Assessing the robustness and vulnerability of genome-scale constraint-based models

Evaluación de la robustez y vulnerabilidad de modelos basados en restricciones a escala genómica

Alexandru Ioan Oarga Hategan

Director: Jorge Emilio Júlvez Bueno

Departmentamento de Informática e Ingeniería de Sistemas (DIIS)
Universidad de Zaragoza

Máster en Ingeniería Informática

2022

Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This dissertation is my own work and contains nothing which is the outcome of work done in collaboration with others, except as specified in the text. This dissertation main body contains fewer than 10,000 words.

Alexandru Ioan Oarga Hategan 2022

Acknowledgements

This work was supported by a 2021 final master thesis grant from The Engineering Research Institute of Aragon (I3A).

Abstract

Despite antimicrobial resistance is an increasing health emergency, the pace of development of new drugs is slow due to the high cost and uncertain success of the process. At the same time, the development of high-throughput technologies has allowed the integration of biological data into genome-scale models of multiple microorganisms that have proven useful in areas such as metabolic engineering. These models have the potential to offer cost-effective means to identify vulnerabilities in metabolism, which can serve as potential therapeutical targets for drugs.

In this work, we formally define chemical reactions of metabolism that are broadly acknowledged as vulnerabilities. In order to exploit all the data available on the model, we develop a procedure to integrate growth constraints into the identification of these vulnerabilities. By doing so we are able to identify vulnerabilities in metabolism that are consistent with a given growth rate of the model. Moreover, we also study how these constraints affect current optimisation methods used to compute growth in a model.

In addition, in this work, we also study the mechanisms of metabolism robustness, this is, the ability to sustain growth against external disruptions. A method is proposed to identify sets of reactions that are essential for growth and sets of reactions that are redundant and therefore account for the robustness of metabolism. We show that growth itself is produced as a combination of the two previous sets. Moreover, the problem of computing a minimum set of reactions that can produce optimum growth is formally stated. It is proven that such a problem is NP-complete and a technique to reduce the search space of the problem is proposed. Finally, we also show that flux variability is an indicator of reactions essentiality and discuss how it is related to redundancy in metabolism. The methods proposed in this work are experimentally applied in a genome-scale model of *Plasmodium Falciparum*.

Resumen

A pesar de la creciente emergencia sanitaria que supone la resistencia microbiana a los antibióticos, el ritmo de desarrollo de nuevos medicamentos es lento debido al alto costo y al éxito incierto del proceso. Al mismo tiempo, el desarrollo de las tecnologías de secuenciación ha permitido la integración de datos biológicos en modelos a escala genómica de múltiples microorganismos, los cuales han demostrado ser útiles en áreas como la ingeniería metabólica. Estos modelos tienen el potencial de ofrecer alternativas computacionales más eficientes para la identificación de vulnerabilidades en el metabolismo, los cuales pueden ser potenciales dianas terapéuticas para fármacos.

En este trabajo, se definen de manera formal aquellas reacciones químicas del metabolismo que son ampliamente reconocidas como vulnerabilidades del metabolismo. Con el objetivo de aprovechar toda la información disponible en el modelo, se desarrolla un procedimiento para integrar restricciones de crecimiento en la identificación de estas vulnerabilidades. De esta manera, conseguimos identificar vulnerabilidades que son consistentes con un determinado ratio de crecimiento en el modelo. Además de esto, se estudia el efecto que estas restricciones tienen en los métodos de optimización actuales utilizados para la estimación de crecimiento.

Además de esto, en este trabajo también se estudian los mecanismos de robustez del metabolismo, esto es, aquellos que le permiten mantener el crecimiento frente a perturbaciones externas. Para ello, se propone un método para identificar aquellos conjuntos de reacciones que resultan esenciales para el crecimiento, y aquellas que resultan redundantes y que por tanto contribuyen a la robustez del metabolismo. Se demuestra que el crecimiento en el metabolismo es el resultado de una combinación de los dos conjuntos anteriores. El problema del cálculo del mínimo conjunto de reacciones necesario para un crecimiento óptimo también se propone formalmente. Se demuestra que este problema es NP-completo y se propone una técnica para reducir el espacio de búsqueda. Finalmente, se demuestra que la variabilidad en el flujo de las reacciones es un indicador de la esencialidad de estas y se discute su relación con la redundancia en el metabolismo. Los métodos propuestos en este trabajo se aplican experimentalmente a un modelo a escala genómica de la bacteria *Plasmodium Falciparum*.

Contents

Chapter 1

Introduction

Antimicrobial Resistance (AMR) occurs when bacteria, viruses, fungi and parasites evolve and no longer are affected by conventional medicines thus making infections harder to treat and increasing the risk of disease spread and severe illness. The emergence of multi-drug resistant bacteria (MDR) is particularly alarming, as they can cause infections untreatable with existing antibiotics. According to the World Health Organisation, AMR is one of the primary global public health threats to humanity due to its high increasing rate [61].

Despite this increasing emergency, the antibiotics development pipeline entails a huge cost with uncertain success. Preclinical stages of the process usually involve searching for antibacterial compounds in nature and then putting them through a series of experiments to study their drug feasibility. In this context, computational methods have the opportunity to offer cost-effective alternatives to traditional screening methods [47].

In recent years, the emergence of high-throughput technologies allowed the integration of transcriptomic data of multiple pathogens into large biological datasets. This integration paved the way for the reconstruction of metabolic models of biological systems which directly led to the possibility of modelling these systems computationally (see Figure 1.1). [50, 48].

Metabolism is the set of basic life processes that take place in the cell, and it is the means by which cells can maintain life and grow from their environment. Metabolism can be represented as a metabolic network, which includes all the metabolic reactions that can occur in a cell. As of 2019, Genome-Scale Models (GEM) of metabolism have been reconstructed for more than 6000 organisms including bacteria, archaea and eukaryotes [16].

To provide an example, Figure 1.2a shows a GEM consisting only of *Escheria Coli* nucleus metabolism. This model consists of 54 compounds and 85 chemical reactions. In 1994, *Varma et.al.* [56] made a significant breakthrough by showing that, with this model of *Escheria Coli* nucleus, they were able to accurately predict glucose uptake in relation to the growth rate (Figure 1.2b). In addition, this model also enabled the modelling of the change of compound concentrations through time (Figure 1.2c). After almost 30 years, technological advances and available data have substantially pushed further the field. Nowadays, models of *E. coli* are comprised of more than 2700 reactions [25] with over 1500 genes [13] and are able to model more complex behaviours such as transcriptomics machinery [41] or stress response [8].

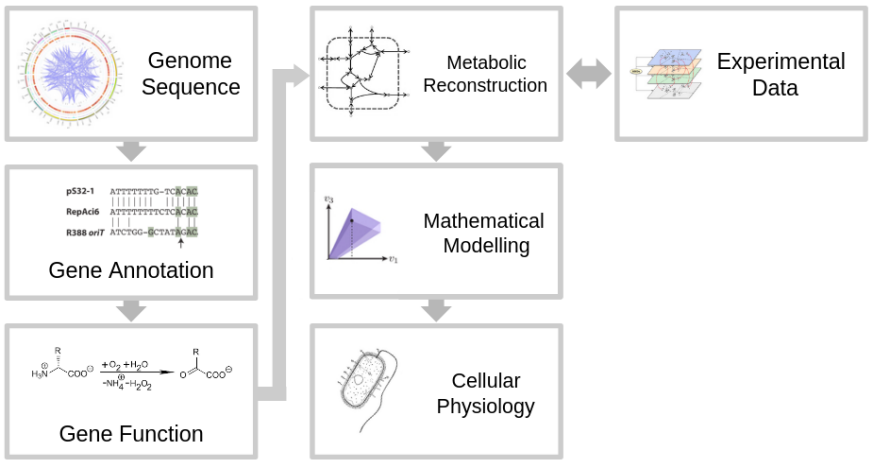
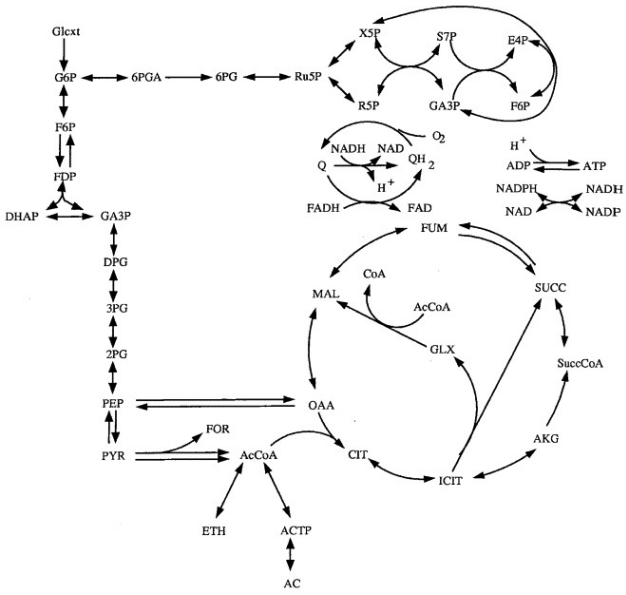
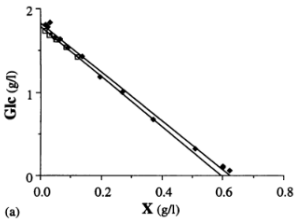


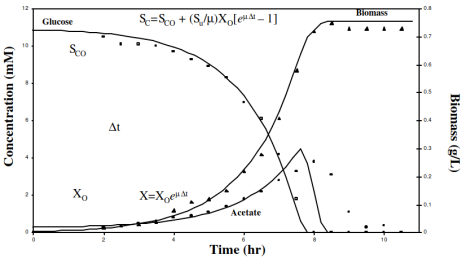
Figure 1.1 Holistic approach of metabolic modelling. High-throughput sequencing technology and automatic annotation tools enabled the reconstruction of microorganisms' metabolism. Mathematical models can be used over these reconstructions to make predictions and gain insight into the cellular behaviour of the modelled biological system. This figure is an extension of Fig.1 in [12].



(a) Metabolic network of *E. coli* nucleus with 54 metabolites and 85 reactions. Source: [34].



(b) Maximum aerobic glucose utilisation rate (Glc) based on growth rate (X). Source: [56].



(c) Predicted growth (Biomass), Glucose and Acetate concentration on aerobic batch culture. Source: [56].

Figure 1.2 *Escheria Coli* nucleus metabolic network and *in-silico* predictions obtained with this model by *Varma et al.*. Model predictions are plotted with continuous lines.

Current applications of GEMs include but are not limited to: expanding knowledge on microorganisms [19, 39] and microbial communities [29, 65], microbial engineering [37, 51], and drug discovery [48]. Moreover, these models have proven useful in areas such as oncology,

by studying drug targets in cancer metabolism [15], and viral diseases [3]. Drug targeting in pathogens is usually performed by considering essential genes whose enzyme inhibition can effectively kill a pathogen [16] or through metabolic network topology analysis [48]. For a comprehensive review of GEMs, applications see [17].

In this work, we will focus on GEMs applications to drug discovery. It is well known that bacteria evolved their metabolism to adapt to different environments and even today to avoid antibiotics drugs. Despite the advances made with GEMs, what confers metabolism and its robustness has been surprisingly understudied. Contributions in this field have promising potential for both understanding and drug targeting on metabolism. This work will aim to define potential drug targets, study metabolism robustness under certain conditions and propose methods to exploit metabolic vulnerabilities for drug targetting.

1.1 Thesis contributions and outline

This work is composed of a series of individual contributions whose shared goal is to provide methods to exploit metabolism vulnerabilities and shed light on the mechanisms that confer robustness to metabolism. The main contributions of this work are listed below:

- Vulnerabilities computation
  - **Provided formal definitions for widely recognised metabolism vulnerabilities** such as chokepoint reactions and essential reactions which can be appealing first-step targets for drug discovery (Chapter 2).
  - Proposed a framework to **study how these vulnerabilities change when the model is producing growth** in a steady state (Chapter 3).
  - Proved formally that **imposing flux variability constraints on each reaction forces the model to produce a given growth** (Chapter 4).
- Metabolism robustness
  - Proposed a **method for computing the minimum metabolism necessary to sustain growth**, which provides insights into metabolism robustness and how growth is produced (Chapter 5).
  - Showed that **reactions that are essential for growth are directly related to the flux** that each reaction is able to carry, which sheds light on the mechanisms that confer robustness to metabolism (Chapter 6).
- Evaluation
  - **Evaluated robustness and vulnerabilities identification** on *Plasmodium Falciparum* genome-scale model (Chapter 7).
- Conclusions and future work
  - Provided the overall conclusions of the work and future approaches (Chapter 8).

Computational tools used in this work are reported in Appendix B.

Chapter 2

Definitions

In this chapter, preliminary definitions are introduced. We will define constraint-based models, types of metabolic reactions and methods to estimate growth and metabolic flux. Furthermore, some well-known vulnerabilities are formally defined.

2.1 Preliminary definitions

2.1.1 Constraint-based models

Definition 2.1.1. A *constraint-based model* [55, 43] is a tuple $\{\mathcal{R},\ \mathcal{M},\ \mathcal{S},\ L,\ U\}$ where $\mathcal{R}$ is a set of *reactions*, $\mathcal{M}$ is a set of *metabolites*, $\mathcal{S} \in \mathbb{R}^{|\mathcal{M}|\times|\mathcal{R}|}$ is the stoichiometric matrix, and $L, U \in \mathbb{R}^{|\mathcal{R}|}$ are *lower and upper flux bounds* of the reactions.

Without loss of generality, it is assumed that $L[r] \leq U[r] \quad \forall r \in \mathcal{R}$.

All reactions are associated with a set of reactant metabolites and a set of product metabolites (one of these two sets can be empty). For example, the reaction $r{:}A \rightarrow 2B$ has a reactant metabolite A, and a product metabolite B with stoichiometric weight 2, i.e. two molecules of type B are produced per each molecule of type A that is consumed by r. The stoichiometric matrix $\mathcal{S}$ accounts for all the stoichiometric weights of the reactions, i.e. $S[m, r]$ is the stoichiometric weight of metabolite $m \in \mathcal{M}$ for reaction $r \in \mathcal{R}$.

Constraint-based models are inherently bipartite directed graphs and thus they can be represented graphically as Petri nets [38, 18], where places, drawn as circles, model metabolites, and transitions, drawn as squares, model reactions. The presence of an arc from a place(transition) to a transition(place) means that the place is a reactant(product) of the reaction modelled by the transition. The weights of the arcs of the Petri net account for the stoichiometry of the constraint-based model. In other words, the stoichiometric matrix of a constraint-based model and the incidence matrix of its corresponding Petri net coincide.

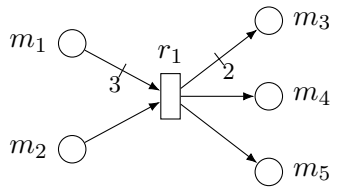


Figure 2.1 Example Petri net modelling a constraint-based model with only one reaction.

Example 2.1.1. The Petri net in Figure 2.1 represents a simple constraint-based model that consists of 1 reaction and 5 metabolites. The only transition r_1, models the reaction $r_1 : 3m_1 + m_2 \rightarrow 2m_3 + m_4 + m_5$.

2.1.2 Structural and flux-based definitions

We will introduce now structural definitions making use of Petri net notation: Let X denote a node of the net, i.e. a reaction or a metabolite. Then, ${}^\bullet X ({X}^\bullet)$ denotes the set of input(output) nodes of X. For instance, for a given reaction $r \in \mathcal{R}$, ${}^\bullet r$ denotes its set of *reactants* and $r^\bullet$ its set of *products*; for a given metabolite $m \in \mathcal{M}$, ${}^\bullet m$ denotes its set of *producing reactions* and $m^\bullet$ its set of *consuming reactions*.

The previous definitions only take into account the structure of the network and disregard the flux bounds of the reactions. In order to capture the fact that reactions can proceed forwards or backwards, e.g. reversible reactions, new sets of reactants, products, consumers and producers that take into account flux bounds are defined:

- *Flux-dependent set of reactants of* r:
  ${}^\star r = \{m \in \mathcal{M} | (S(m,r) < 0 \wedge U[r] > 0) \vee (S(m,r) > 0 \wedge L[r] < 0)\}$
- *Flux-dependent set of products of* r:
  $r^\star = \{m \in \mathcal{M} | (S(m,r) > 0 \wedge U[r] > 0) \vee (S(m,r) < 0 \wedge L[r] < 0)\}$
- *Flux-dependent set of producers of* m:
  ${}^\star m = \{r \in \mathcal{R} | m \in r^\star\}$
- *Flux-dependent set of consumers of* m:
  $m^\star = \{r \in \mathcal{R} | m \in {}^\star r\}$

The flux bounds can be also used to classify reactions as *dead*, *reversible* or *non-reversible*:

Definition 2.1.2. A reaction $r \in \mathcal{R}$ is dead if $L[r] = U[r] = 0$.

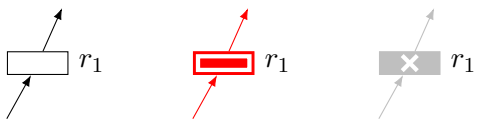
Definition 2.1.3. A reaction $r \in \mathcal{R}$ is reversible if $L[r] < 0 < U[r]$.

Definition 2.1.4. A reaction $r \in \mathcal{R}$ is non-reversible if r is not dead and r is not reversible.

From the above definitions, it can be deduced that r is non-reversible if $(0 \leq L[r] \wedge 0 < U[r]) \vee (L[r] < 0 \wedge U[r] \leq 0)$.

Notice that dead reactions will never have flux. Since this might indicate a deficiency of the model or a blocked pathway of the organism, special attention will be paid to dead reactions.

Non-reversible reactions, reversible reactions and dead reactions will be represented as white rectangles, red double rectangles, and grey crossed rectangles respectively (see Figure 2.2) For convenience, metabolites only produced or only consumed by dead reactions will be represented with grey circles as well.



(a) Non-reversible reaction (b) Reversible reaction (c) Dead reaction

Figure 2.2 Types of reactions.

Example 2.1.2. The Petri net in Figure 2.3 represents a constraint-based model with 6 metabolites and 10 reactions. Lets say that reactions r_3, r_4 have all flux bounds equal to $[-10, 10]$, reactions $r_1, r_2, r_5, r_6, r_7, r_8, r_g$ have all flux bounds equal to $[0, 10]$ and reaction r_9 has flux bounds equal to $[0, 0]$. In this model, reactions r_3, r_4 are reversible reactions (i.e. $RR = \{r_3, r_4\}$), reaction r_9 is a dead reaction (i.e. $DR = \{r_9\}$) and reactions $r_1, r_2, r_5, r_6, r_7, r_8, r_g$ are non-reversible reactions (i.e. $NR = \{r_1, r_2, r_5, r_6, r_7, r_8, r_g\}$).

