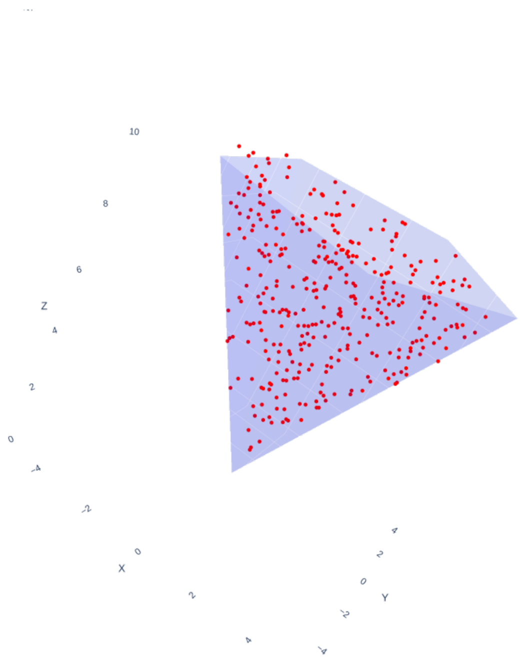
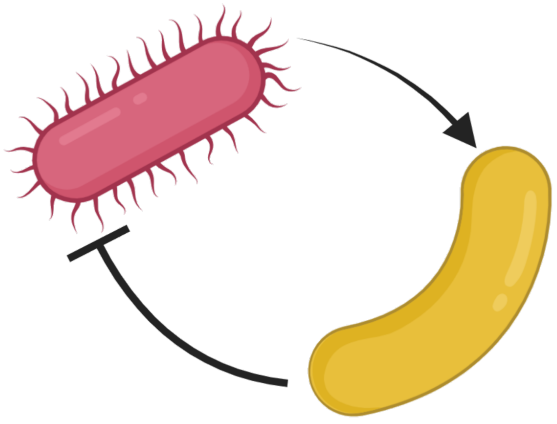
Notes on metabolic modelling analysis

Haris Zafeiropoulos

Jul 01, 2024

CONTENTS





STATUS IN PREPARATION

CHAPTER

ONE

INTRODUCTION

This is a *book* about microbial metabolic models, their reconstructions and analysis at the strain and the community level. It is intended to give only some insight from the user's perspective and not a thorough background on each analysis presented. Yet, the basics will do be shown but mostly *when* to use a type of analysis, *what* can we learn from it, *how* to interpret their results and what are the assumptions made.

The book contains numerous examples *as programs*, including implementations of many concepts. Each chapter is generated from a self-contained Jupyter Notebook. You can click on the "download" button at the top-right of the chapter, and then select ".ipynb" to download the notebook for that chapter, and you'll be able to execute the examples yourself. Many of the examples are generated by code that is hidden (for readability) in the chapters you'll see here. You can show this code by clicking the "Click to show" labels adjacent to these cells.

This *book* is open source, and the latest version will always be available online here. The source code is available on GitHub. If you would like to fix a typo, suggest an improvement, or report a bug, please open an issue on GitHub.

The techniques described in this book have developed out of the study of *data privacy*. For our purposes, we will define data privacy this way:

Definition 1 (M-models)

Genome-scale metabolic models (**M-models**) provide for a metabolic description of genotype–phenotype relationship without accounting explicitly for synthesis of enzymes. M-models employ Boolean logic statements relating genes, proteins, and reactions, or the Gene–Protein–Reaction associations, or Gene-Protein-Reactions (GPRs). A reaction can only carry a non-zero flux if its GPR statement evaluates to True [1].

integrated models of metabolism and expression (ME-Models) account explicitly for the genotype–phenotype relationship. Macromolecular expression is directly integrated with cellular metabolism [1].

CHAPTER

TWO

METABOLIC MODELS

Definition 2 (flux)

(from [2]) The metabolic flux can be defined as the rate at which material is processed through a metabolic pathway. A reaction's flux refers to the **rate** at which the biochemical reaction proceeds in a biological system. It's a measure of *how quickly* reactants are being converted into products within a specific cellular context.

For insightful visualization that may help you with the concept of a flux, you may have a look here.

In a simplified picture of balanced growth, all metabolic processes are balanced: the rate at which material flows into the cell matches the rate at which it is converted, which again matches the production rate of macromolecule precursors. In addition, we assume that these fluxes are constant, such that the whole metabolic network is in a 'steady-state'. Taken together, we thus assume that the metabolic network can take up and produce external metabolites (e.g. extracellular metabolites and macromolecular precursors), but that all internal metabolites ("inside" the metabolic network) are mass-balanced, that is, for each of these metabolites, production and consumption cancel out.

