, K^k_b and K^k_u, may vary in a large range. Values $K^k_b \subset [3,\, 15]$ (molec$^{-1}\cdot$min^{-1}) are found in the literature (see Table A.2). We use a conservative upper bound $K^{\max}_b = 10$ (molec^{-1}) considering binding is diffusion controlled. From the literature, we consider a range for the dissociation rate $K^k_u \subset [3,\, 135]$ (min^{-1}). Overall, these estimates give us a range (under the assumption of intracellular substrate saturation) $K^k_{C^0} \subset [0.02,\, 0.2]$ (molec^{-1}).

From the results shown in Figure A.9, notice that a maximum number of free ribosomes $r \approx 350$ can confidently explain the translation efficiencies per mRNA $Y_{p/\mathrm{mRNA}}$ for almost all proteins while maintaining the value of the RBS strength-related term $K^k_{C^0} < 0.2$. This estimation for the amount of free ribosomes is in complete agreement with the estimation $r = \mathbf{N}(350, 35)$ in [53].

With the upper limit $r \approx 350$ we could explain the translation efficiencies per mRNA $Y_{p/\mathrm{mRNA}}$ calculated as β_{pk}/d_{mk} (see equation (A.93)) using the data in [42] but for a small set of 80 non-ribosomal proteins out of 3551 (2.25%). In 52 of them, this could be attributed to their extremely long-living transcripts. In the remaining 28 ones, to their very high translation efficiency per mRNA $Y_{p/\mathrm{mRNA}}$ expected from their values of β_{pk} and d_{mk} given in [42]. This can be explained by rewriting (A.93) as:

$$Y_{p/\mathrm{mRNA}} = 0.62\frac{\nu}{l_e}\frac{f(s_i)\frac{K^k_{C^0}(s_i)}{d_{mk}}r}{1+\mu_m f(s_i)\frac{K^k_{C^0}(s_i)}{d_{mk}}r} = \frac{\beta_{pk}}{d_{mk}} \,. \tag{A.96}$$

Figure A.10 shows a plot of the function (A.96), as a function of its argument $f(s_i)K_{C^0}(s_i)r/d_m$ for two mRNA degradation rates corresponding to short and long-living mRNAs and ribosomes densities in the range $l_e = [18, 31]$. Notice $Y_{p/\mathrm{mRNA}} = \beta_{pk}/d_{mk}$ saturates at the maximum attainable value:

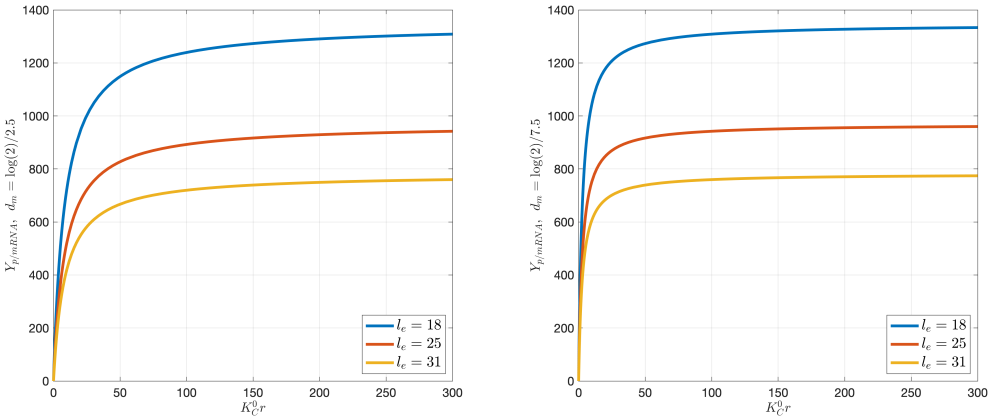

Figure A.10: Translation efficiency per mRNA $Y_{p/\mathrm{mRNA}}$ as a function of $f(s_i)K_{C^0}r$ for two mRNA degradation rates corresponding to mRNA half-lives 2.5 minutes (left) and 7.5 minutes (right) for ribosomes densities corresponding to $l_e = \{18, 25, 31\}$ codons.

$$Y_{p/\mathrm{mRNA,max}} = 0.62\frac{\nu}{l_e}\frac{1}{\mu_m}\,. \tag{A.97}$$

Saturation is reached for lower values of the argument $f(s_i)K_{C^0}(s_i)r/d_m$ as the transcripts have longer half-lives, i.e. smaller values of the degradation rate d_m.

The few cases our model could not predict all have values of β_{pk}/d_{mk} above $Y_{p/\mathrm{mRNA,max}}$ for the values of ν and l_e used in the model.