2.1.3 Flux Balance Analysis

Flux Balance Analysis (FBA) [44] is a mathematical procedure for the estimation of steady-state fluxes in constraint-based models. FBA can be used, for instance, to predict the maximum growth rate of an organism. Let $v \in \mathbb{R}^{|\mathcal{R}|}$ be the vector of fluxes of reactions and $v[r]$ denote the flux of reaction r. At a steady state, it holds that $S \cdot v = 0$, where S is the stoichiometric matrix. Thus, the linear programming problem (LPP) for FBA is:

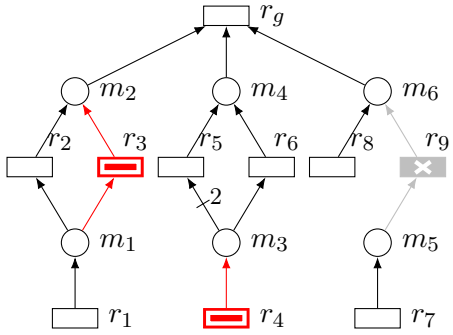


Figure 2.3 Example Petri net modelling a constraint-based model.

$$\begin{aligned} &\max\ z \cdot v \\ &st.\ \ S \cdot v = 0 \\ &\qquad L \leq v \leq U \end{aligned} \tag{2.1}$$

where $z \in \mathbb{R}^{|\mathcal{R}|}$ expresses the objective function.

Let r_g be the reaction that models growth (or biomass production). Without loss of generality, it will be assumed that $L[r_g] \geq 0$. A theoretical optimum for the growth rate can be obtained by the following FBA:

$$\begin{aligned} &\max\ v[r_g] \\ &st.\ \ S \cdot v = 0 \\ &\qquad L \leq v \leq U \end{aligned} \tag{2.2}$$

The maximum $v[r_g]$ obtained by the above LPP (2.2) will be denoted μ_{max}.

2.1.4 Flux Variability Analysis

Flux Variability Analysis (FVA) [35] computes the minimum and maximum fluxes of reactions that are compatible with a given state. For instance, FVA can be used to compute the fluxes that are compatible with a growth $\gamma \cdot \mu_{max}$ where $\gamma \in [0, 1]$. FVA can be computed by solving two independent LPPs per reaction $r \in \mathcal{R}$. One programming problem maximises $v[r]$, and the other minimises $v[r]$. The constraints of both problems are the same: the steady state condition $S \cdot v = 0$, the flux bounds $L \leq v \leq U$, and the constraint $\gamma \cdot \mu_{max} \leq v[r_g]$. The two

LPPs for a given $r \in \mathcal{R}$ can be expressed as:

$$\begin{aligned} \max / \min\ & v[r] \\ st.\ \ & S \cdot v = 0 \\ & L \leq v \leq U \\ & \gamma \cdot \mu_{max} \leq v[r_g] \end{aligned} \tag{2.3}$$

2.2 Vulnerabilities

This section defines two types of reactions that are widely acknowledged as potential drug targets: chokepoint reactions and essential reactions.

2.2.1 Chokepoint reactions

A chokepoint is a reaction that is the only consumer or the only producer of a given metabolite. Thus, the inhibition of a chokepoint may lead to the unlimited accumulation of potentially toxic metabolites or the lack of production of an essential compound. This makes chokepoint reactions appealing drug targets [62].

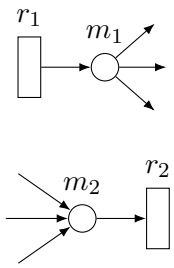


Figure 2.4 Reaction r_1 is the only producer of m_1. Reaction r_2 is the only consumer of m_2.

Formally, chokepoint reactions can be defined as:

Definition 2.2.1. A reaction $r \in \mathcal{R}$ is a *chokepoint* if there exists $m \in \mathcal{M}$ such that $m^\bullet = \{r\}$ or ${}^\bullet m = \{r\}$.

The previous definition only takes into account the structure of the network and disregards the flux bounds of the reactions. In order to capture the fact that reactions can proceed forwards or backwards, e.g. reversible reactions, a flux-based definition of chokepoint reactions is proposed:

Definition 2.2.2. [40] A reaction $r \in \mathcal{R}$ is a *flux-dependent chokepoint* if there exists $m \in \mathcal{M}$ such that $m^\star = \{r\}$ or ${}^\star m = \{r\}$. The set of flux-dependent chokepoints is denoted $CP_\star$.

Example 2.2.1. In the Petri net in Figure 2.3, r_1 is a flux-dependent producer of m_1, i.e. $r_1 \in {}^\star m_1$; Since r_1 is the only flux-dependent producer of m_1, r_1 is a flux-dependent chokepoint, i.e. $m_1{}^\star = \{r_1\}$ and $r_1 \in CP_\star$.

2.2.2 Essential reactions

A reaction is said to be essential if it is required by the organism to grow. In other words, the deletion of an essential reaction implies null growth. Consequently, these reactions have the potential to cause the death of the modelled organism [45].

Definition 2.2.3. [45] A reaction $r \in \mathcal{R}$ is an *essential reaction* if the solution of the following LPP:

$$\begin{aligned} &\max\ v[r_g] \\ &st.\ \ S \cdot v = 0 \\ &\qquad L \leq v \leq U \\ &\qquad v[r] = 0 \end{aligned} \tag{2.4}$$

is equal to 0 or the LPP is infeasible.

The set of essential reactions, which is denoted ER, can be computed straightforwardly by solving (2.4) for each $r \in \mathcal{R}$.

Example 2.2.2. Consider now the Petri net of Figure 2.3, where reaction r_g represents growth. Here reactions r_1, r_4 and r_8 are essential reactions. This is because, if the flux of any of these reactions is set to 0, then it is not possible to produce metabolites m_2, m_4, m_6 respectively, which are necessary for the growth reaction r_g.

In Appendix C, a couple of examples from the literature have been included that show how chokepoint reactions and essential reactions are exploited in drug discovery pipelines.

Chapter 3

Sustaining growth

In the previous chapter, flux-dependent definitions of reactions were introduced. These reaction sets however are not fixed. Depending on the environment in which microorganisms are located, and the nutrients available in the medium, flux might vary considerably in metabolism. The aim of this chapter is to explore the question: *What happens to the vulnerabilities identified when a model is forced to produce a certain growth rate?*. The chapter presents how growth constraints can be incorporated into the identification of vulnerabilities. The resulting vulnerabilities are considered *growth-dependent*.

3.1 Growth-dependent definitions

3.1.1 Essential reactions

Similarly to essential reactions, which are those reactions that are necessary to produce non-null growth on the model, growth-dependent essential reactions are those reactions that are necessary to produce a certain growth on the model. This certain growth will be expressed as $\gamma \cdot \mu_{max}$ where $\gamma \in [0, 1]$ and μ_{max} is the solution of (2.2).

A reaction is said to be a growth-dependent essential reaction for a given growth $\gamma \cdot \mu_{max}$ if its deletion implies that the maximum possible growth is below $\gamma \cdot \mu_{max}$. More formally,

Definition 3.1.1. Let μ_{max} be the solution of the LPP in (2.2). Given $\gamma \in [0, 1]$, a reaction $r \in \mathcal{R}$ is a *growth-dependent essential reaction* if the solution of LPP (2.4) is lower than $\gamma \cdot \mu_{max}$ or the LPP is infeasible.

The set of growth-dependent essential reactions for a given growth specified by $\gamma \in [0, 1]$ will be denoted ER_γ. This set can be computed straightforwardly by solving LPP (2.4) for each reaction.

This chapter includes results from the preprint article:
A. Oarga, B. P. Bannerman and J. Júlvez. CONTRABASS: Exploiting flux constraints in genome-scale models for the detection of vulnerabilities. Submitted to the journal Bioinformatics.

Special attention is given to the set of reactions ER_1, as it will consist of those reactions that are necessary to produce the optimum growth of the model. This set will be named *essential reactions for optimum growth* (EROG).

Example 3.1.1. In the Petri net of Figure 2.3 where r_g models growth, reactions r_1, r_2, r_3, r_4 are essential reactions for optimum growth (i.e. $r_1, r_4, r_6, r_8 \in EROG$). Reactions r_1, r_4 and r_8 are essential reactions, and thus, they are also essential reactions for optimum growth. Regarding reaction r_6, if this reaction is forced to have flux equal to 0, metabolite m_4, which is essential for growth, can only be produced through a non-optimal path. Then, the model will not be able to achieve optimum growth, thus making these reactions, essential reactions for optimum growth.

3.1.2 Dead, reversible and chokepoint reactions

The computation of the flux bounds by means of FVA (2.3), can be carried out in an optimum state i.e. $\gamma = 1$, or in a suboptimal state i.e. $0 \leq \gamma < 1$. In the optimal state, all fluxes must be optimally directed towards growth, whereas in suboptimal states, fluxes are allowed to deviate towards other functionalities.

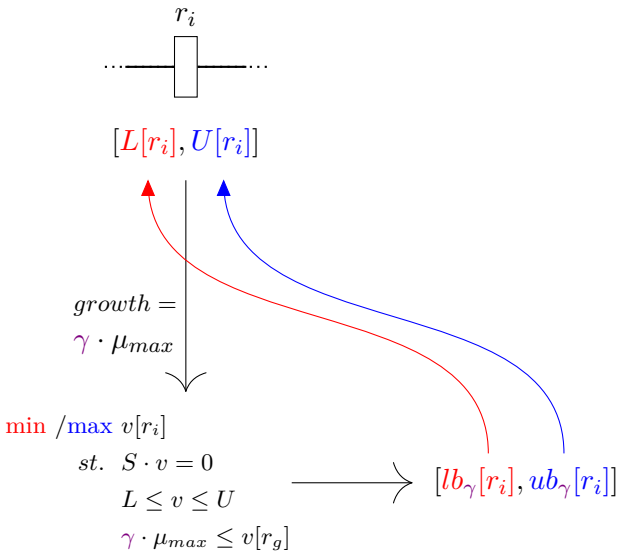


Figure 3.1 Procedure for turning reaction r_i into a growth-dependent reaction. FVA is computed for a growth specified by γ and the initial flux bounds $[L[r_i], U[r_i]]$ are replaced with FVA minimum and maximum bounds $[lb_\gamma[r_i], ub_\gamma[r_i]]$.

Let $lb_\gamma, ub_\gamma \in \mathbb{R}^{|\mathcal{R}|}$ be the result of computing FVA (2.3) on a constraint-based model $\{\mathcal{R},\ \mathcal{M},\ \mathcal{S},\ L,\ U\}$ for a given γ, i.e. $lb_\gamma[r]$ and $ub_\gamma[r]$ are the minimum and maximum fluxes given by FVA for reaction r. If the flux bounds L, U of the constrained-based model are replaced by lb_γ, ub_γ (as depicted in Figure 3.1), a new constraint-based model, $\{\mathcal{R},\ \mathcal{M},\ \mathcal{S},\ lb_\gamma,\ ub_\gamma\}$, with more restrictive (and realistic) flux bounds is obtained.

Given $\gamma \in [0, 1]$, the sets of *growth-dependent products, reactants, consumers, and producers* of the model $\{\mathcal{R},\ \mathcal{M},\ \mathcal{S},\ L,\ U\}$, which are denoted $r^\gamma, {}^\gamma r, m^\gamma, {}^\gamma m$ respectively, are defined as the flux-dependent products, reactants, consumers and products of $\{\mathcal{R},\ \mathcal{M},\ \mathcal{S},\ lb_\gamma,\ ub_\gamma\}$ as discussed in Subsection 2.1.2.

Similarly, given $\{\mathcal{R},\ \mathcal{M},\ \mathcal{S},\ L,\ U\}$ and $\gamma \in [0, 1]$, we can define sets of *growth-dependent dead, reversible and non-reversible* reactions, which are denoted DR_γ, RR_γ and NR_γ, the

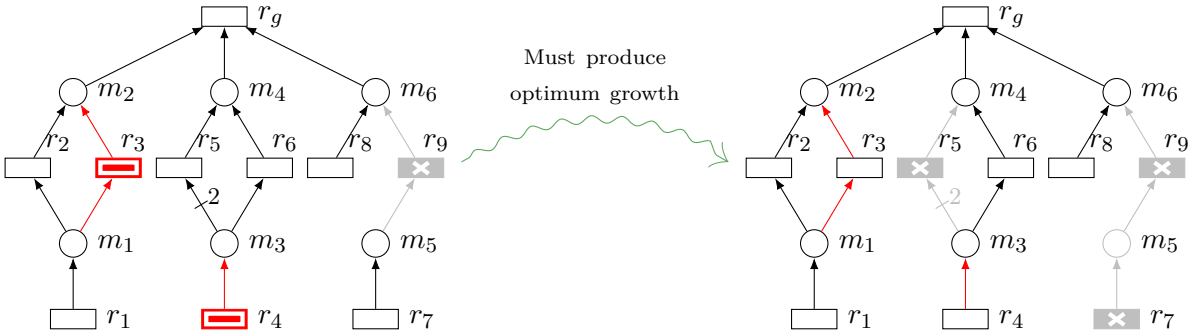


Figure 3.2 Effects that growth-constraints have on the reactions of a constraint-based. When the optimum growth constraint $\gamma = 1$ is imposed on the model (left), a new model with new flux (and hence directionality) is obtained (right).

sets of *growth-dependent chokepoint reactions*, which are denoted as CP_γ, are also defined as the corresponding flux-dependent elements of $\{\mathcal{R},\ \mathcal{M},\ \mathcal{S},\ lb_\gamma,\ ub_\gamma\}$. Chapter 7 shows how growth-dependent sets vary with growth in a real GEM of *Plasmodium Falciparum* bacteria.

Example 3.1.2. Let us assume that r_g in the model on the left of Figure 3.2 represents growth and that the flux bounds of the reactions are as defined in Example 2.1.2. In order to obtain growth-dependent sets, the flux bounds computed by FVA with $\gamma = 1$ are assigned to the reactions and the net on the right in 3.2 is obtained. In this new net, r_5, r_7 and r_9 are dead reactions, i.e. $r_5, r_7, r_9 \in DR_1$, and r_3, r_4, which were reversible reactions, become non-reversible reactions, i.e. $r_3, r_4 \in NR_1$.

3.2 Dead reactions and growth

When applying the previous computation on different models, one can notice that the set of growth-dependent dead reactions DR_γ, always follows a pattern similar to the one shown in Figure 3.3, this is, the size is always constant in suboptimal states (i.e. with $\gamma \in (0, 1)$).

The goal of this subsection is to show that the set of growth-dependent dead reactions is the same for any non-null suboptimal state, i.e. for any growth strictly lower than the maximum growth μ_{max} and greater than 0. Such a set coincides with the set of blocked reactions [60], where a reaction $r \in \mathcal{R}$ is said to be blocked if its flux is 0 at any possible steady state. More formally:

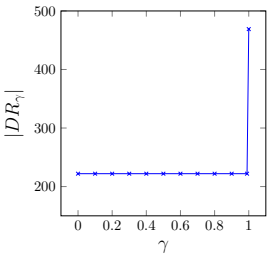


Figure 3.3 Size of $|DR_\gamma|$ in model iAM-Pf480 of *Plasmodium falciparum*.

Definition 3.2.1. A reaction $r \in \mathcal{R}$ is a blocked reaction if for every $v \in \mathbb{R}^{|\mathcal{R}|}$ such that $S \cdot v = 0$, it holds $v[r] = 0$.

Example 3.2.1. In the Petri net in Figure 3.4, reaction r_1 is a blocked reaction. This is because r_1 consumes m_1, which is a metabolite not produced by any reactions. If the flux of r_1 were positive(negative), the amount of m_1 would decrease(increase) indefinitely, which contradicts the steady-state constraint, therefore the only possible steady-state flux for r_1 is $v[r_1] = 0$.

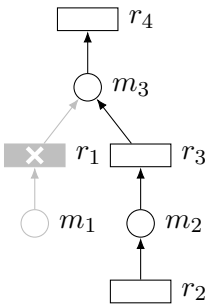


Figure 3.4 Reaction r_1 is a blocked reaction because there is no producer for m_1.

In [60] blocked reactions are obtained by solving the following linear programming problems that compute the maximum and minimum feasible fluxes of the reactions subject to the steady state constraint $S \cdot v = 0$:

$$\begin{aligned} \max / \min \quad & v[r] \\ st. \quad & S \cdot v = 0 \\ & L \leq v \leq U \end{aligned} \tag{3.1}$$

A reaction with null maximum and minimum feasible flux is a blocked reaction. Note that this procedure is equivalent to computing FVA, see (2.3), with $\gamma = 0$. Thus, the set of blocked reactions is equal to DR_0.

Interestingly, the set of dead reactions in suboptimal states, regardless of the growth rate imposed on the model, is equivalent to the set of blocked reactions. This fact will be proved through several steps. Let us first prove that the range of feasible fluxes of a reaction r, i.e. the interval $[lb_\gamma[r], ub_\gamma[r]]$, cannot increase as γ increases.

Lemma 3.2.1. $[lb_{\gamma_2}[r], ub_{\gamma_2}[r]] \subseteq [lb_{\gamma_1}[r], ub_{\gamma_1}[r]]$ $\forall$ $r \in \mathcal{R} \wedge \forall$ γ_1, γ_2 *such that* $0 \leq \gamma_1 < \gamma_2 \leq 1$.

See proof on page 45.

Then, the set of growth-dependent dead reactions cannot decrease with γ:

Lemma 3.2.2. $DR_{\gamma_1} \subseteq DR_{\gamma_2}$ $\forall$ γ_1, γ_2 *such that* $0 \leq \gamma_1 < \gamma_2 \leq 1$.

See proof on page 45.

Let us now show that the set of growth-dependent dead reactions cannot increase with γ in suboptimal states, i.e. with $\gamma < 1$:

Lemma 3.2.3. $DR_{\gamma_1} \supseteq DR_{\gamma_2}$ $\forall$ γ_1, γ_2 *such that* $0 \leq \gamma_1 < \gamma_2 < 1$.

See proof on page 45.

From Lemmas 3.2.2 and 3.2.3, the following theorem can be derived straightforwardly:

Theorem 3.2.4. $DR_{\gamma_1} = DR_{\gamma_2}$ $\forall \gamma_1, \gamma_2 \in [0, 1)$.

Thus, in particular, the set of blocked reactions coincides with the set of dead reactions in suboptimal states.

Corollary 3.2.5. $DR_\gamma = DR_0$ $\forall \gamma \in [0, 1)$.