Since each enzyme has a maximal catalytic rate (the k_{cat} value), a reaction flux will require a certain (minimal) amount of enzyme, which takes up cellular space; since cellular space is limited, fluxes cannot increase infinitely since there is always an upper bound on a weighted sum of reaction fluxes. This constraint implies compromises between different reaction fluxes: one flux can only be increased at the expense of others.

Definition 3 (Balanced growth)

Balanced growth is the average state of a cell in a cell bacterial population growing exponentially at the specific (constant) growth rate \$$\mu \geq 0$\$, i.e. the amount of produced biomass per biomass per cell per unit of time.

The mathematical model:

- *variables* to describe: the metabolic **fluxes** in steady-state metabolism,
- *constraints* to apply: the **balance** of production and consumption of all **internal** metabolites

Importantly, the model will be able to describe compromise: for example, with a given carbon influx and assuming mass balance, the carbon atoms can either be used to generate energy **or** biomass; if one function increases, the other one goes down.

To obtain realistic predictions, we may introduce additional constraints, for example known flux directions or experimentally measured uptake rates.

All this information will not suffice to predict metabolic fluxes precisely, but it allows us to narrow down the possible flux distributions.

$$N \times v = 0 = N \times v^+ - N \times v^- = [N \; -N] \begin{bmatrix} v^+ \\ v^- \end{bmatrix} \tag{2.1}$$

The mass-balance constraints in the previous equation, combined with the property that v_i^+ , $v_i^- \geq 0$ can be expressed in the form

$$A \begin{bmatrix} v^+ \\ v^- \end{bmatrix} \geq 0 \tag{2.2}$$

where:

$$A = \begin{bmatrix} N & -N \\ -N & N \\ I & 0 \\ 0 & I \end{bmatrix}$$

The set of constraints on (v^+ , v^-) define a **polyhedral cone** and since they are non-negative, the cone is also pointed, meaning it contains no complete line and the zero vector is the only vertex (extreme point) of the cone.

The space of solutions that satisfies is called the **flux cone**.

2.1 Kinetic models

Kinetic models are typically formulated as a set of deterministic **ordinary differential equations (ODEs)**.

Definition 4 (kinetic variables)

kinetic parameters:

- k_{cat}: It is the maximum rate at which an enzyme can catalyze a specific reaction when it is saturated with substrate. It indicates the number of substrate molecules converted into product per enzyme molecule per unit time under optimal conditions. In simpler terms, it reflects how fast an enzyme can convert substrate into product.
- K_M:
- $\frac{k_{cat}}{K_M}$:

Assumptions used in the formulation of biological network models

Assumption	Description
Continuum assumption	Do not deal with individual molecules, but treat medium as a continuum
Finer spatial structure ignored	Medium is homogeneous
Constant-volume assumption	V is time-invariant, $\frac{dV}{dt} = 0$
Constant temperature	Isothermal systems; Kinetic properties a constant
Ignore physico-chemical factors	Electroneutrality and osmotic pressure can be important factors, but are ignored

The **stoichiometric matrix** (S) represents the reaction topology of a network. For an overview on its characteristics see [3].

Definition 5 (gradient matrix)

(from [3]) Each link in a reaction map has kinetic properties with which it is associated. The reaction rates that describe the kinetic properties are found in the rate laws, $v(x; k)$, where the vector k contains all the kinetic constants that appear in the rate laws. Ultimately, these properties represent time constants that tell us how quickly a link in a network will respond to the concentrations that are involved in that link.

The *reciprocal* of these time constants is found in the gradient matrix G, whose elements are

$$g_{ij} = \frac{\partial v_i}{\partial x_j}$$

These constants may change from one member to the next in a biopop- ulation, given the natural sequence diversity that exists. Therefore, the gradient matrix is a *genetically determined* matrix. Two members of the population may have a different G matrix.

Mathematically speaking, G has several challenging features. Unlike the stoichiometric matrix, its numerical values vary over many orders of magnitude. Some links have very fast response times, while others have long response times. The entries of G are real numbers and, therefore, are not "knowable." The values of G will always come with an error bar associated with the experimental method used to determine them. It has though, the same sparsity properties as the matrix S.

Definition 6 (Jacobian matrix)

S gives us network structure and G gives us kinetic parameters of the links in the network. Their product, the **Jacobian matrix** (J) gives us the network dynamics.

Observation 1

Fluxes are measured in moles per unit of time per cell.

Definition 7 (MASS model)

a metabolic network model that explicitly accounts for the regulatory enzymes, and all their bound states, as components in the network. The result is a data-driven process for constructing mass action stoichiometric simulation (MASS) models that are based on mapping top-down omics data onto bottom-up network reconstructions.