On the other hand, note that for very low values of $f(s_i)K_{C^0}(s_i)r/d_m$ such that $f(s_i)K_{C^0}(s_i)r \ll d_m/\mu_m$ we can approximate:

$$Y_{p/\mathrm{mRNA}} \approx 0.62\frac{\nu}{l_e}f(s_i)\frac{K_{C^0}^k(s_i)}{d_{mk}}r = \frac{\beta_{pk}}{d_{mk}}\,, \tag{A.98}$$

from which we get:

$$\beta_{pk} \approx 0.62\frac{\nu}{l_e}f(s_i)K_{C^0}^k(s_i)r\,. \tag{A.99}$$

That is, for a highly competitive scenario where the number of free ribosomes is sufficiently small (e.g. in the order of few tens to few hundreds for typical values of d_{mk}, $\mu_m = 0.032\,\mathrm{min}^{-1}$ and $f(s_i)K_{C^0}(s_i)$ at its maximum estimated

value $f(s_i)K_{C^0}(s_i) = 0.2$) the translation rate (proteins per mRNA per time unit) is proportional to the ribosomes density $0.62/l_e$, the effective maximum translation rate per codon attainable for a given substrate availability $\nu f(s_i)$, the RBS strength $K^k_{C^0}$, and the available free ribosomes r.

Notice under this scenario the translation rate will suffer large stochastic fluctuations caused by stochastic fluctuations in the number of free ribosomes. In this case, the transcription rate for a given RBS-strength mainly depends on the competition for cellular resources and, therefore, on the number of free ribosomes r, and it is largely independent of the specific growth rate.

A.14 Estimation of the parameters for ribosomal and non-ribosomal endogenous proteins

We considered the model expressions at steady-state in Section 2.3.1 and estimated the RBS-strength related parameters K^k_b, K^k_u with the transcription rates ω_{k} with $k = \{r, \mathrm{nr}\}$ and the fraction Φ_m, so that our model provided a good fit of the specific growth rate at steady-state.

The only input information given to the model was the value of the peptide chain elongation rate values $\nu_t(s_i)$ as a function of growth rate obtained from [15]. This is equivalent to feeding the model only with the available amount of substrate s_i. To this end, we expressed the effective maximum translation rate as $\nu_t(s_i) = \nu f(s_i)$, where ν is the maximum attainable peptide synthesis rate (see equation (2.2)) and $f(s_i) = s_i/(K_{sc} + s_i)$. Notice $f(s_i)$ is monotonous with the amount of intracellular substrate s_i. From the experimental values of $\nu_t(s_i)$ as a function of growth rate, and knowing the maximum attainable peptide synthesis rate ν (see Table A.2), we obtained the experimental values of $f(s_i)$ for each growth rate. We used these to feed our model. This is tantamount to feeding the model with the substrate s_i but the value of the substrate and host dependent Michaelis constant K_{sc} needs not to be known.

Then, we fitted the model parameters using the experimental growth rate as the output to predict. So as not to penalize large errors in excess, which in our case are more prone to happen for larger values of the growth rate, we minimized the sum over the experimental data points of the absolute prediction error of the growth rate:

$$I = \sum_{k=1}^{n_p} \left| \mu_{\exp}(s_{i,k}) - \hat{\mu}(s_{i,k}) \right| . \tag{A.100}$$

We considered $N_r = 57$; and $N_{nr} = 1735$, corresponding to the number of genes that explain 99% of the cumulative sum of the resources recruitment strengths for ribosomal and non-ribosomal proteins respectively (see Section 2.3.2). We also considered the average mRNA degradation rates $d_{m,r} = 0.16$ min^{-1} and $d_{m,nr} = 0.2$ min^{-1} (see Table A.2). Using the value of ν in Table A.2 and the range of l_e obtained in Appendix section A.15, we estimated $\frac{\nu}{l_e} \subset [40, 70]$ (molec$^{-1} \cdot$ min^{-1}).

Moreover, the values of the association and dissociation rates of the ribosome to the RBS, K_b^k and K_u^k, may vary in a large range. Values $K_b^k \subset [3, 15]$ (molec$^{-1} \cdot$ min^{-1}) are found in the literature (see Table SI.2). We used a conservative upper bound $K_b^{\max} = 10$ (molec^{-1}) for the search space, considering binding is diffusion controlled. From the literature, we also considered a search range for the dissociation rate $K_u^k \subset [3, 135]$ (min^{-1}). Overall, these estimates gave us a range $K_{C^0}^k \subset [0.02, 0.2]$ (molec^{-1}) for the effective RBS-strength under the assumption of intracellular substrate saturation. We ran 200 instances of the parameter fitting algorithm using the global optimization software MEIGO [29], available at `http://gingproc.iim.csic.es/meigo.html`, and obtained the weighted mean of the 25 runs achieving the best minimum value for the sum over the experimental data points of the absolute growth rate prediction error. The resulting average best fit estimated parameters are given in Table 2.1.