Notice that DR_1 can be strictly greater than DR_γ with $\gamma \in [0, 1)$. An example is shown in Chapter 7.

Chapter 4

Enforcing growth, one reaction at a time

In Section 3.1.2 we were able to obtain growth-dependent vulnerabilities by substituting the flux bounds of the model (L, U) with the ones obtained with FVA in an optimum state (lb_1, ub_1). However one can experimentally check that, if optimum growth FVA flux bounds are substituted on a model, the model always produces the optimum growth. A question then arises: *does imposing FVA flux bounds individually on each reaction actually forces the whole model to produce optimum growth?* The aim of this chapter is to formally prove that the answer to this question is *"yes"*.

4.1 Unbounded homogeneous case

Let us first consider the following LPP:

$$\begin{aligned} \max \quad & c^T x \\ st. \quad & A \cdot x = 0 \\ & x \geq 0 \end{aligned} \tag{4.1}$$

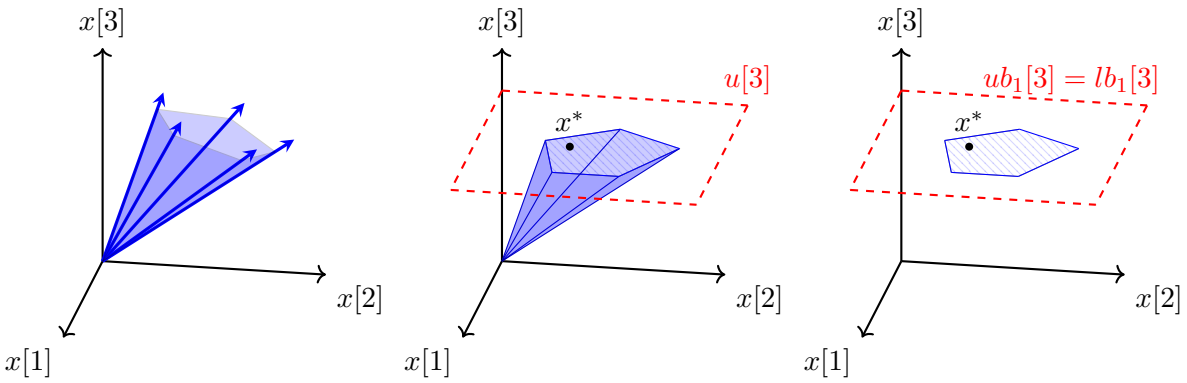
Lemma 4.1.1. *Let H be any hyperplane that delimits the convex cone solution space of the LPP in* (4.1)*. Let x^* denote an optimal solution defined by the LPP. If any x^* is located exclusively in H, then for all x solution of the LPP is 0.*

See proof on page 45.

Lemma 4.1.2. *Having $A \cdot x = 0$ and $x \geq 0$, if the LPP has at least one solution and max $c^T x > 0$, then the LPP in* (4.1) *is unbounded.*

See proof on page 46.

Notice that the solution space of (4.1) defines a convex cone as the one in Figure 4.1a. Lemma 4.1.2 implies that any non-null solution is located in the interior of the cone.



(a) Cone defined representing the solution space of LPP (4.1).

(b) Convex cone representing the solution space of LPP (4.2).

(c) Convex face solution space obtained from imposing $l = lb_1$ and $u = ub_1$ on the LPP (4.2).

Figure 4.1 Graphical proof: If we impose bounding box constraints on an unbounded convex cone, and then constrain all the solutions to an optimal face, then the solution space becomes exclusively the optimal solution space.

4.2 Bounded homogeneous case

Let us now consider the following LPP:

$$\begin{aligned} \max \quad & c^T x \\ st. \quad & A \cdot x = 0 \\ & 0 \leq l \leq x \leq u \\ & x \geq 0 \end{aligned} \tag{4.2}$$

Proposition 4.2.1. *Given the unbounded LPP of* (4.1) *with max* $c^T x > 0$*, if we include the orthogonal constraints* $0 \leq l \leq x \leq u$*, then with the resulting LPP,* $\exists i \in [1, n]$ *such that that* $lb_1[i] = ub_1[i] = u[i]$ *or* $lb_1[i] = ub_1[i] = l[i]$*.*

See proof on page 46.

Continuing with our graphical example, if we impose bounding box constraints on (4.1), we obtain a cropped convex cone as the one in Figure 4.1b. Proposition 4.2.1 implies that the optimal solution now has to be located at one of the new faces imposed.

Proposition 4.2.2. *Let* $\mu_{max} > 0$ *be the solution of* (4.2)*. Given the LPP in* (4.2)*, if we have* $l = lb_1$ *and* $u = ub_1$*, then max* $c^T x = min\ c^T x = \mu_{max}$*.*

See proof on page 46.

Informally speaking, Proposition 4.2.2 means that, if we limit the solution space to a certain face with optimal solutions, then all the solutions yielded by the new LPP produce the optimum objective value (see Figure 4.1c). With this, we have shown that, if we impose FVA flux bounds on each reaction, then the model is only able the produce the optimal solution.

4.3 Bounded non-homogeneous case

Until now we have been considering LPPs defined by a homogeneous linear system. However, it is also interesting to see that, under certain conditions, the results provided here can be generalised to non-homogeneous linear systems as well. Since this is out of the scope of our initial objective, it is provided separately in Appendix E.

Chapter 5

Minimal microorganisms

Minimum metabolism is the minimum genome necessary for cells to grow and divide [4]. The study of the minimal metabolism has been an appealing subject as it could provide an understanding of evolutionary plasticity, which confers pathogens the ability to evolve their metabolism towards drug-resistant configurations [4]. Besides it could also help in the identification of simplest possibles forms of life [30]. For our work, the computation of the minimum metabolism is relevant as it could provide insights into the robustness of the networks. In this chapter, we study the problem of computing the minimum metabolism.

5.1 Reactions for growth

Before we start, first we will propose definitions of different sets of reactions that are involved in growth, that is, *reactions for growth* and the *minimum set of reactions for optimum growth.*

5.1.1 Reactions for optimum growth

Let $||v||$ denote the support of $v \in \mathbb{R}^{|\mathcal{R}|}$, i.e., $||v|| = \{r \in \mathcal{R} \mid v[r] \neq 0\}$. The *set of reactions for optimum growth* is defined as follows:

Definition 5.1.1. A set of reactions F is *a set of reactions for optimum growth* (ROG) if $\exists v \in \mathbb{R}^{|\mathcal{R}|}$ such that $S \cdot v = 0$, $L \leq v \leq U$, $v[r_g] = \mu_{max}$ and $||v|| = F$.

Notice that set $EROG$, which was introduced in Chapter 3, is a subset of ROG, i.e. $EROG \subseteq ROG$. Moreover, since there can be multiple flux distributions that produce optimum growth, ROG might not be unique. Given that the reactions in a ROG are sufficient to produce optimum growth, the model can produce the optimum growth even if all the reactions in $\mathcal{R} - ROG$ are inhibited.

This chapter includes results from the conference article:
A. Oarga and J. Júlvez. On the computation of the minimum set of reactions for optimal growth in constraint-based models. Accepted at IEEE Conference on Decision and Control 2022 (CDC 2022).

5.1.2 Minimum set of reactions for optimum growth

Let $\mathcal{O}$ be the set of all ROG sets of a model.

Definition 5.1.2. A set of reactions $O_i \in \mathcal{O}$ is a *minimum set of reactions for optimum growth* (MROG) if $|O_i| \leq |O_j| \; \forall O_j \in \mathcal{O}$.

Similarly to *ROG*, the set $MROG$ might not be unique.

Example 5.1.1. The model in Figure 5.1 has 2 feasible $MROG$ sets: $\{r_1, r_2, r_4, r_6, r_8\}$ and $\{r_1, r_3, r_4, r_6, r_8\}$. Metabolite m_2 is necessary for growth and can be equally produced by reactions r_1, r_2 or r_1, r_3. Metabolite m_6 can be produced by various reactions, however, the minimum number of reactions required to produce it optimally is 1 (i.e. r_8), thus r_8 is in $MROG$. Finally, reactions r_1, r_4, r_6 are in $EROG$ and therefore are present in any $MROG$ set.

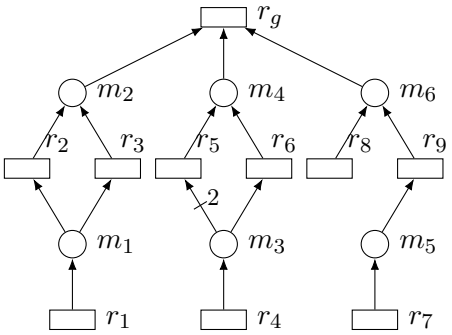


Figure 5.1 Example Petri net modelling a constraint-based model.

5.2 Minimum set of reactions computation

We are ready to formally define the problem of computing a MROG:

Problem 5.2.1. Given a constraint-based model $\{\mathcal{R},\ \mathcal{M},\ \mathcal{S},\ L,\ U\}$, and an objective reaction $r_g \in \mathcal{R}$, the *minimum set of reactions for optimum growth problem* ($MROGP$) is the problem of finding a minimum set of reactions for optimum growth $MROG$.

It will be shown that $MROGP$ can be solved by a Mixed-Integer Linear Programming problem (MILP) where the objective is to minimise the number of reactions required for optimum growth. We will make use of a vector of *initial* fluxes, $w \in \mathbb{R}^{|\mathcal{R}|}$, and a vector of binary variables, $\delta \in \{0,1\}^{|\mathcal{R}|}$, that indicates which fluxes are cancelled out, i.e. $\delta[r] = 0$ implies that there is no flux through r regardless of $w[r]$. Thus, the actual flux of a given reaction, r, is $v[r] = \delta[r] \cdot w[r]$. Let us consider the following programming problem:

$$\begin{aligned} \min \; & \sum_{r \in \mathcal{R}} \delta[r] \\ st. \; & S \cdot v = 0 \\ & v[r] = \delta[r] \cdot w[r] \; \forall \; r \; \mathcal{R} \\ & L \leq w \leq U \\ & v[r_g] = \mu_{max} \end{aligned} \tag{5.1}$$

Given that the number of reactions with non-null flux is minimised, the support of a vector v that is a solution to the programming problem (5.1) is an $MROG$.

Equation $v[r]{=}\delta[r]{\cdot}w[r]$ makes the problem (5.1) non-linear. Such an equation is equivalent to the following inequalities:

$$\begin{aligned} &v[r] \leq U[r]{\cdot}\delta[r] \ \forall r \in \mathcal{R} \\ &v[r] \geq L[r]{\cdot}\delta[r] \ \forall r \in \mathcal{R} \\ &v[r] \leq w[r] - L[r]{\cdot}(1-\delta[r]) \ \forall r \in \mathcal{R} \\ &v[r] \geq w[r] - U[r]{\cdot}(1-\delta[r]) \ \forall r \in \mathcal{R} \end{aligned} \tag{5.2}$$

Thus, the replacement of $v[r]{=}\delta[r] \cdot w[r]$ in (5.1) by the above inequalities results in a MILP which solves MROGP.

5.3 Computational complexity

This section proves that a solution for *MROGP* can not be found in polynomial time. Let us first restate the problem as a decision problem:

Problem 5.3.1. Given a constraint-based model $\{\mathcal{R},\ \mathcal{M},\ \mathcal{S},\ L,\ U\}$, an objective reaction $r_g \in \mathcal{R}$, and integer k, the *set of reactions for optimum growth problem* (*ROGP*) is the problem of determining whether there exists a *ROG* set O_i with $|O_i| \leq k$.

We will prove that *ROGP* is NP-complete. First, it is proved that this problem is in NP.

Lemma 5.3.1. *ROGP is in NP.*

Proof. Given a set of reactions $O_i \subseteq \mathcal{R}$, we can verify that the set is a *ROG* set for a constraint-based model $\{\mathcal{R},\ \mathcal{M},\ \mathcal{S},\ L,\ U\}$ with objective reaction $r_g \in \mathcal{R}$, by removing all reactions not in O_i from the model and solving the LPP in (2.2). If the growth obtained is equal to μ_{max} and $|O_i| \leq k$, then the set O_i is a *ROG* set with size at most k. Since LPPs can be solved in polynomial time [22], *ROGP* is in NP. □

Let us now prove that *ROGP* is NP-hard by reducing the vertex cover problem [23] to *ROGP*.

The vertex cover problem is defined as:

Problem 5.3.2. Given an undirected graph $G = (V, E)$, a vertex cover V' is a subset of V such that $uv \in E \rightarrow u \in V' \vee v \in V'$. The vertex cover problem is the problem of determining whether there exists a vertex cover of size at most k.

Lemma 5.3.2. *ROGP is NP-hard.*

Proof. Let us reduce an instance of the vertex cover problem, consisting of an undirected graph $G = (V, E)$, to a *ROGP*. First, the undirected graph is transformed into a bipartite graph (represented graphically as a Petri net) of reactions and metabolites as follows:

- For each vertex $v_i \in V$ create a *source* reaction r_i with $L[r_i]{=}0, U[r_i]{=}2$.
- For each edge $e_i \in E$ create a metabolite m_i.

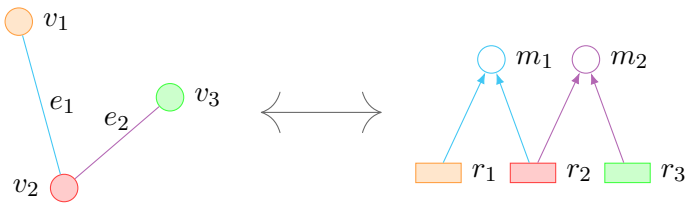


Figure 5.2 Undirected graph with 3 vertices and 2 edges (left). Network of source reactions and metabolites resulted from transforming the undirected graph (right).

- For each adjacent edge e_i of each vertex v_j make the corresponding metabolite m_i a product of the corresponding reaction r_j.

In Figure 5.2 we can see an example of a network $\mathcal{R}$=$\{r_1, r_2, r_3\}$ and $\mathcal{M}$=$\{m_1, m_2\}$, resulting from the transformation of the undirected graph shown in the left with V=$\{v_1, v_2, v_3\}$ and E=$\{e_1, e_2\}$.

In addition to the previous transformations, the following ones are also performed:

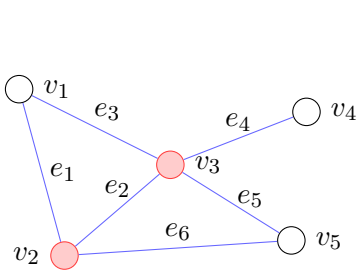
- For each metabolite(edge) $m_i \in \mathcal{M}$ create a *sink* reaction r_j with $|V|$<j≤$|V| + |E|$ with $L[r_j]$=1, $U[r_j]$=$|V|$.
- Add an objective reaction r_g with $L[r_g]$=0, $U[r_g]$=1 that consumes all metabolites $m_i \in \mathcal{M}$.

Figure 5.3b shows the final constraint-based model resulting from applying the described transformation to the graph in Figure 5.3a.

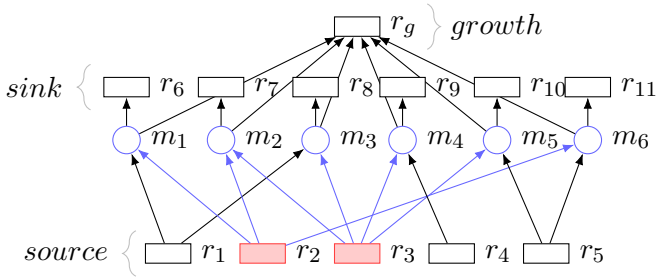
In the obtained constraint-based model, each source reaction r_i will act as an input to the network and sink reactions will balance the potential excess of produced metabolite. Notice that in order to achieve the optimum growth, all the metabolites must be produced, and as long as $|E| > 0$, this model will always be able to produce the maximum growth (i.e. μ_{max}=1) with a certain $v \in \mathbb{R}^{|\mathcal{R}|}$ obtained by the LPP in (2.2).

It can be seen that any *ROG* set will have the following number of reactions: all sink reactions (the number of sink reactions is $|E|$) since all sink reactions are constrained to have non-null flux; the growth reaction; and a number k of reactions, with $1 \leq k \leq |V|$, that correspond to a set of reactions necessary to produce all the metabolites in the model. To summarise, any *ROG* set will have a size equal to: k (source reactions) + $|E|$ (sink reactions) + 1 (growth reaction). The set of k source reactions will be used to derive a solution for the vertex cover problem. Let us prove the following claim: a vertex cover of size k exists if and only if a *ROG* set of size $k + |E| + 1$ exists. We proceed by proving both directions of the claim:

1. If a *ROG* set of size $k + |E| + 1$ exists, then a vertex cover of size k exists: Let $R_{in} \subseteq \mathcal{R}$ be the set of k source reactions of a given *ROG* set. A vertex cover $V' \subseteq V$ of the graph G can be built as follows: $v_i \in V'$ if $r_i \in R_{in}$. Here, a source reaction producing metabolites is considered equivalent to a vertex covering its adjacent edges. If we consider any source reaction $r_i \in R_{in}$, it produces a set of metabolites $m_j, ..., m_k \in \mathcal{M}$ that is equivalent to the set of edges $e_j, ..., e_k \in E$ that would be covered by the corresponding vertex $v_i \in V$. Since the k source reactions produce all metabolites in the model, it is guaranteed that the resulting vertex set V' covers all edges of the graph, thus making V' a vertex cover.



(a) Undirected graph with a vertex cover of size k = 2 highlighted in red.



(b) Constraint-based model resulting from transforming the undirected graph in Figure a). Input reactions mapped from the vertex cover of Figure a) are highlighted in red. Arcs and metabolites, highlighted in blue, show that reactions r_2 and r_3 are able to produce all the metabolites of the model.

Figure 5.3 Vertex cover problem transformation to MROGP.

2. If a vertex cover of size k exists, then a *ROG* set of size $k+|E|+1$ exists: let $V' \subseteq V$ be a vertex cover with $|V'| = k$. Since all edges of the graph are covered by k vertices and given the equivalence between source reactions and vertices, it is guaranteed that the corresponding k source reactions of the model are sufficient to produce all metabolites of the model and achieve optimum growth, hence a *ROG* of size at most $k + |E| + 1$ exists.

Figure 5.3a shows highlighted in red a vertex cover V' of size $k = 2$ (i.e. $V' = \{v_2, v_3\}$) of the undirected graph. Figure 5.3b shows that the corresponding source reactions r_2, r_3 (highlighted in red) are able to produce all metabolites of the model (highlighted in blue). Therefore, there exists a *ROG* set of size at least $k + |E| + 1$ with $k = 2$ and $|E| = 6$.