For more about MASS models you may check Palsson's book on dynamic models [4] and one of his papers [5]

CHAPTER

THREE

HOW TO

In this notebook, we will keep track of handy implementations for metabolic modeling related tasks.

3.1 Media and environment setup

3.1.1 Extracellular reactions

COMETS includes the capability to simulate reactions happening in the extracellular environment, without association to a specific organism. Users can implement either elementary reactions of arbitrary order based on mass-action kinetics, or enzyme-catalyzed reactions obeying Michaelis–Menten kinetics, e.g., for the simulation of extracellular enzymes.

3.1.2 Get complete medium for ModelSEED

With the term *complete medium** we describe an *in silico* object where any compound that could be used as a nutrient, it is available for the model.

To build this object for the case of ModelSEED, we need to first get all the possible compounds. And we can do this by first, getting locally the ModelSEEDDatabase repo.

Then we can explore the `Biochemistry` folder of that to retrieve all possible nutrients that could be imported in our model.

From the Biochemistry folder of the dev branch of the ModelSEEDDatabase repository, run:

```
awk -F"\t" '$6 != 1 && $18==0  {print $5}' reaction_*.tsv   > TRANSPORT_REACTIONS.tsv
```

```
  Cell In[1], line 1
    awk -F"\t" '$6 != 1 && $18==0  {print $5}' reaction_*.tsv   > TRANSPORT_
 ↪REACTIONS.tsv
                                              ^
SyntaxError: invalid syntax
```

Now, with something like the following Python chunk, you can build the complete medium and export in a `.csv` file that with the applied format, could be used for gapfilling with the `fill` command of the `gapseq` tool.

```
def write_to_gapseq_format(all_compounds, cpd2name, output_file):
    """
    Write a 3-col csv file with the compound id, its name and a boundary flux of 1000
    """
    with open(output_file, "w") as f:
```

(continues on next page)

(continued from previous page)

```python
        counter = 0
        for compound in all_compounds:
            if compound in cpd2name:
                counter += 1
                f.write(f"{compound}\t{cpd2name[compound]}\t1000\n")
            else:
                print(f"Compound {compound} not found in cpd2name dictionary")

    print(f"Total compounds written: {counter}")


def process_transport_reactions(input_file, output_file=None):
    """
    Parse the TRANSPORT_REACTIONS.tsv file to export compounds that should be part of␣
↪the complete medium.
    """
    with open(input_file) as f:
        lines = f.readlines()

    ex = [line.strip() for line in lines if len(line.split(";")) == 2]

    cpd2name = {}
    all_compounds = set()

    for reaction in ex:
        compounds = reaction.split(";")
        c1 = compounds[0].split(":")[1]
        c2 = compounds[1].split(":")[1]

        if c1 == c2:
            name = compounds[0].split(":")[-1]
            all_compounds.add(c2)
            if c2 not in cpd2name:
                cpd2name[c2] = name

    if output_file is not None:
        write_to_gapseq_format(all_compounds, cpd2name, output_file)

# Main execution
if __name__ == "__main__":
    process_transport_reactions("TRANSPORT_REACTIONS.tsv", "complete_modelseed_medium.
↪csv")
```

CHAPTER

FOUR

SOFTWARE PACKAGES

In this page we will keep a list of the software approaches we are aware of for the various metabolic modeling tasks. Apparently, this list can never be complete, but it can be improved with your contribution! So, feel free to make a PR adding something or contact us to do that for you.

4.1 Reconstruction, gap-filing, validation

- DEMETER
-

4.2 Topological approaches

- QFCA

4.3 Static approaches

4.4 Dynamic approaches

- **dfba** GitLab repo Documentation paper A recent approach for dynamic FBA that considers the solution non-uniqueness.

-COMETS

-BacArena

4.5 Community modelling

- mergem
- PyCoMo
-

CHAPTER

FIVE

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] Bernhard Palsson. *Systems biology*. Cambridge university press, 2015.

[2] Gregory Stephanopoulos. Metabolic fluxes and metabolic engineering. *Metabolic engineering*, 1(1):1–11, 1999.

[3] Bernhard Ø Palsson. *Systems biology: properties of reconstructed networks*. Cambridge university press, 2006.

[4] Bernhard Ø Palsson. *Systems biology: simulation of dynamic network states*. Cambridge University Press, 2011.

[5] Neema Jamshidi and Bernhard Ø Palsson. Mass action stoichiometric simulation models: incorporating kinetics and regulation into stoichiometric models. *Biophysical journal*, 98(2):175–185, 2010.

PROOF INDEX