A.15 Estimated fractions of ribosomes

Additional results for Section 2.3.2 are given here. The total number of ribosomes (both experimental and estimated) much increases for very fast growing cells. Thus, the fraction of free ribosomes with respect to the total number only increases from 0.08% up to 1.37% for cell duplication times between 100 and 24 minutes respectively even though the number of free ribosomes multiplies by almost 200-fold (see Figure A.11). The estimated value of the fraction of mature ribosomes with respect to the total number of ribosomes was $\Phi_m \approx 0.90$ and the estimated fraction of active bound ribosomes kept roughly constant with growth rate at $\Phi_t^h \Phi_m \approx 0.78$.

Notice also the logarithmic affine relationship between the number of free ribosomes r and its flux μr, ($\log_{10}(r) \approx 4.07 + 0.78 \log_{10}(\mu r)$) reflecting a power-law relationship between growth rate and number of free ribosomes.

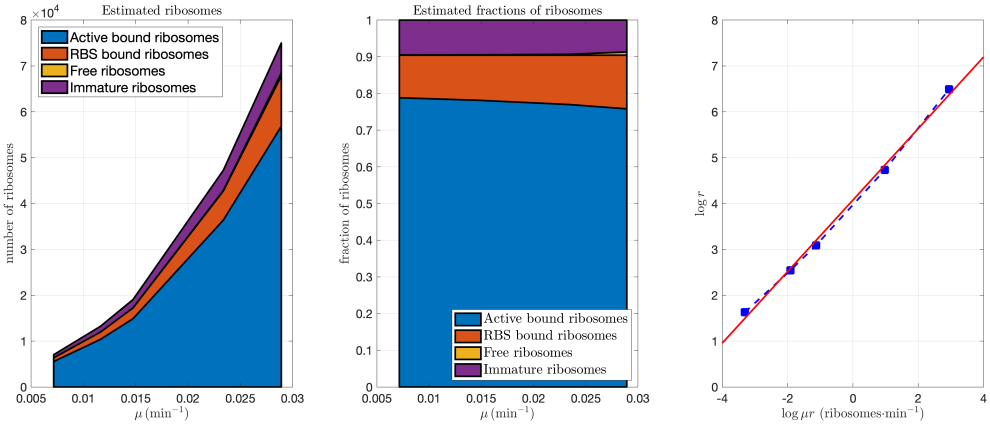

Figure A.11: Estimated number of ribosomes (Left) fractions of ribosomes (Center) and logarithmic affine relationship between the number of free ribosomes r and its flux μr (Right)

A.16 Effect of substrate availability of growth rate and specific synthesis rate

Additional images A.12 and A.13 for Results Section 2.3.4 are given here. In both cases, the effect of substrate variation is considered. The function $f(s_i)$ corresponds to the normalized attainable peptide synthesis rate defined in equation (A.4), so that:

$$f(s_i) \triangleq \frac{\nu_t(s_i)}{\nu} = \frac{s_i}{K_{sc} + s_i}\,. \tag{A.101}$$

Therefore $f(s_i)$ is a saturated monotonous increasing function with the intracellular substrate s_i, taking values in the range $[0, 1]$.

A.17 Software code and data

The software code and data are available in a GitHub repository at:
`https://github.com/sb2cl/Resources-allocation-NoPi21`