Clearly, the same reasoning can be applied reversely to obtain a vertex cover from the *ROG* set. Given the k source reactions of the *ROG* set (e.g. r_2, r_3 in Figure 5.3b), the corresponding vertices are guaranteed to be a vertex cover (e.g. v_2, v_3 in Figure 5.3a). □

The following theorem is derived straightforwardly from Lemmas 5.3.1 and 5.3.2.

Theorem 5.3.3. *Reactions for Optimum Growth Problem (ROGP) is in NP-complete.*

5.4 Problem size reduction

Given that *ROGP* is NP-complete, we can not expect to find a solution in polynomial time, and consequently, we can not expect to solve *MROGP* in polynomial time either. It is possible, however, to achieve a reduction in the size of *MROGP* and along with that a potential reduction in the execution time required to solve it.

As mentioned previously, for any set *ROG* it holds that $EROG \subseteq ROG$. Consequently, the search space for reactions in *MROG* can be reduced from $\mathcal{R}$ to $\mathcal{R} - EROG$. Similarly, the set DR_1 contains all reactions whose only compatible flux with optimum growth is the null flux, hence, this set can also be ignored in the search for *MROG*. The search space of the problem is then reduced from $|\mathcal{R}|$ to $|\mathcal{R} - EROG - DR_1|$.

Note that the MILP in (5.1) had $|\mathcal{R}|$ binary variables, and the above reasoning reduces the number of binary variables to $|\mathcal{R} - EROG - DR_1|$. The resulting *reduced* MILP is:

$$\begin{aligned} \min \ & \sum_{r \in F} \delta[r] \\ st. \ & S \cdot v = 0 \\ & v[r] = \delta[r] \cdot w[r] \ \forall r \in F \\ & v[r] = w[r] \ \forall r \in EROG \\ & L \leq w \leq U \\ & v[r_g] = \mu_{max} \end{aligned} \tag{5.3}$$

where $F = \mathcal{R} - EROG - DR_1$.

Chapter 6

What makes reactions essential anyway?

Despite the significant success that *in-silico* knock-outs have shown regarding essentiality detection [42, 45], the root cause of essentiality has not been cautiously studied, at least from a constraint modelling perspective. Although we are not able yet to answer the question *What makes reactions essential?*, we expect that, through the tools, definitions and procedures exposed in this work, we will be able to shed light on the root cause that makes certain reactions essential. First, we need to introduce a couple more definitions based on flux bounds: forced reactions and knockable reactions.

6.1 Reactions sets

In this subsection we briefly introduce forced reactions and knockable reactions:

Definition 6.1.1. A reaction $r \in \mathcal{R}$ is *forced* if $0 \notin [L[r], U[r]]$.

Definition 6.1.2. A reaction $r \in \mathcal{R}$ is *knockable* if $0 \in [L[r], U[r]] \wedge L[r] < U[r]$.

Sets of forced and knockable reactions are denoted FR and KR respectively.

Sets of *growth-dependent forced and knockable reactions* can be obtained as explained in Section 3.1.2. These sets are denoted FR_γ and KR_γ respectively. By definition, growth-dependent knockable reactions are those for which having a zero flux is compatible with growth. Similarly, growth-dependent forced reactions are those reactions that necessarily need to carry flux for the model to produce a given growth.

6.2 Flux variability and essentiality

We can prove now that flux variability can be used to identify essential reactions and vice-versa. We assume that all reactions initially have loose flux bounds, this is, all reactions are initially knockable reactions.

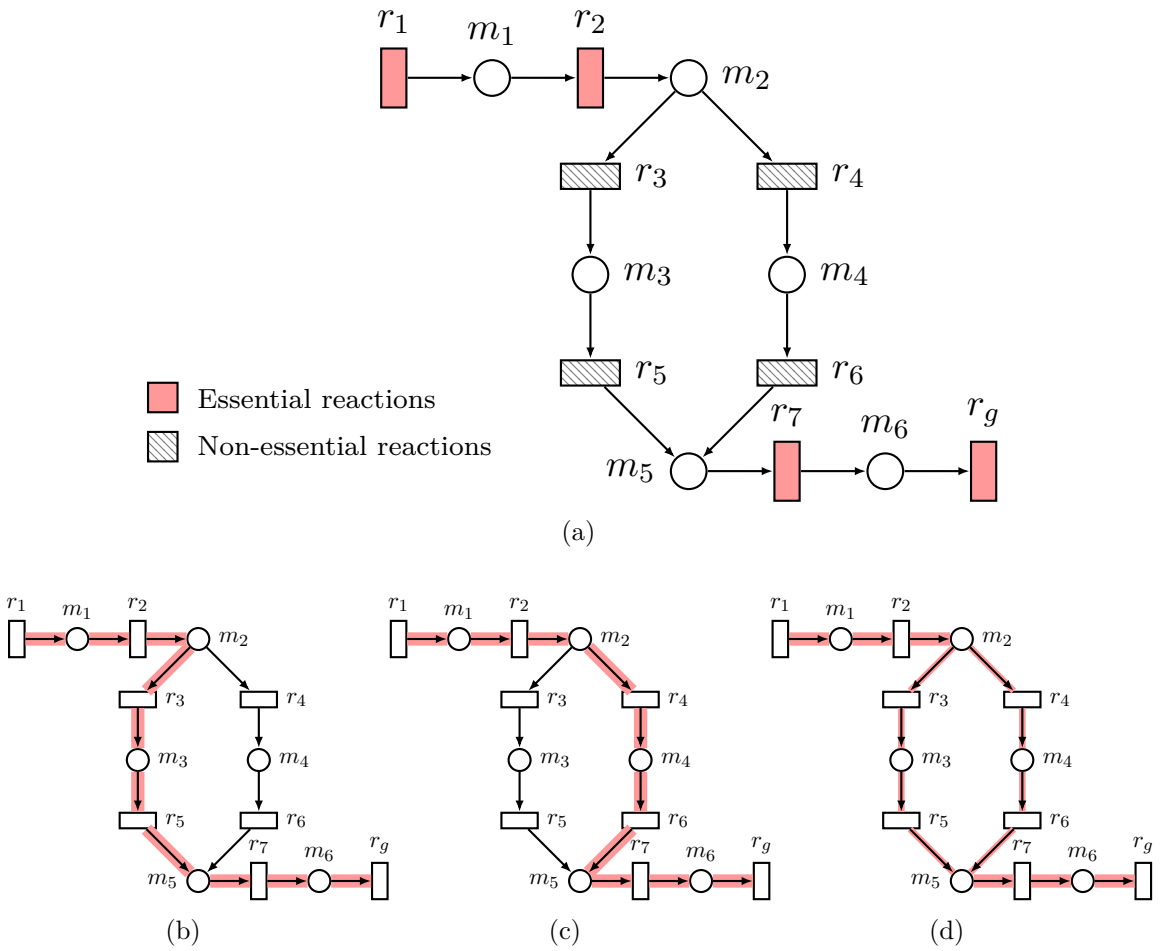


Figure 6.1 Example constraint-based model showing how flux variability in the model conditions essentiality.

Proposition 6.2.1. *Assuming* $r \in KR\ \forall r \in \mathcal{R}$*, given* $\gamma \in [0,1]$ *then* $FR_\gamma = ER_\gamma$.

See proof on page 46.

Proposition 6.2.2. *Assuming* $r \in KR\ \forall r \in \mathcal{R}$*, given* $\gamma \in [0,1]$ *then* $KR_\gamma = \mathcal{R} - ER_\gamma - DR_\gamma$.

See proof on page 47.

Example 6.2.1. Figure 6.1a depicts a constraint-based model where r_g models growth. Figures 6.1b, 6.1c and 6.1c depict 3 feasible configurations by which a certain growth given by γ can be obtained in the model. In these Figures, red lines indicate the distribution of flux through the network, with the lines' height being proportional to the amount of flux carried. Reactions without red lines are considered to carry no flux (e.g. reaction $r4$ in Figure 6.1b does not have flux).

In Figure 6.1a reactions $r3, r4, r5, r6$ are growth-dependent knockable reactions (i.e. $r3, r4, r5, r6 \in KR_\gamma$). It can be seen in Figures 6.1b and 6.1c that there are flux configurations where these reactions have zero flux, but the model is still able to produce the given growth.

As a consequence, these reactions also are not growth-dependent essential reactions (i.e. $r3, r4, r5, r6 \notin ER_\gamma$). On the other hand, reactions $r1, r2, r7$ always need to carry flux to produce growth. These reactions won't allow loose fluxes or null fluxes because they must always carry flux for the model to produce growth. Hence according to flux variability, these are growth-dependent forced reactions (i.e. $r1, r2, r7 \in FR_\gamma$). This also makes these reactions growth-dependent essential reactions (i.e. $r1, r2, r7 \in ER_\gamma$).

From the previous propositions, we can see that reaction essentiality can be identified from flux variability and vice-versa. Since variability provides information about the essentiality of each reaction we can leverage this to hypothesise about the properties of essential reactions. In [40], we showed that, in synthetic models, reactions in alternative paths belonged to the set of knockable reactions. In addition, in this work, we have seen that growth itself is produced through a combination of essential reactions that are always mandatory and a subset of optional reactions chosen from the set of knockable reactions. We have therefore reasons to believe that knockable reactions provide redundancies (or robustness) in the production of biomass, which makes them immune to individual knock-outs. On the other hand, essential reactions contribute to the production of substrates that are not produced through other reactions, which totally impedes growth when these types of reactions are knocked out individually.

From a biological perspective, redundancy in metabolism has also been acknowledged as a mechanism for robustness in yeast metabolism, through the means of isoenzymes [5]. Besides, yeast metabolism robustness is believed to be an evolutionary result towards more metabolic efficiency or robustness under specific environmental conditions [9, 46].

Chapter 7

Case study: plasmodium falciparum

This chapter presents the results obtained for the constraint-based model of *Plasmodium falciparum* (iAM-Pf480) [1]). The models include a total of 1083 reactions, 909 metabolites, and 480 genes. The sizes of the sets of flux-dependent reactions are $|RR| = 493, |NR| = 590, |DR| = 0$.

7.1 Growth-dependent reactions

Figure 7.1 shows the sizes of the growth-dependent sets NR_γ, RR_γ and DR_γ. To assess the impact of γ in these sets, different values of γ in the interval $[0, 1]$ have been used. In addition to the sizes of the sets obtained with γ, the leftmost value (depicted in red) of plots refers (from top to bottom plot) to the size of the flux-dependent set prior to FVA. i.e. NR, RR and DR.

Recall that dead reactions were already studied in Section 3.2 and a formal reason for dead reactions progression with growth was given. Furthermore, the increase in the set of dead reactions that takes place at $\gamma = 1$ is due to the fact that, in the optimal growth state, the flux must be necessarily distributed through optimal paths for biomass production and no flux can be diverted through other paths. Thus, non-optimal reactions for biomass production become dead reactions.

With respect to reversible reactions, the steady state constraint $S \cdot v = 0$ reduce the size of this set from $|RR| = 493$ to $|RR_0| = 210$. Such a reduction is caused by the blocked reactions that belonged to RR and become dead reactions, and by the reversible reactions that become non-reversible with the steady-state constraint.

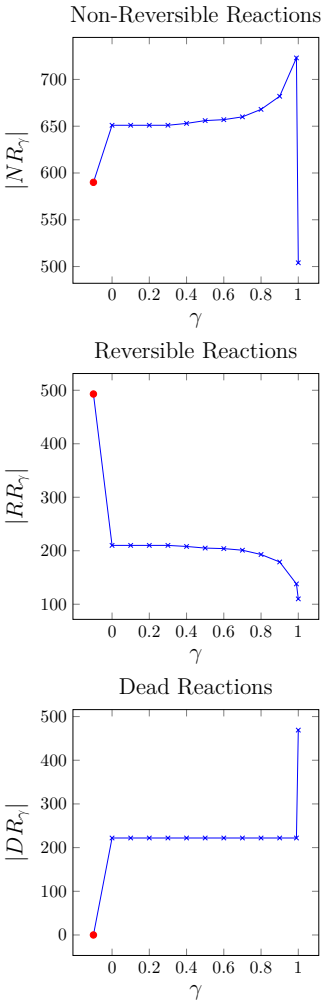


Figure 7.1 Size of $|NR_\gamma|$, $|RR_\gamma|$ and $|DR_\gamma|$ in model iAM-Pf480.

A considerable amount of non-reversible reactions turn into blocked reactions (dead reactions) with the steady-state constraint. However, due to the significant amount of reversible reactions that become non-reversible reactions, the size of the set is increased from $|NR| = 590$ to $|NR_0| = 651$.

7.1.1 Vulnerabilities

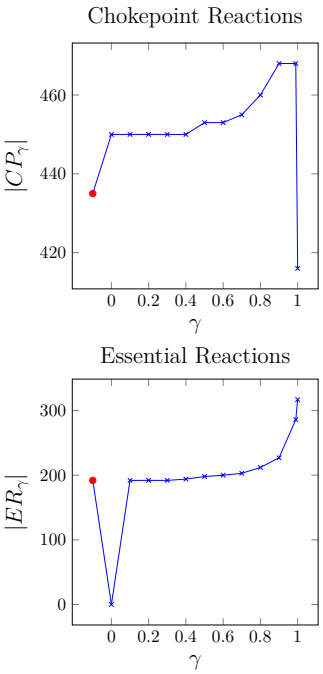


Figure 7.2 Size of $|CP_\gamma|$ and $|ER_\gamma|$ in model iAM-Pf480 of *Plasmodium falciparum*.

The number of initial flux-dependent chokepoint reactions is $|CP| = 453$. The sets of essential reactions and essential reactions for optimal growth were also computed, and the following sizes were obtained: $|ER| = 192$ and $|EROG| = 372$.

Notice that if $\gamma = 0$, the constraint $\gamma \cdot \mu_{max} \leq z \cdot v$ in (2.3) does not impose a minimum growth on the model, and only the steady state condition $S \cdot v = 0$ must be satisfied.

Concerning chokepoint reactions, see Figure 7.2, the number of flux-dependent chokepoints is $|CP_\star| = 453$ (reported in red in the leftmost part of the Figure). At $\gamma = 0$ there is an increase to $|CP_0| = 450$, and then the set increases slowly until $|CP_{0.99}| = 468$. As in the sets of non-reversible reactions and reversible reactions, the set of chokepoints decreases at $\gamma = 1$ as many reactions become dead reactions.

Notice that the set of chokepoints at $\gamma = 1$, CP_1, is smaller than the set of flux-dependent chokepoints, $CP_\star$. Moreover, CP_1 is not contained in $CP_\star$. This is due to the changes produced in the sets of non-reversible reactions and reversible reactions as γ increases.

Figure 7.2 also reports the the size of the set of essential reactions ER_γ with respect to γ. The leftmost value of this graph refers to the ER set, i.e. the set of essential reactions with no growth constraints.

Notice that no reaction is mandatory to produce a null growth, hence at $\gamma = 0$, $|ER_0|$ is always 0. Furthermore, for positive values of γ the size of ER_γ increases as the size of RR_γ decreases. This is because, as γ increases, some reversible reactions become non-reversible, and the flux of these reactions is necessary, i.e. essential for growth. On the other hand, as expected, the amount of growth-dependent essential reactions increases with γ.

7.2 Robustness and minimal metabolism

Finally, Figure 7.3 shows the size of the set KR_γ with respect to γ, with the leftmost point corresponding to $\mathcal{R} - ER - DR$. As mentioned in Chapter 6, this set is of interest as it is composed of those reactions that can contribute to growth, that is, they are not dead reactions, and at the same time are not growth-dependent essential reactions, that is, if one of these reactions is knocked out the growth can still be kept the same. Notice that this does not mean that all the reactions in this set can be knocked simultaneously without

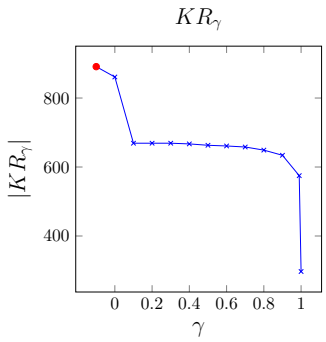


Figure 7.3 Size of $|KR_\gamma|$ in model iAM-Pf480 of *Plasmodium falciparum*.

affecting growth. As discussed, the presence of these reactions might then imply the presence of redundancy in the metabolism. Therefore, this set of reactions provides the metabolism with resilience and flexibility. As it can be seen, the size of this set decreases with γ. This fact means that producing higher growths is more demanding as it requires a higher amount of reactions to sustain it. In other words, the higher the growth the more reactions are necessary to produce such growth, and therefore, less flexibility is given to the metabolism.

Regarding optimum growth production, a ROG set with $|ROG| = 486$ was computed with FBA. We also had $|KR_1| = 242$. The $MROG$ computed with the procedure explained in Chapter 5 had a size $|MROG| = 478$. This is, the $MROG$ set has 8 reactions less than a ROG set obtained with FBA. The size of this set represents 45% of all model reactions (478 out of 1083). As it has been explained, this $MROG$ set is composed of the 372 reactions from the $EROG$ set plus 106 reactions out of the total 242 reactions of KR_1.

Appendix D provides a visualisation of the sets $EROG$ and KR_1 of iAM-Pf480 model and how these are combined to form the $MROG$ set.

Chapter 8

Conclusions and Future Work

8.1 Conclusion

Computational methods have the potential to provide a cost-effective alternative to traditional screening methods for drug discovery. In this master thesis, we aimed to provide computational methods to accurately identify critical reactions in metabolism. These reactions can serve as drug targets when applied to models of pathogenic bacteria. At the same time, we also aimed to understand the mechanisms that confer robustness to bacteria metabolism.

In this work, we have formally defined types of reactions in metabolism (i.e. reversible reactions, non-reversible, and dead reactions) and vulnerable reactions (i.e. chokepoint and essential reactions). We have provided a novel method to identify growth-dependent vulnerable reactions consistent with the production of growth in the modelled microorganisms. On a general homogeneous LPP, we have proved that growth-dependent reactions do indeed force the model to produce the optimum growth. Furthermore, we also formally prove that the sets of growth-dependent dead reactions are fixed in suboptimal growth states.

We have also studied how growth is produced in metabolism. We provided methods to identify sets of reactions essential for growth and sets of redundant reactions that account for the robustness of metabolism. We showed that growth involves a combination of both sets. All this helps to break the black-box conception of growth in constraint-based models. We have also shown that finding a minimum set of reactions that supports optimal growth is computationally complex and a method to optimise this procedure has been proposed. Finally, we have shown that flux variability and essentiality are coupled and proposed redundancy as one mechanism behind non-essentiality.

The vulnerabilities proposed in this work have been identified on a genome-scale model of *Plasmodium Falciparum*. We have seen that growth-constrained vulnerabilities produce novel vulnerabilities that are different from the ones obtained when such constraints are neglected. Finally, we have used our proposed method to compute the minimum metabolism necessary for optimum growth in a GEM of *Plasmodium Falciparum*.

8.2 Future work

The contributions of this work have been possible thanks to the large amounts of data that technological advances have produced. When omic data was first available, the integration was merely prohibitive except for GEMs. However, this inability to integrate all the available data means that these models only reflect a partial understanding of the modelled system [64].

To tackle this issue, current approaches in the field are moving towards the integration of multiple heterogeneous data sources into GEMs, and this is a trend expect to be maintained in the forthcoming years [13, 10]. Future approaches need therefore to move towards this direction as it will lead to more accurate predictions on models. For instance, recent publications show that the identification of essential reactions can be significantly improved by integrating constraints into the model such as thermodynamic [14] or transcriptomic constraints [42].

Current modelling approaches also face another burden. Modelling solutions usually depend on the availability of certain types of data (e.g. kinetic parameters) which might not be available [21]. Future approaches should focus on extending current modelling solutions. This might include being able to deal with uncertainty or being able to estimate the lacking data [21, 64].

At the same time as this field develops, current data-driven approaches w.r.t. machine learning modelling approaches have had sounding success in biological modelling. As of 2022, these methods have already successfully estimated k_{cat} kinetic constraints parameters directly from SMILES sequence [31] or Michaelis k_M constants from structural data [28]. It is expected that data-driven modelling approaches will bring substantial contributions to the data-integrative development of the field [2, 24, 64].

The proposal of machine learning procedures is out of the scope of this work, however, the curious reader could find a first proposal of a hybrid model/data-driven approach in Appendix F. In this Appendix, we show how classical Petri Nets models can be leveraged to model biological networks through neural networks.

Nomenclature

Greek Symbols

γ Fraction of optimum growth

μ_{max} Optimum growth

Acronyms / Abbreviations

AMR Antimicrobial Resistance

CP ChokePoint reacctions

DR Dead Reactions

ER Essential Reactions

FBA Flux Balance Analysis

FR Forced Reactions

FVA Flux Variability Analysis

GEM Genome Scale Models

KR Knockable Reactions

LPP Linear Programming Problem

MDR Multi-Drug Resistant bacteria

$MROG$ Minimum set of Reactions for Optimum Growth

$MROGP$ Minimum set of Reactions for Optimum Growth Problem

NR Non Reversible reactions

ROG Reactions for Optimum Growth

$ROGP$ Reactions for Optimum Growth Problem

EROG Essential Reactions for Optimum Growth

RR Reversible Reactions

List of Figures

Bibliography

[1] Abdel-Haleem, A. M., Hefzi, H., Mineta, K., Gao, X., Gojobori, T., Palsson, B. O., Lewis, N. E., and Jamshidi, N. (2018). Functional interrogation of plasmodium genus metabolism identifies species-and stage-specific differences in nutrient essentiality and drug targeting. *PLoS computational biology*, 14(1):e1005895.

[2] Antonakoudis, A., Barbosa, R., Kotidis, P., and Kontoravdi, C. (2020). The era of big data: Genome-scale modelling meets machine learning. *Computational and structural biotechnology journal*, 18:3287–3300.

[3] Bannerman, B. P., Júlvez, J., Oarga, A., Blundell, T. L., Moreno, P., and Floto, R. A. (2021). Integrated human/sars-cov-2 metabolic models present novel treatment strategies against covid-19. *Life science alliance*, 4(10).

[4] Barve, A., Rodrigues, J. F. M., and Wagner, A. (2012). Superessential reactions in metabolic networks. *Proceedings of the National Academy of Sciences*, 109(18):E1121–E1130.

[5] Blank, L. M., Kuepfer, L., and Sauer, U. (2005). Large-scale 13c-flux analysis reveals mechanistic principles of metabolic network robustness to null mutations in yeast. *Genome biology*, 6(6):1–16.

[6] Bynum, M. L., Hackebeil, G. A., Hart, W. E., Laird, C. D., Nicholson, B. L., Siirola, J. D., Watson, J.-P., and Woodruff, D. L. (2021). *Pyomo–optimization modeling in python*, volume 67. Springer Science & Business Media, third edition.

[7] Cappart, Q., Chételat, D., Khalil, E., Lodi, A., Morris, C., and Veličković, P. (2021). Combinatorial optimization and reasoning with graph neural networks. *arXiv preprint arXiv:2102.09544*.

[8] Chen, K., Gao, Y., Mih, N., O'Brien, E. J., Yang, L., and Palsson, B. O. (2017). Thermosensitivity of growth is determined by chaperone-mediated proteome reallocation. *Proceedings of the National Academy of Sciences*, 114(43):11548–11553.

[9] Deutscher, D., Meilijson, I., Kupiec, M., and Ruppin, E. (2006). Multiple knockout analysis of genetic robustness in the yeast metabolic network. *Nature genetics*, 38(9):993–998.

[10] Domenzain, I., Sánchez, B., Anton, M., Kerkhoven, E. J., Millán-Oropeza, A., Henry, C., Siewers, V., Morrissey, J. P., Sonnenschein, N., and Nielsen, J. (2022). Reconstruction of a catalogue of genome-scale metabolic models with enzymatic constraints using gecko 2.0. *Nature communications*, 13.

[11] Ebrahim, A., Lerman, J. A., Palsson, B. O., and Hyduke, D. R. (2013). Cobrapy: constraints-based reconstruction and analysis for python. *BMC systems biology*, 7(1):1–6.

[12] Edwards, J. S., Covert, M., and Palsson, B. (2002). Metabolic modelling of microbes: the flux-balance approach. *Environmental microbiology*, 4(3):133–140.

[13] Fang, X., Lloyd, C. J., and Palsson, B. O. (2020). Reconstructing organisms in silico: genome-scale models and their emerging applications. *Nature Reviews Microbiology*, 18(12):731–743.

[14] Feist, A. M., Henry, C. S., Reed, J. L., Krummenacker, M., Joyce, A. R., Karp, P. D., Broadbelt, L. J., Hatzimanikatis, V., and Palsson, B. Ø. (2007). A genome-scale metabolic reconstruction for escherichia coli k-12 mg1655 that accounts for 1260 orfs and thermodynamic information. *Molecular systems biology*, 3(1):121.

[15] Folger, O., Jerby, L., Frezza, C., Gottlieb, E., Ruppin, E., and Shlomi, T. (2011). Predicting selective drug targets in cancer through metabolic networks. *Molecular Systems Biology*, 7(1):501.

[16] Gu, C., Kim, G. B., Kim, W. J., Kim, H. U., and Lee, S. Y. (2019a). Current status and applications of genome-scale metabolic models. *Genome Biology*, 20(1):1–18.

[17] Gu, C., Kim, G. B., Kim, W. J., Kim, H. U., and Lee, S. Y. (2019b). Current status and applications of genome-scale metabolic models. *Genome biology*, 20(1):1–18.

[18] Heiner, M., Gilbert, D., and Donaldson, R. (2008). Petri nets for systems and synthetic biology. volume 5016, pages 215–264.

[19] Henry, C. S., Jankowski, M. D., Broadbelt, L. J., and Hatzimanikatis, V. (2006). Genome-scale thermodynamic analysis of Escherichia coli metabolism. *Biophysical Journal*, 90(4):1453–1461.

[20] Jin, Z., Wang, Y., Wang, Q., Ming, Y., Ma, T., and Qu, H. (2022). Gnnlens: A visual analytics approach for prediction error diagnosis of graph neural networks. *IEEE Transactions on Visualization and Computer Graphics*.

[21] Júlvez, J. and Oliver, S. G. (2019). Flexible nets: a modeling formalism for dynamic systems with uncertain parameters. *Discrete Event Dynamic Systems*, 29(3):367–392.

[22] Karmarkar, N. (1984). A new polynomial-time algorithm for linear programming. In *Proceedings of the sixteenth annual ACM symposium on Theory of computing*, pages 302–311.

[23] Karp, R. M. (1972). Reducibility among combinatorial problems. In *Complexity of computer computations*, pages 85–103. Springer.

[24] Kim, Y., Kim, G. B., and Lee, S. Y. (2021). Machine learning applications in genome-scale metabolic modeling. *Current Opinion in Systems Biology*, 25:42–49.

[25] King, Z. A., Lu, J., Dräger, A., Miller, P., Federowicz, S., Lerman, J. A., Ebrahim, A., Palsson, B. O., and Lewis, N. E. (2016). Bigg models: A platform for integrating, standardizing and sharing genome-scale models. *Nucleic acids research*, 44(D1):D515–D522.

[26] Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

[27] Kipf, T. N. and Welling, M. (2016). Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.

[28] Kroll, A., Engqvist, M. K., Heckmann, D., and Lercher, M. J. (2021). Deep learning allows genome-scale prediction of michaelis constants from structural features. *PLoS biology*, 19(10):e3001402.

[29] Kumar, M., Ji, B., Zengler, K., and Nielsen, J. (2019). Modelling approaches for studying the microbiome. *Nature Microbiology*, 4(8):1253–1267.

[30] Lachance, J.-C., Matteau, D., Brodeur, J., Lloyd, C. J., Mih, N., King, Z. A., Knight, T. F., Feist, A. M., Monk, J. M., Palsson, B. O., et al. (2021). Genome-scale metabolic modeling reveals key features of a minimal gene set. *Molecular systems biology*, 17(7):e10099.

[31] Li, F., Yuan, L., Lu, H., Li, G., Chen, Y., Engqvist, M. K., Kerkhoven, E. J., and Nielsen, J. (2022). Deep learning-based kcat prediction enables improved enzyme-constrained model reconstruction. *Nature Catalysis*, pages 1–11.

[32] Li, Y., Tarlow, D., Brockschmidt, M., and Zemel, R. (2015). Gated graph sequence neural networks. *arXiv preprint arXiv:1511.05493*.

[33] M, H., FT, B., C, C., A, D., S, H., SM, K., M, K., NL, N., CJ, M., BG, O., S, S., JC, S., R, S., LP, S., D, W., DJ, W., and F, Z. (2019). The Systems Biology Markup Language (SBML): Language Specification for Level 3 Version 2 Core Release 2. *Journal of integrative bioinformatics*, 16(2).

[34] Mahadevan, R., Edwards, J. S., and Doyle III, F. J. (2002). Dynamic flux balance analysis of diauxic growth in escherichia coli. *Biophysical journal*, 83(3):1331–1340.

[35] Mahadevan, R. and Schilling, C. H. (2003). The effects of alternate optimal solutions in constraint-based genome-scale metabolic models. *Metabolic Engineering*, 5(4):264–276.

[36] Malik-Sheriff, R. S., Glont, M., Nguyen, T. V., Tiwari, K., Roberts, M. G., Xavier, A., Vu, M. T., Men, J., Maire, M., Kananathan, S., et al. (2020). Biomodels—15 years of sharing computational models in life science. *Nucleic acids research*, 48(D1):D407–D415.

[37] McAnulty, M. J., Yen, J. Y., Freedman, B. G., and Senger, R. S. (2012). Genome-scale modeling using flux ratio constraints to enable metabolic engineering of clostridial metabolism in silico. *BMC Systems Biology*, 6(1):1–15.

[38] Murata, T. (1989). Petri Nets: Properties, Analysis and Applications. *Procs. of the IEEE*, 77(4):541–580.

[39] Navid, A. and Almaas, E. (2012). Genome-level transcription data of Yersinia pestis analyzed with a New metabolic constraint-based approach. *BMC Systems Biology*, 6(1):1–18.

[40] Oarga, A., Bannerman, B., and Júlvez, J. (2020). Growth dependent computation of chokepoints in metabolic networks. In *International Conference on Computational Methods in Systems Biology*, pages 102–119. Springer.

[41] O'brien, E. J., Lerman, J. A., Chang, R. L., Hyduke, D. R., and Palsson, B. Ø. (2013). Genome-scale models of metabolism and gene expression extend and refine growth phenotype prediction. *Molecular systems biology*, 9(1):693.

[42] Oftadeh, O., Salvy, P., Masid, M., Curvat, M., Miskovic, L., and Hatzimanikatis, V. (2021). A genome-scale metabolic model of saccharomyces cerevisiae that integrates expression constraints and reaction thermodynamics. *Nature Communications*, 12(1):1–10.

[43] Orth, J. D., Conrad, T. M., Na, J., Lerman, J. A., Nam, H., Feist, A. M., and Palsson, B. Ø. (2011). A comprehensive genome-scale reconstruction of Escherichia coli metabolism–2011. *Molecular Systems Biology*, 7(1).

[44] Orth, J. D., Thiele, I., and Palsson, B. Ø. (2010). What is flux balance analysis? *Nature Biotechnology 2010 28:3*, 28(3):245–248.

[45] Oyelade, J., Isewon, I., Uwoghiren, E., Aromolaran, O., and Oladipupo, O. (2018). In Silico Knockout Screening of Plasmodium falciparum Reactions and Prediction of Novel Essential Reactions by Analysing the Metabolic Network. *BioMed Research International*, 2018.

[46] Papp, B., Pál, C., and Hurst, L. D. (2004). Metabolic network analysis of the causes and evolution of enzyme dispensability in yeast. *Nature*, 429(6992):661–664.

[47] Plackett, B. (2020). Why big pharma has abandoned antibiotics. *Nature*, 586(7830):S50–S50.

[48] Ramos, P. I. P., Fernández Do Porto, D., Lanzarotti, E., Sosa, E. J., Burguener, G., Pardo, A. M., Klein, C. C., Sagot, M. F., De Vasconcelos, A. T. R., Gales, A. C., Marti, M., Turjanski, A. G., and Nicolás, M. F. (2018). An integrative, multi-omics approach towards the prioritization of Klebsiella pneumoniae drug targets. *Scientific Reports*, 8(1):1–19.

[49] Recalde, L., Haddad, S., and Silva, M. (2007). Continuous petri nets: Expressive power and decidability issues. In *International Symposium on Automated Technology for Verification and Analysis*, pages 362–377. Springer.

[50] Richelle, A., David, B., Demaegd, D., Dewerchin, M., Kinet, R., Morreale, A., Portela, R., Zune, Q., and von Stosch, M. (2020). Towards a widespread adoption of metabolic modeling tools in biopharmaceutical industry: a process systems biology engineering perspective. *npj Systems Biology and Applications*, 6(1):1–5.

[51] Roberts, S. B., Gowen, C. M., Brooks, J. P., and Fong, S. S. (2010). Genome-scale metabolic analysis of Clostridium thermocellum for bioethanol production. *BMC Systems Biology*, 4(1):1–17.

[52] Rosenblatt, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386.

[53] Scarselli, F., Gori, M., Tsoi, A. C., Hagenbuchner, M., and Monfardini, G. (2008). The graph neural network model. *IEEE transactions on neural networks*, 20(1):61–80.

[54] Shiloh-Perl, L. and Giryes, R. (2020). Introduction to deep learning. *arXiv preprint arXiv:2003.03253*.

[55] Varma, A. and Palsson, B. Ø. (1994a). Metabolic Flux Balancing: Basic Concepts, Scientific and Practical Use. *Nature Biotechnology*, 12(10):994–998.

[56] Varma, A. and Palsson, B. O. (1994b). Stoichiometric flux balance models quantitatively predict growth and metabolic by-product secretion in wild-type escherichia coli w3110. *Applied and environmental microbiology*, 60(10):3724–3731.

[57] Vedithi, S. C., Malhotra, S., Acebrón-García-de Eulate, M., Matusevicius, M., Torres, P. H. M., and Blundell, T. L. (2021). Structure-guided computational approaches to unravel druggable proteomic landscape of mycobacterium leprae. *Frontiers in Molecular Biosciences*, 8:663301.

[58] Veličković, P. and Blundell, C. (2021). Neural algorithmic reasoning. *Patterns*, 2(7):100273.

[59] Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., and Bengio, Y. (2017). Graph attention networks. *arXiv preprint arXiv:1710.10903*.

[60] Vlassis, N., Pacheco, M. P., and Sauter, T. (2014). Fast reconstruction of compact context-specific metabolic network models. *PLoS Comput Biol*, 10(1):e1003424.

[61] WHO (2021). Antimicrobial resistance. https://www.who.int/health-topics/antimicrobial-resistance. Accessed: 2021-07-03.

[62] Yeh, I., Hanekamp, T., Tsoka, S., Karp, P. D., and Altman, R. B. (2004). Computational analysis of plasmodium falciparum metabolism: organizing genomic information to facilitate drug discovery. *Genome research*, 14(5):917–924.

[63] You, J., Ying, Z., and Leskovec, J. (2020). Design space for graph neural networks. *Advances in Neural Information Processing Systems*, 33:17009–17021.

[64] Zampieri, G., Vijayakumar, S., Yaneske, E., and Angione, C. (2019). Machine and deep learning meet genome-scale metabolic modeling. *PLoS computational biology*, 15(7):e1007084.

[65] Zomorrodi, A. R. and Maranas, C. D. (2012). OptCom: A multi-level optimization framework for the metabolic modeling and analysis of microbial communities. *PLoS Computational Biology*, 8(2):1002363.

Appendix A

Proofs

Lemma 3.2.1. $[lb_{\gamma_2}[r], ub_{\gamma_2}[r]] \subseteq [lb_{\gamma_1}[r], ub_{\gamma_1}[r]] \ \forall\ r \in \mathcal{R} \wedge \forall\ \gamma_1,\ \gamma_2$ *such that* $0 \leq \gamma_1 < \gamma_2 \leq 1$.

Proof of Lemma 3.2.1 . Given $\gamma \in [0, 1]$, the range of feasible fluxes of $r \in \mathcal{R}$, $[lb_\gamma[r], ub_\gamma[r]]$, is given by the solutions of FVA (2.3). Notice that the constraints of such linear programming problem define a convex set of possible solutions which can only shrink as γ increases, i.e. as the constraint $\gamma \cdot \mu_{max} \leq z \cdot v$ becomes more restrictive. Thus, if $\gamma_1 < \gamma_2$ then $lb_{\gamma_1}[r] \leq lb_{\gamma_2}[r]$ and $ub_{\gamma_1}[r] \geq ub_{\gamma_2}[r]$. □

Lemma 3.2.2. $DR_{\gamma_1} \subseteq DR_{\gamma_2}\ \forall\ \gamma_1,\ \gamma_2$ *such that* $0 \leq \gamma_1 < \gamma_2 \leq 1$.

Proof of Lemma 3.2.2 . Let $r \in DR_{\gamma_1}$, i.e. $lb_{\gamma_1}[r] = ub_{\gamma_1}[r] = 0$, then by Lemma 3.2.1, it follows that $lb_{\gamma_2}[r] = ub_{\gamma_2}[r] = 0$ and hence $r \in DR_{\gamma_2}$. □

Lemma 3.2.3. $DR_{\gamma_1} \supseteq DR_{\gamma_2}\ \forall\ \gamma_1,\ \gamma_2$ *such that* $0 \leq \gamma_1 < \gamma_2 < 1$.