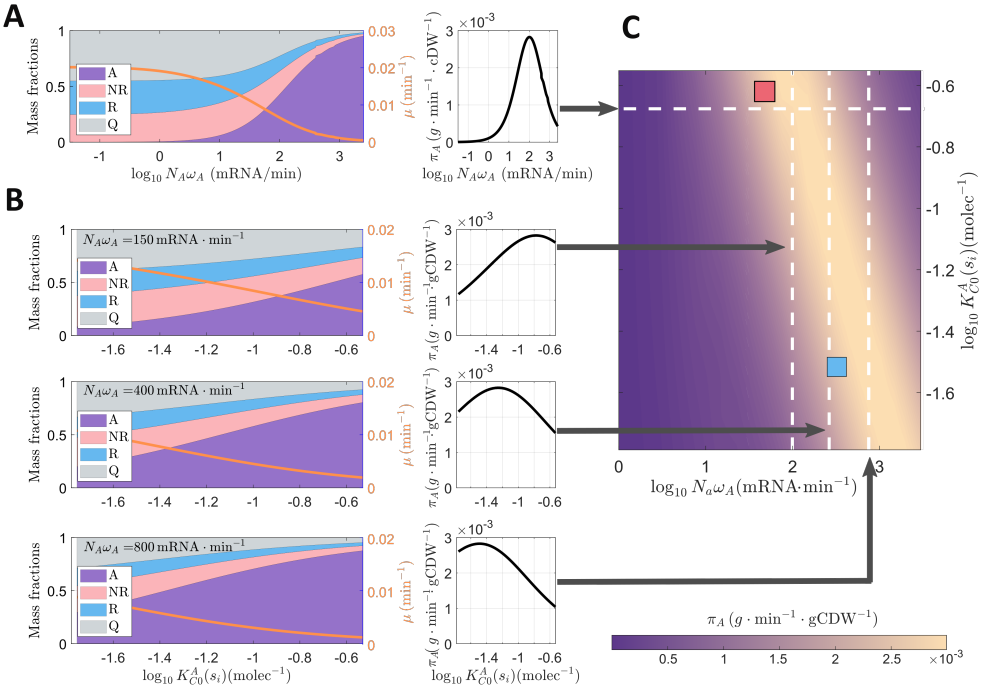

Figure A.12: **A:** Effect of increasing mRNA synthesis rate on growth rate and mass fractions (*left*) and specific protein synthesis rate (*right*). **B:** Effect of RBS strength variation on growth rate and mass fractions (*left*) and specific protein synthesis rate (*right*) for the three values $N_A\omega_A = \{150, 400, 800\}$. **C:** Specific protein synthesis rate across the expression space $N_A\omega_A, K^A_{C0}$. The pink and blue squares correspond to the average lumped values of $N_x\omega_x, K^x_{C0}(s_i)$ for the non-ribosomal (pink) and ribosomal (blue) endogenous protein coding-genes in an *E. coli* host respectively. The dashed white lines correspond to the four scenarios analyzed in the panels A and B.

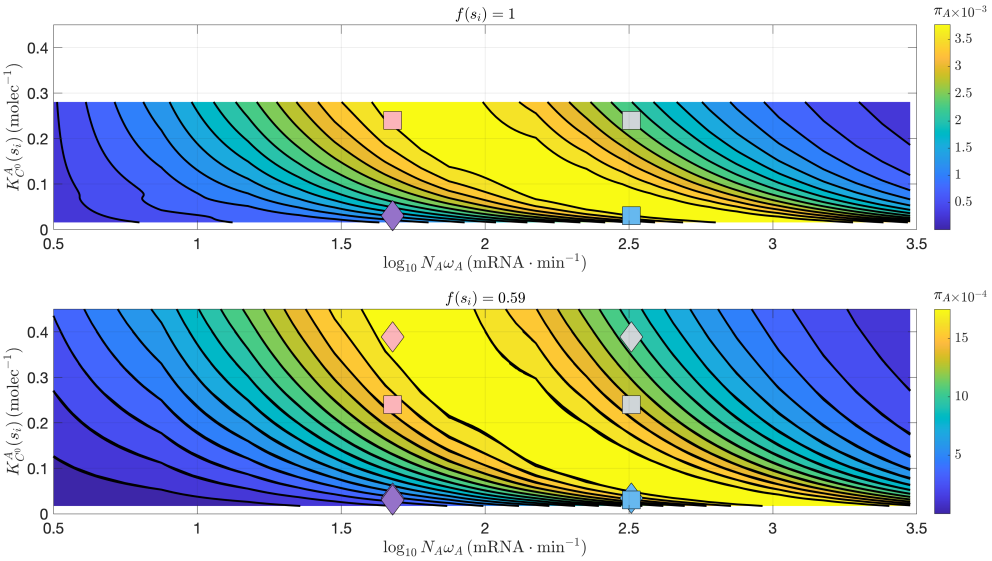

Figure A.13: Effect of the variation of substrate on the specific optimal synthesis rate of exogenous protein-coding genes. The contour lines show the specific synthesis rate as a function of the effective RBS strength and the mRNA synthesis rate. They were obtained by simulating the expression of a generic exogenous protein-coding gene with varying strengths across the expression space $N_A\omega_A, K^A_{C^0}$. We considered genes with four differential characteristics. **Top:** Saturated substrate. **Bottom:** Low substrate condition. The values of RBS strength and mRNA synthesis rate are now depicted as diamonds for the new scenario. The values of the previous substrate-saturated scenario are kept as squares for the sake of comparison. Recall the dependence of the effective RBS strength $K^x_{C^0}(s_i)$ on the availability (tantamount, nutrient quality) of the intracellular substrate. In our model, strong RBSs are more affected by variations of the substrate than weak ones. As the substrate decreases, the effective RBS-strength for genes with weak RBSs do not appreciably change. Notice that these genes require less resources (per gene) than the ones with strong RBSs. For the genes with strong RBSs, the apparent RBS strength increases as the substrate decreases. They increase their avidity for scarce resources. This way, they both keep their relative positions within the specific synthesis rate space.