Proof of Lemma 3.2.3 . The convex set of possible solutions defined by the constraints in (2.3) can only decrease as γ increases, see Lemma 3.2.1. Assume there exist γ_a, γ_b such that $0 \leq \gamma_a < \gamma_b < 1$, $lb_{\gamma_a}[r] < ub_{\gamma_a}[r]$ and $lb_{\gamma_b}[r] = ub_{\gamma_b}[r] = 0$, i.e. r becomes dead when γ_a is increased to γ_b. Then, given that the constraints in (2.3) are linear, if γ_b is further increased by $\epsilon \in \mathbb{R}$ such that $\epsilon > 0$ and $\gamma_b + \epsilon < 1$, then $lb_{\gamma_b+\epsilon}[r] > ub_{\gamma_b+\epsilon}[r]$ should hold which is not possible because a lower bound cannot exceed an upper bound. □

Lemma 4.1.1. *Let H be any hyperplane that delimits the convex cone solution space of the LPP in* (4.1)*. Let x^* denote an optimal solution defined by the LPP. If any x^* is located exclusively in H, then for all x solution of the LPP is 0.*

Proof of Lemma 4.1.1 . From the definition of hyperplane, H can be defined as $n^T x + b = 0$, where n is the normal of the plane and b the offset. Here $b = 0$ since H passes through the origin of the coordinates. Since any x^* is located exclusively at H, then the objective function hyperplane, whose normal is c, is parallel to H hyperplane. Consequently, and given H definition, any x will hold $n^T x = c^T x = 0$. □

Lemma 4.1.2. *Having $A \cdot x = 0$ and $x \geq 0$, if the LPP has at least one solution and max $c^T x > 0$, then the LPP in* (4.1) *is unbounded.*

Proof of Lemma 4.1.2 . Let us suppose for the sake of contradiction that the LPP in (4.1) is bounded and let us denote μ the objective value. Since the LPP is solvable, it exist one x such that $c^T x = \mu$. Given this x and given $\lambda > 1$, vector λx is also a solution of the LPP (4.1). However, the optimal solution obtained with λx is equal to $c^T \lambda x = \lambda \mu$, this is, the vector yields an objective value greater than μ which is a contradiction. □

Proposition 4.2.1. *Given the unbounded LPP of* (4.1) *with max $c^T x > 0$, if we include the orthogonal constraints $0 \leq l \leq x \leq u$, then with the resulting LPP, $\exists i \in [1, n]$ such that that $lb_1[i] = ub_1[i] = u[i]$ or $lb_1[i] = ub_1[i] = l[i]$.*

Proof of Proposition 4.2.1 . We will prove that for any x that is an optimal solution of (4.2) it holds that $x[i] = u[i]$ (or $x[i] = l[i]$) with $i \in [1, n]$. If we impose orthogonal constraints $0 \leq l \leq x \leq u$ to the LPP in (4.1) it can be seen that the LPP is no longer unbounded since we have imposed restrictions on all variables. Given this, the extreme points at which the optimal solution is located must be at one of the faces defined by the new constraints $x[j] = u[j]$ or $x[j] = l[j]$ with $j \in [1, n]$. If the optimal solution is located at one of the faces defined by a constraint $x[j] \leq u[j]$ (or $l[j] \leq x[j]$), then all solutions must satisfy $x[j] = u[j]$ (or $x[j] = l[j]$). Similarly, the previous condition is also satisfied when the optimal solution is located at the intersection of 2 or more faces. Hence if *max* $c^T x > 0$, then all solutions will satisfy $x[j] = u[j]$ (or $x[j] = l[j]$) with $j \in [1, n]$. □

Proposition 4.2.2. *Let $\mu_{max} > 0$ be the solution of* (4.2)*. Given the LPP in* (4.2)*, if we have $l = lb_1$ and $u = ub_1$, then max $c^T x = min\ c^T x = \mu_{max}$.*

Proof of Proposition 4.2.2 . By proposition 4.2.1 we know that for at least one $i \in [1, n]$ it holds that $lb_1[i] = ub_1[i] = u[i]$ (or $lb_1[i] = ub_1[i] = l[i]$). By imposing this constraint in the LPP we are delimiting the solution space to the specific face or face intersection at which the optimal solution is located. If there exists a solution x such that the previous condition is not satisfied, the solution is not located at such faces and the optimum value can not be obtained. On the other hand, any solution that satisfies the previous condition will necessarily be located at the face (or faces) at which the optimal solutions are located and therefore will also be an optimal solution. □

Proposition 6.2.1. *Assuming $r \in KR\ \forall r \in \mathcal{R}$, given $\gamma \in [0, 1]$ then $FR_\gamma = ER_\gamma$.*

Proof of Proposition 6.2.1 . If a reaction $r \in \mathcal{R}$ is a forced reactions (i.e. $r \in FR_\gamma$), then to sustain the growth given by γ, reaction r needs to have a non-null flux, i.e. $r \in ER_\gamma$. On the other side, if a reaction $r \in \mathcal{R}$ is a growth-dependent essential reaction (i.e. $r \in ER_\gamma$), the reaction must have a non-null flux to produce the growth specified by γ, hence, FVA will yield an interval $[lb_\gamma, ub_\gamma]$ with $0 \notin [lb_\gamma, ub_\gamma]$, consequently $r \in FR_\gamma$. □

Proposition 6.2.2. *Assuming $r \in KR\ \forall r \in \mathcal{R}$, given $\gamma \in [0, 1]$ then $KR_\gamma = \mathcal{R} - ER_\gamma - DR_\gamma$.*

Proof of Proposition 6.2.2 . It can be seen that a reaction $r \in \mathcal{R}$ such that $r \notin DR_\gamma$ will be in FR_γ or KR_γ, i.e. $r \in FR_\gamma \cup KR_\gamma$. Clearly forced and knockable sets are disjoint, $FR_\gamma \cap KR_\gamma = \varnothing$. Similarly, $FR_\gamma \cap DR_\gamma = \varnothing$ and $KR_\gamma \cap DR_\gamma = \varnothing$. It can also be seen that $FR_\gamma \cup KR_\gamma \cup DR_\gamma = \mathcal{R}$. As a consequence of Proposition 6.2.1 we have $KR_\gamma = \mathcal{R} - ER_\gamma - DR_\gamma = \mathcal{R} - FR_\gamma - DR_\gamma$. □

Appendix B

Methods

B.1 Computation

The methods presented in this work that involved computation were mostly implemented with the Python language. The manipulation of the constraint-based models presented, FBA, and FVA computation were performed with the Python toolbox COBRApy [11]. The MILP presented for MROG computation was implemented using Pyomo language [6] and solved by the commercial solver Gurobi Optimizer 9.1.2. *MROG* computation always took less than 4 minutes with an Intel Core i5-9300H CPU @ 2.40GHz $\times$ 8. All the models used for validation purposes were obtained from Biomodels [36] or BiGG [25] repositories.

B.2 CONTRABASS

For the computation of vulnerabilities in genome-scale models, we have developed the tool CONTRABASS. CONTRABASS is a software tool distributed as a Python command line tool but that can also be executed through an online web server at http://contrabass.unizar.es. The CONTRABASS web server is designed to offer an intuitive interface to access the operations of the tool (see Figure B.1).

The tool takes as an input a model in Systems Biology Markup Language (SBML) (33) and computes the set of chokepoints reactions, essential reactions, dead reactions and dead-end metabolites on the model by taking into account the dynamic constraints on the model as explained in this work. The results are then exported as a spreadsheet file and as an interactive HTML report (see Figure B.2). The operations that CONTRABASS allows include the computation of sets of chokepoints, dead, reversible, non-reversible and essential reactions with different values of γ; computation and removal of dead-end metabolites from a model; and update the flux bounds of the reactions according to FVA.

In addition to the above, through the interactive HTML report users can also access the data available in the model, this is, reactions, genes and metabolites along with their databases identifiers if available; explore the reaction sets defined in this work, and also explore the intersection of different sets of vulnerable reactions (see Figure B.3).

The documentation of the tool is available at https://contrabass.readthedocs.io. The source code of the command line tool and the web server is available at https://github.com/openCONTRABASS/CONTRABASS and https://github.com/openCONTRABASS respectively. All the code is released under GPL-3.0 License.

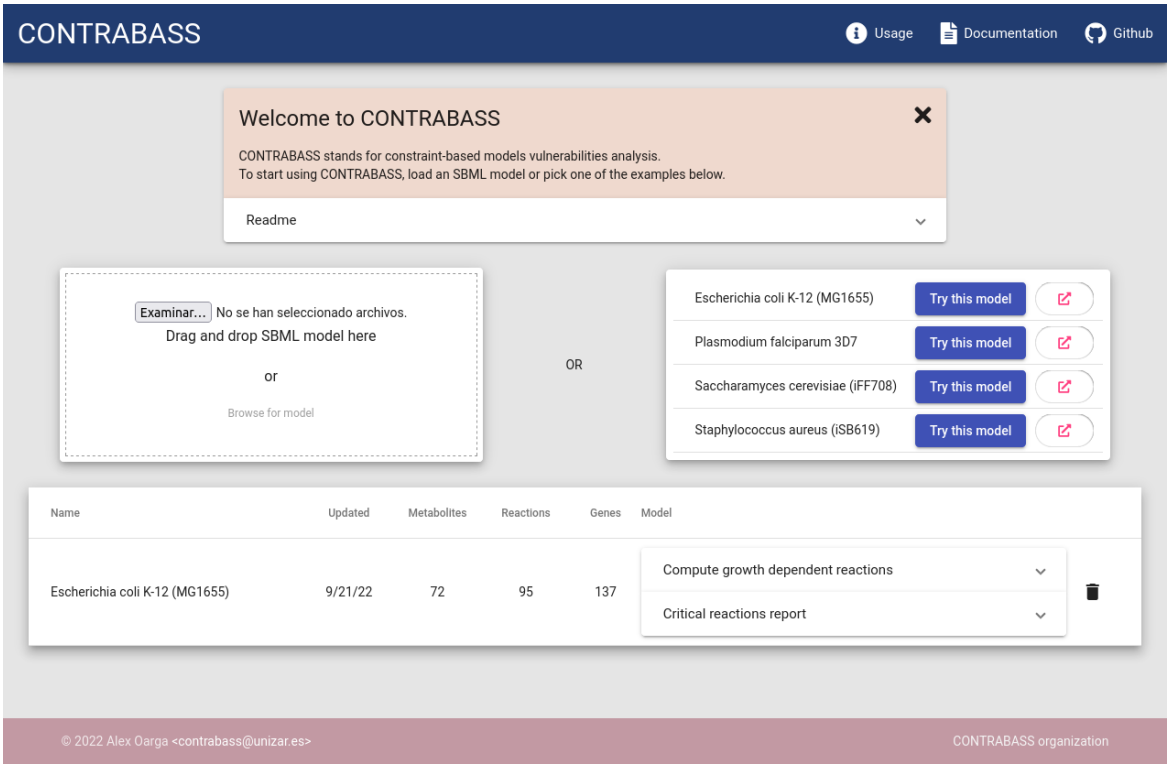


Figure B.1 CONTRABASS initial page view.

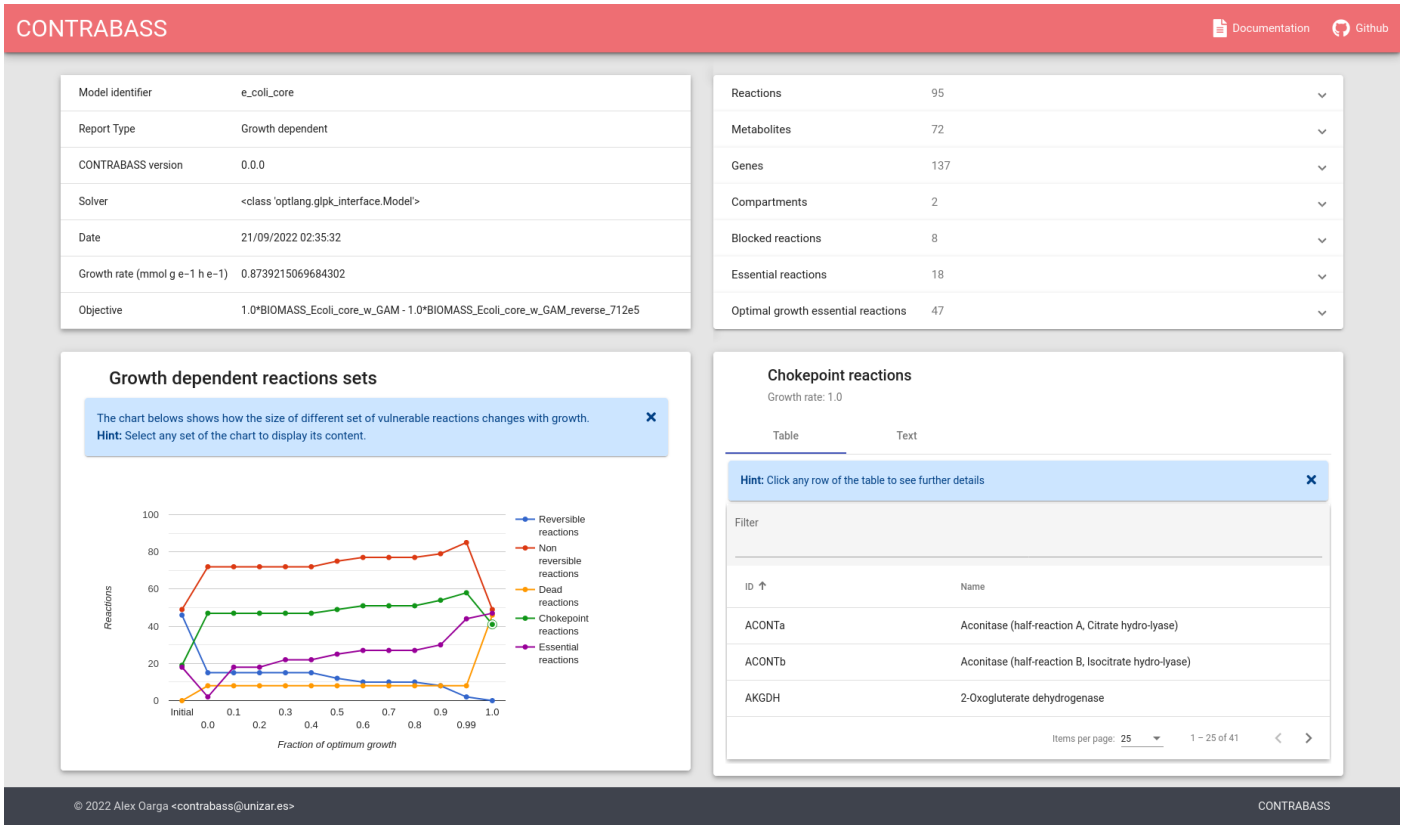


Figure B.2 CONTRABASS growth-dependent reactions report view.

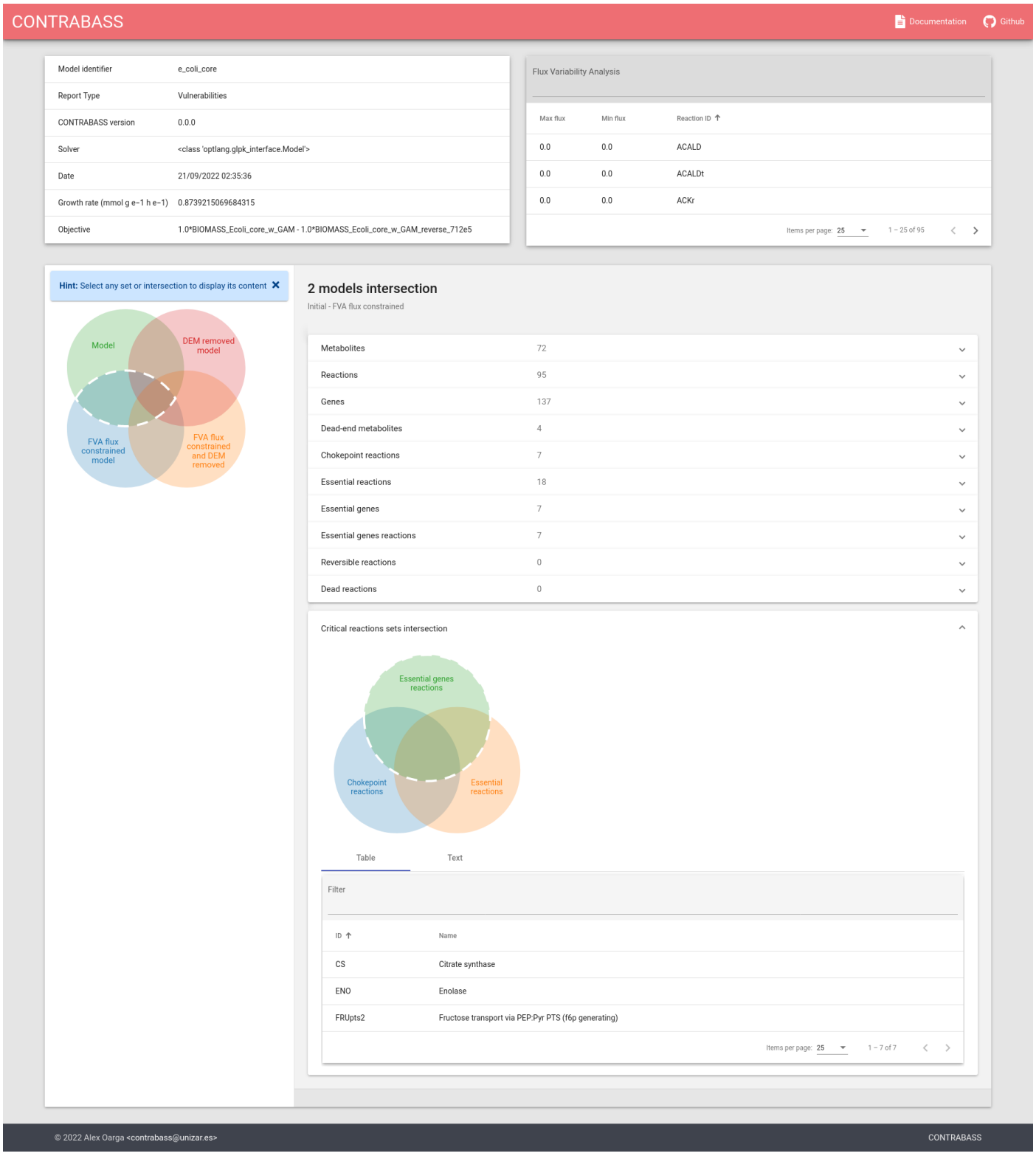


Figure B.3 CONTRABASS vulnerabilities report view.

Appendix C

Vulnerabilities in the literature

As mentioned in Chapter 2, both essential reactions and chokepoint reactions are recognised as potential drug targets. Here, we provide a couple of examples from the literature to better motivate and contextualise our work. Figure C.1 shows a workflow for drug target prioritisation in *M. leprae* were both chokepoint reactions, and essential genes, which are dependent on essential reactions, are involved in the procedure of target selection. Figure C.2 shows a subset of fatty acid metabolism of *Klebsiella pneumoniae* where structures that are chokepoints and essential are ranked higher as druggable targets.

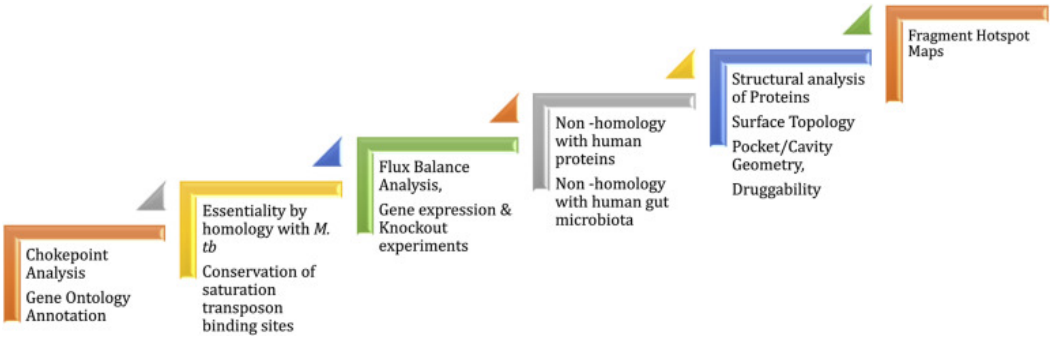


Figure C.1 Vedithi *et al.* proposed workflow for drug target prioritisation in *Mycobacterium leprae*. The druggable proteome is identified from chokepoint reactions. Flux Balance Analysis could also help in the identification of essential genes. Source: [57]

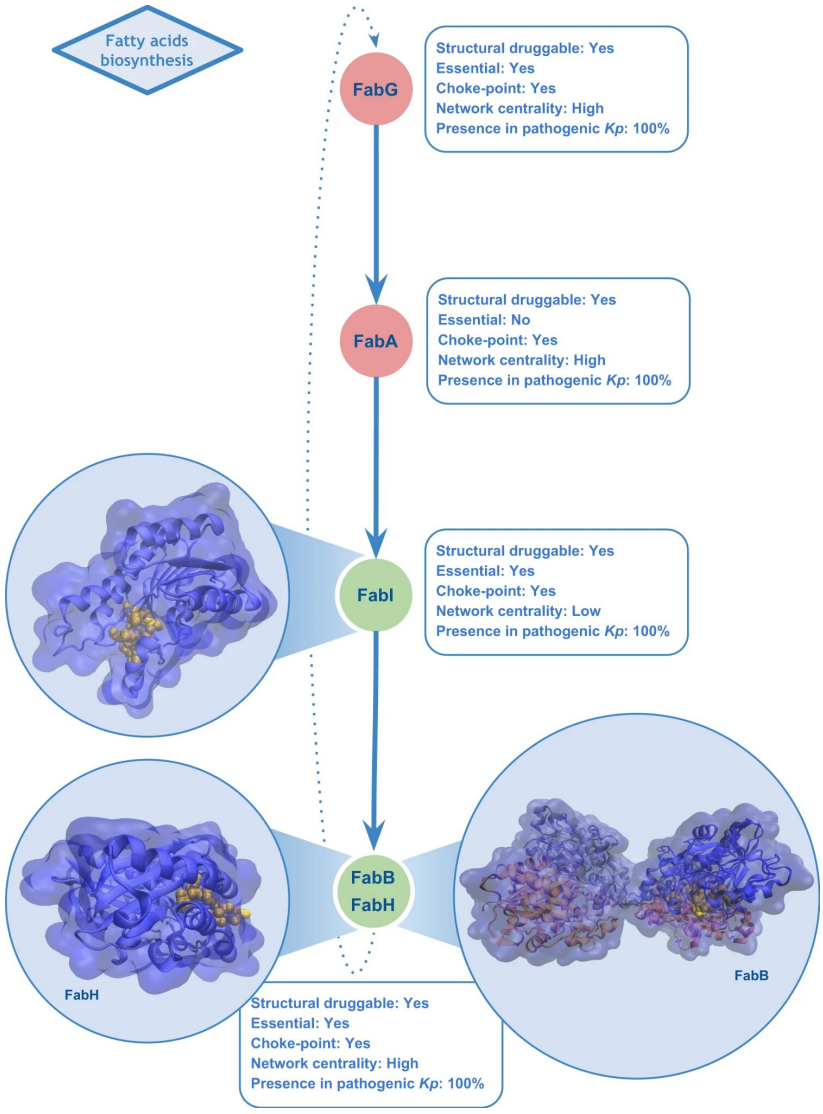


Figure C.2 Subset of the fatty acid pathway where FabB, FabI and FabH are among the 15 top-ranked candidates in the scoring pipeline for drug target selection in *Klebsiella pneumoniae*. Here chokepoint reactions and essential reactions are prioritised as potential drug targets. Source: [48].

Appendix D

Visualisation of model iAM-Pf480 of *Plasmodium falciparum.*

Recall that all *ROG* sets were composed of *EROG* and a subset of knockable reactions. This is shown graphically in Figure D.2. In Figure D.1 a full view of the reactions (grey) and metabolites (yellow) of the model iAM-Pf480 is shown. In D.2a we can see the *EROG* set and in Figure D.2b the KR_1 set. The metabolites present in both sets' reactions are shown in green. The *MROG* shown in Figure D.2c is composed of all reactions from Figure D.2a plus certain reactions from D.2b.

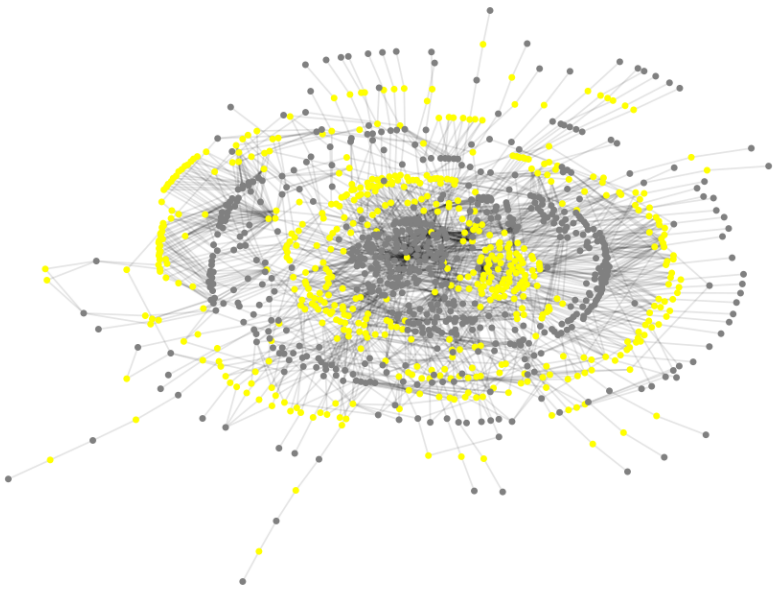
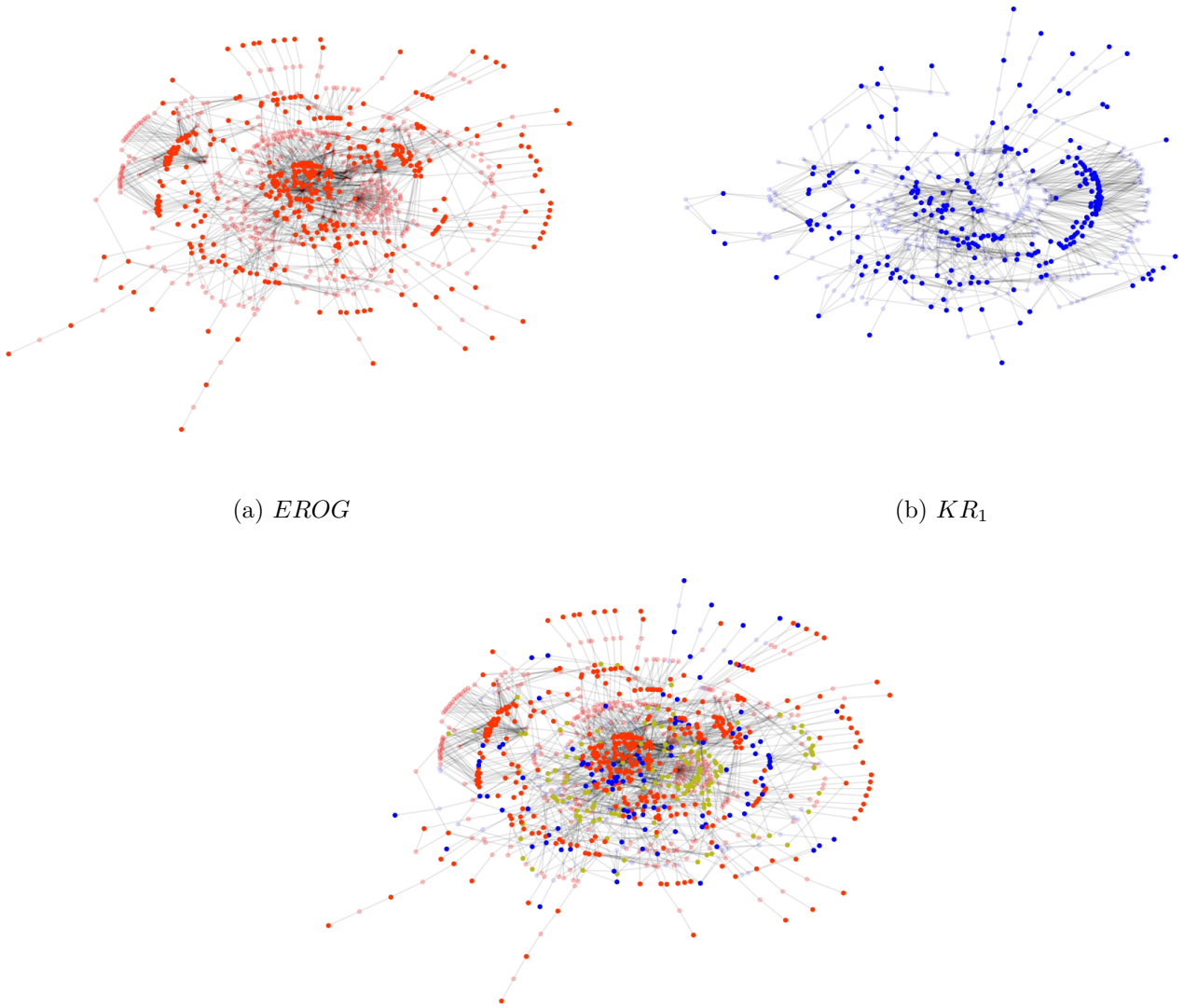


Figure D.1 Reactions (gray) and metabolites (yellow) of model iAM-Pf480 of *Plasmodium falciparum*. Dead reactions and their metabolites have been previously removed.



(a) $EROG$

(b) KR_1

(c) $MROG$

Figure D.2 D.2a) Reactions of $EROG$ are highlighted in red. Metabolites are shown in a less intense red color. D.2b) Reactions of KR_1. Reactions are shown in blue and metabolites in a less intense blue. D.2c) Set $MROG$ is composed of reactions of $EROG$ (red) and reactions of KR_1 (blue). Lighter red nodes represent metabolites of reactions in $EROG$ and lighter blue nodes represent metabolites in KR_1. Green nodes represent metabolites involved in both $EROG$ and KR_1 reactions. In all figures, dead reactions and their metabolites have been removed.

Appendix E

Further reading: Bounded Non-Homogeneous Case

In this appendix, we are going to prove that, if we compute maximum and minimum bounds on each variable of an LPP and then impose them as bounding-box constraints, then we can only obtain the optimum objective value.

Let us now consider the following LPP:

$$\begin{aligned} \max \quad & c^T x \\ st. \quad & A \cdot x \leq b \\ & 0 \leq l \leq x \leq u \\ & x \geq 0 \end{aligned} \tag{E.1}$$

As done until now, let us denote μ_{max} the solution of the LPP in (E.1). The equivalent procedure of FVA for non-homogeneous systems is defined by the following maximisation/minimisation problem:

$$\begin{aligned} \max / \min \quad & c^T x \\ st. \quad & A \cdot x \leq b \\ & 0 \leq l \leq x \leq u \\ & x \geq 0 \\ & \gamma \cdot \mu_{max} \leq c^T x \end{aligned} \tag{E.2}$$

with $\gamma \in [0, 1]$. We will denote $lb_\gamma, ub_\gamma \in \mathcal{R}^{|x|}$ the solutions obtained from minimasing/maximising the LPP in (E.2).

First, we will prove Lemma E.0.1 for the sake of arriving at a contradiction.

Lemma E.0.1. *Given the LPP of* (E.1) *with* $lb_1[i] < ub_1[i]\ \forall i \in [1, n]$ *and* $n > 1$*, then* $\exists y \in \mathbb{R}^n$ *such that* y *is a solution of the LPP and* $lb_1[i] < y[i] < ub_1[i]\ \forall i \in [1, n]$.

Proof. Let x_i with $i \in [1, n]$ denote the i-th variable of the LPP. Let us also denote $\mathbb{S}$, the convex set of solutions defined by the LPP. We are going to prove our claim by induction. First we prove the base case $n = 2$:

(base case): Given $x_1, x_2 \in \mathbb{S}$, by definition we have that:

$$\begin{aligned}
&\exists\ s_1^u \in \mathbb{S} \text{ with } s_1^u[x_1] = ub_1[x_1] \\
&\exists\ s_1^l \in \mathbb{S} \text{ with } s_1^l[x_1] = lb_1[x_1] \\
&\exists\ s_2^u \in \mathbb{S} \text{ with } s_2^u[x_2] = ub_1[x_2] \\
&\exists\ s_2^l \in \mathbb{S} \text{ with } s_2^l[x_2] = lb_1[x_2]
\end{aligned}$$

Notice that $s_1^u \neq s_1^l$ and $s_2^u \neq s_2^l$. Since all the previous points belong to the convex set $\mathbb{S}$, it also holds that:

$$\begin{aligned}
&\exists\ s_1^m \in \mathbb{S} \text{ such that } s_1^m = \lambda_1 s_1^l + (1 - \lambda_1) s_1^u \text{ with } 0 < \lambda_1 < 1 \\
&\exists\ s_2^m \in \mathbb{S} \text{ such that } s_2^m = \lambda_2 s_2^l + (1 - \lambda_2) s_2^u \text{ with } 0 < \lambda_2 < 1
\end{aligned}$$

Notice that for s_1^m it is true that $lb_1[x_1] < s_1^m[x_1] < ub_1[x_1]$. Similarly, for s_2^m it is true that $lb_1[x_2] < s_2^m[x_2] < ub_1[x_2]$. This is because:

$$\begin{aligned}
lb_1[x_1] < \lambda_1 lb_1[x_1] + (1 - \lambda_1) ub_1[x_1] < ub_1[x_1] \text{ with } 0 < \lambda_1 < 1 \\
lb_1[x_2] < \lambda_2 lb_1[x_2] + (1 - \lambda_2) ub_1[x_2] < ub_1[x_2] \text{ with } 0 < \lambda_2 < 1
\end{aligned}$$

We can now obtain a point $y \in \mathbb{S}$ equal to:

$$y = \lambda_y s_1^m + (1 - \lambda_y) s_2^m \text{ with } 0 < \lambda_y < 1$$

This point y holds $lb_1[x_1] < y[x_1] < ub_1[x_1]$ and $lb_1[x_2] < y[x_2] < ub_1[x_2]$. Since y is in $\mathbb{S}$, y is a solution of the LPP.

(induction step): let say proposition holds for n variables, i.e. $\exists y$ such that $lb_1[i] < y[i] < ub_1[i]\ \forall i \in [1, n]$. Given a LPP with $n + 1$ variables, lets also say that we have $y[n + 1] = lb_1[n + 1]$ or $y[n + 1] = lb_1[n + 1]$ (otherwise the proof has ended). As before, we have:

$$\begin{aligned}
&\exists\ s_{n+1}^u \in \mathbb{S} \text{ with } s_{n+1}^u[x_{n+1}] = ub_1[x_{n+1}] \\
&\exists\ s_{n+1}^l \in \mathbb{S} \text{ with } s_{n+1}^l[x_{n+1}] = lb_1[x_{n+1}] \\
&\exists\ s_{m+1}^m \in \mathbb{S} \text{ such that } s_{m+1}^m = \lambda_{n+1} s_{m+1}^l + (1 - \lambda_{n+1}) s_{m+1}^u \text{ with } 0 < \lambda_{n+1} < 1
\end{aligned}$$

We can now obtain a point $y' \in \mathbb{S}$ equal to:

$$y' = \lambda s_{n+1}^m + (1 - \lambda) s_{n+1}^m \text{ with } 0 < \lambda < 1$$

This point y' holds $lb_1[i] < y'[i] < ub_1[i]\ \forall i \in [1, n + 1]$. □

We are ready to prove now that $\exists i$ such that for any solution x of the LPP in (E.1), $x[i] = lb_1[i] = ub_1[i]$.

Proposition E.0.1. *Given the LPP of* (E.1) *of n variables. Let us denote S the polyhedra defined by the LPP. If $n > 1$, S has a non-empty interior and $c^T x$ is an objective function whose maximum is reached exclusively at a face or vertex of S, then $\exists i \in [1, n]$ such that $lb_1[i] = ub_1[i] = u[i]$ (or $lb_1[i] = ub_1[i] = l[i]$).*

Proof. If the condition does not hold, from Lemma E.0.1 we can obtain the optimum objective value with an interior point, hence we reach a contradiction. □

Finally, we can prove that if we impose the minimum and maximum bounds lb_1 and ub_1 on each variable of the LPP, then the optimum objective value is always $mu_{\max}$.

Proposition E.0.2. *Given the LPP of* (E.1) *of n variables. Let us denote S the polyhedra defined by the LPP. If $n > 1$, S has a non-empty interior, $c^T x$ is an objective function whose maximum is reached exclusively at a face or vertex of S, and $l = lb_1$, $u = ub_1$, then $max\ c^T x = min\ c^T x = \mu_{max}$.*

Same proof as applied in Proposition 4.2.2. If the solution space is constrained to a face or vertex where the optimum value is located, then the constrained solution space can only yield the optimum objective value.

Appendix F

Further reading: towards hybrid data-model approaches

Through this work, a common assumption of steady state is made. This assumption has produced promising results as been shown. Beyond the steady state, GEMs have also enabled the study of metabolic systems in dynamic conditions. This modelling however is more expensive and, as mentioned, requires more data on the modelled system. Consequently, it remains challenging and feasible only for small systems. Another main issue regarding metabolic modelling is clearly stated by *Zampieri et al.* [64]:

"Measurement is subject to intrinsic noise and uncertainty that has to be corrected. Additionally, traditional omics are affected by sampling or technology-specific systematic errors".

When systems are modelled by current approaches they are usually assumed to be free of error. Modelling approaches presented in this work, such as Petri nets, assume the data to be free of error, which in the end hampers our modelling abilities. Ideally, models should be able to operate on multidimensional data that is able to capture the richness of the input data. Based on the latest advances in algorithmic reasoning [7], here we present Neural Petri Nets, an approach for modelling Petri Nets, that operate in a high dimensional space.

F.1 Preliminary Definitions

F.1.1 Neural Networks Fundamentals

Before introducing Neural Petri Nets we need to introduce the fundamentals of neural networks. Neural network are machine learning models composed of *neurons* or *perceptrons* [52]. Perceptrons receive an input vector $x \in \mathbb{R}^N$. The output of the model $y \in \mathbb{R}$ is given by: $y = \sigma(b + \sum_i^N x_i w_i)$, where $w \in \mathbb{R}^N$ is the weight vector, b is a bias value and σ is the activation function. Usual activation functions include logistic sigmoid, hyperbolic tangent, and rectified linear function (ReLU).

Neural networks are usually stacked in layers, where each layer input receives the previous layer's output. The most popular of such networks is the *mulilayer perceptron* (MLP), where each perceptron is fully connected to the previous layer output. To further know about neural

networks, deep neural networks, and their training procedure, a gentle introduction can be found at: [54].

F.1.2 Graph Neural Networks

Graph Neural Networks (GNN) [53, 32, 27] are machine learning models that learn on data that is accompanied by a graph structure. Proper design of GNNs is currently a major ongoing challenge in machine learning [63]. GNNs are composed of layers of message-passing networks. In each layer, the embedding vector h_i of node i is computed from the aggregation of the embeddings of their neighbour nodes $\mathcal{N}(i)$ of the previous layer (see Figure F.1). The initial embedding vector is usually the input feature vector that each node is given. Embedding vectors h_i can be therefore considered as representations of the input feature vector aggregated with the features' representations received from their neighbours.

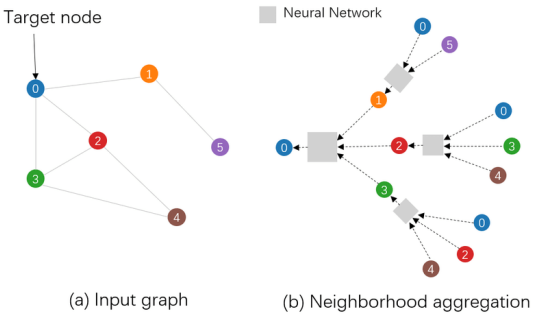


Figure F.1 Each GNN layer computes a node representation by aggregating its neighbours' representations. Source: [20].

Notice that, unlike conventional neural networks, stacking layers of GNNs does noes improve the network expressivity, but instead it expands the graph computation of neighbours aggregations. Formally and generally, the $(k+1)$-th GNN layer of node v can be defined as:

$$h_v^{(k+1)} = AGG(\{ACT(W^{(k)}h_u^{(k)} + b^{(k)}), u \in \mathcal{N}(v)\}) \tag{F.1}$$

where $h_v^{(k)}$ is the k-th layer embedding of node v, $W^{(k)}$ and $b^{(k)}$ are the trainable weight matrix and bias respectively, ACT is an activation function and AGG is a commutative aggregation function such as maximisation, summation or mean [63].

F.2 Neural Petri Nets

We are going to formally introduce Neural Petri Nets:

Definition F.2.1. A Neural Petri Net (NPN) is a tuple $\{\mathcal{N}, X_P^{(t)}, X_T, f_N\}$, where $\mathcal{N}$ is a generalised Petri Net i.e. $\mathcal{N} = \{P, T, Pre, Post\}$, $X_P^{(t)} \in \mathbb{R}^{|P|\times|n|}$ is the set of n features of each place of the net at instant t, $X_T \in \mathbb{R}^{|T|\times|m|}$ is the set m features of the transitions of the network and f_N is a neural network that defines the firing function $f_N : \mathcal{N} \times \mathcal{T} \times \mathbb{R}^{|P|\times|n|} \times \mathbb{R}^{|T|\times|m|} \to \mathbb{R}^{|P|\times|n|}$, with $\mathcal{T}$ being a subset of T (i.e. $\mathcal{T} \subseteq T$).

In NPNs, $X_P^{(t)}$ are the equivalent of the marking in Petri Nets. However, contrary to conventional Petri Nets, NPNs do not have a clearly defined firing logic for transitions. Instead, the new features $X_P^{(t+1)}$, obtained after firing a certain set of transitions $\mathcal{T}$ at instant t, are obtained as follows:

$$X_P^{(t+1)} = f_N(\mathcal{N}, \mathcal{T}, X_P^{(t)}, X_T) \tag{F.2}$$

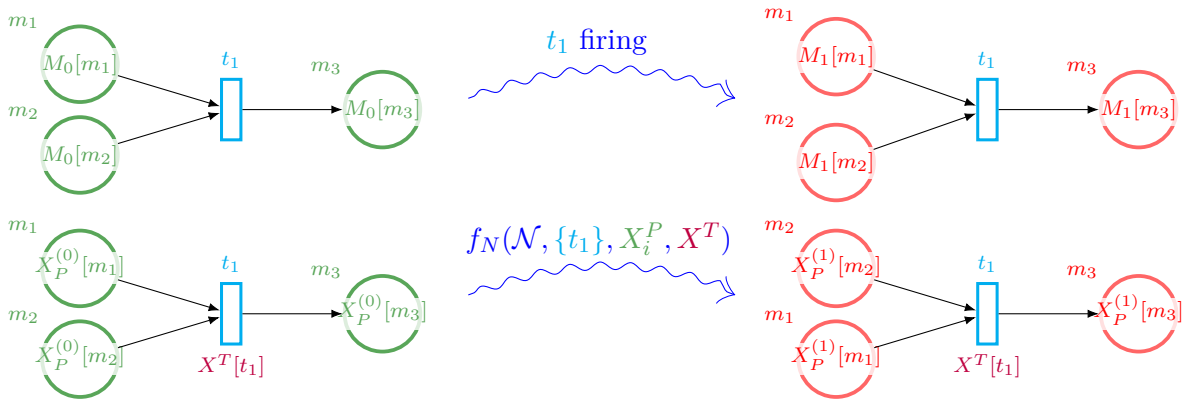


Figure F.2 Comparison of firing semantics between discrete Petri Nets (top) and Neural Petri Nets (bottom). In discrete Petri Nets, marking M_1 is a result of firing transition t_1. In NPNs, place features $X_P^{(1)}$ are the output of f_N.

This is, NPNs do not have explicitly enabled transitions and firing logic, but rather the function f_N that encodes the firing logic. Notice that this function f_N is shared for all transitions.

Example F.2.1. Figure F.2 shows a transition t_1 of a Petri Net. In Petri Nets (top of the Figure), the firing of t_1 turns the marking M_0 into the new marking M_1. In Neural Petri Nets (bottom of the Figure), features $X_P^{(0)}$ are turned into features $X_P^{(1)}$ by function $f_N(\mathcal{N}, \{t_1\}, X_P^{(t)}, X_T)$.

F.2.1 Training NPNs

Notice that NPNs are particularly interesting, as one could potentially emulate various types of conventional Petri Nets through them. For instance, if $X_P^{(t)} = M_t$, with $M_t \in \mathbb{N}^{|P|}$ being the marking of a discrete Petri Net, NPNs can learn to execute the firing of transitions (i.e. generate M_{t+1} as output). The main motivation of NPNs however is that they operate over high-dimensional latent spaces, hence making them appealing for certain data-driven modelling tasks (e.g. biological systems as mentioned earlier).

Suppose that we want to learn to emulate various types of Petri Nets through a NPN, and for modelling purposes, we want to apply the learnt modelling rules in natural noisy inputs. An ideal approach for this task is to follow a Neural Algorithmic Reasoner blueprint, and more specifically, a 1-step Algorithmic Graph Executioner [58].

Figure F.3 shows an example of how Neural Petri Nets can be trained following a Neural Algorithmic Reasoner blueprint. The following lines explain how this training is performed:

1. Train f_N for abstract inputs, in this case, Petri Nets firing sequences. In the example, x and $\hat{x}$ could represent inputs from discrete, hybrid, and continuous Petri Nets. Each network has its given weights for each place and parameters for each transition. The expected output $g(f_N(f(x)))$ and $\hat{g}(f_N(\hat{f}(x)))$ in this case are the results of firing all

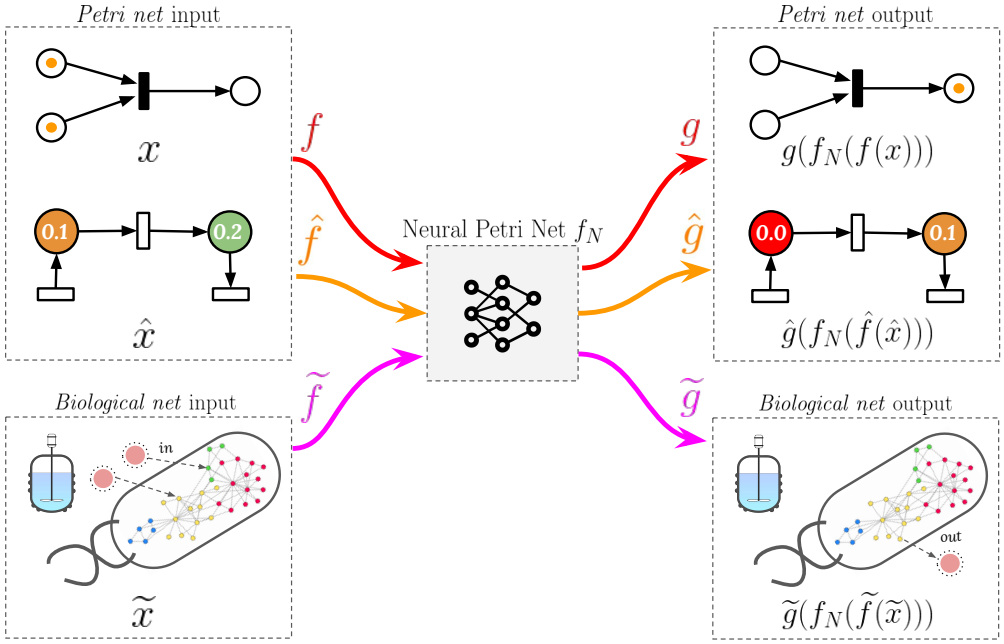


Figure F.3 Neural Petri Nets following a Neural Algorithmic Reasoner blueprint.

enabled reactions in the input network (assuming no conflicts in the firing). Here f, $\hat{f}$ are encoder networks that map input markings and networks parameter to the high dimension input of f_N and g, $\hat{g}$ are decoder networks that map the high dimensional output of f_N to Petri Nets again.

2. When given biological network data corresponding to natural inputs, choose encoder $\tilde{f}$ and decoder $\tilde{g}$ that match input $\tilde{x}$ and output $\tilde{y}$ dimensionality.

3. Swap f, $\hat{f}$, g, $\hat{g}$ with $\tilde{f}$, $\tilde{g}$ and train their weights with gradient descent while keeping f_N gradients frozen.

Through these steps, $\tilde{f}$ and $\tilde{g}$ learn to map natural features to a high-dimensional space. With this, we achieve to model the natural inputs through the previously trained NPN. This yields a network able to model a system from raw and noisy input through classical Petri Nets models without the need for feature engineering.

Besides the execution of Petri Nets, NPNs are appealing models as they could be potentially extended to also perform inference on the modelled system. As an example, given an input x, the encoder and decoder f and g and the NPN f_N, we can extend the system with function $f_E(f(x), f_N(f(x)))$, whose output values are estimated parameters of the modelled systems.

In the next section, we include results from a NPN training where a Petri Net and a Timed Continuous Petri Net are learnt simultaneously.

F.3 Experiments

F.3.1 Petri Nets

The proposed Petri Nets to be learnt are discrete Petri Nets (PN) and Timed Continuous Petri Nets (TCPN). Discrete Petri Nets have been previously explained in this work. It is assumed that the reader is familiar with them.

Timed Continuos Petri Nets (TCPN) [49] are a tuple $\{\mathcal{N}, \lambda, m_0\}$, where:

- $\mathcal{N}$ is the structure of a Petri Net.
- $\lambda \in \mathbb{R}_{>0}^{|T|}$ is a vector with the speed of each transition.
- m_0 is the initial marking of the network.

In this case, we are assuming infinite server semantics for the firing of a transition t, this is, the flow of t, given marking M is:

$$f(M)[t] = \lambda[t] \cdot \min_{p \in t}\{M[p]/Pre[p,t]\}$$

In the case of a discrete Petri Net, the task to be learnt is, given $\mathcal{N}$ and M_i, estimate M_{i+1}. In the case of TCPN, given $\{\mathcal{N}, \lambda, m_i\}$, the task is to estimate m_{i+1}. For simplicity, we will assume that any enabled transition will be fired and no conflicts exist during the firing.

F.3.2 Learning architecture

To make the notation more readable, in the next lines we will denote $m_i^{(t)}$ the marking of place $i \in P$ at instant time t. The proposed architecture first encodes the marking $m_i^{(t)}$ of each place $i \in P$ and the parameters λ_j of each transition $j \in T$ into the latent space with the encoder function f:

$$z_i^{(t)} = f(m_i^{(t)}) \text{ with } i \in P$$
$$w_j = f(\lambda_j) \text{ with } j \in T$$

We have $Z^{(t)} = \{z_i^{(t)} \in \mathbb{R}^{|P| \times K} |\ i \in P\}$ and $W = \{w_j \in \mathbb{R}^{|T| \times K} |\ j \in T\}$, with K being the latent feature dimension. The NPN f_N then maps all transitions T, the structure of the network $\mathcal{N}$ and the latent features $Z^{(t)}$, W to produce the output latent features $H^{(t)} = \{h_i^{(t)} \in \mathbb{R}^{|P \cup T| \times W} \mid i \in P \cup T\}$:

$$H^{(t)} = f_N(\mathcal{N}, T, Z^{(t)}, W) \tag{F.3}$$

Finally, the output marking $m_i^{(t+1)}$ of each place $i \in P$ is obtained with the decoder network g from the latent output variables:

$$m_i^{(t+1)} = g(h_i^{(t)}) \text{ with } i \in P \tag{F.4}$$

In our experimental setting, we will have a separate encoder f and decoder g for PNs, and a separate encoder $\hat{f}$ and decoder $\hat{g}$ for TCPN. All f, g, $\hat{f}$, $\hat{g}$ will be implemented with simple linear transformation layers.

Model	MSE PN	MSE TCPN
GNN-max [63]	0.91	**0.57**
GNN-sum [63]	3.20	1.57
GNN-mean [63]	5.52	2.88
GAT [59]	**0.53**	1.45

Table F.1 Averaged test MSE obtained in the NPNs experiments based on the model employed.

Given the graph-oriented sense of the task, and the fact that PN firing depends on its neighbour nodes, we propose Graph Neural Networks as a feasible model to learn NPNs. GNNs take a graph as an input along with the features of nodes and edges, which makes them ideal for modelling NPNs.

The chosen GNN for f_N is a general Graph Neural Network (F.2). Since the input is a bipartite graph and therefore the output depends on nodes 2-hops away, we use 2 layers of GNNs, so as to reach 2-hops in the reachability graph (as it can be seen in Figure F.1, n layers of GNN imply aggregation nodes that are n-hops away in the input graph). Regarding the input, since our input networks are bipartite, to discern between places and transition nodes, we have used a one-hot encoding approach and encoded the node type in the features input vector. The marking of the input network is directly included as an input feature for each place. The same field is left as a constant in the transitions feature input vector. In the case of TCPN, parameters λ are included as a feature in transitions feature input. The edge weight of each arc of the Petri Nets is given as the only edge parameter of the networks. For the GNN aggregation function, we consider maximisation, mean and summation. As a comparison, we also include Graph Attention Networks (GAT) [59] for our experiment.

The models were optimised with Adam SGD optimiser [26] with a learning rate of 0.0005. In each case loss was computed with mean squared error (MSE). The embedding size feature used was $K = 64$.

The training was performed with 20 Discrete Petri Nets with predefined values, and 20 TCPN with the weights of the places randomly chosen in the range $[0, 5]$ and speed parameters randomly chosen in the range $[0.001, 5]$. The training was performed in a continual manner, by producing random parameters on each epoch. Each model was trained for 3000 epochs and the best model was chosen based on the validation set MSE. For validation, 3 Discrete Petri Nets and 3 TCPNs with predefined values were used. Finally, for testing purposes, 5 predefined Petri Nets and 5 TCPNs were used. For training and testing, we use 5 consecutive outputs for each net, this is, we fire each net 5 consecutive times and use the values for training.

F.3.3 Results

Here we report the MSE obtained separately on PN and TCPN with the test set of networks. These values were obtained with the model that yielded the lower MSE on the validation set in each case. Table F.1 contains the minimum validation and test MSE obtained for each net type, averaged over 5 steps and over the number of nets.

In the table, we can see that, GNN-max perform better at learning TCPN and that GAT perform well with PN. If we consider the sum of both MSEs, GNN-max yields the best performance when learning both models simultaneously.

A close revision has been made to identify general errors made by the neural networks. Generally, GNN-max and GAT perform better than the rest of the networks because they are able to correctly identify those transitions that are enabled in Petri Nets and can correctly compute the subtraction from the input places and the addition to the output places. GNN-max shows also a great performance in generalising to increasing the number of input or output places, unlike the other types of models.

The main error made by models comes when two transitions are fired simultaneously in Petri Nets and tokens are accumulated in the same destination place. More generally, all the models seem to struggle in the accumulation of values coming from more than one neighbour. More research is, therefore, necessary into layer configurations that can generalise addition while preserving performance in the other simulated tasks.