Practical Bioinformatics

Wagner Section

Exercise Block 1. Introduction of metabolic networks and metabolic network analysis

In this file, we will introduce you to important concepts and methods, which you will be applying to incrementally more complex models in the subsequent exercises. Note that background information is shown in *italics*, and the description of specific exercises is given in roman typeface.

Flux Balance Analysis (FBA)

FBA is a method to predict the flux of material through a metabolic network that maximizes the flux through a target reaction given two kinds of constraints; we will explain both below. The first one arises from the relative proportions of reactants and products in chemical reactions (given in the stoichiometric matrix) and the other is how much flux a reaction is allowed to carry (reaction bounds). FBA is a very powerful tool in the analysis of metabolic networks, and also computationally efficient even for genome-scale models, as it assumes the network is in ***steady state*** *– metabolites do not accumulate, they are produced as fast as they are consumed and consumed as fast as they are produced.*

Stoichiometric matrix

Stoichiometric coefficients represent the relative molar amounts of reactants (educts, substrates) and products participating in a chemical reaction. For example, in the reaction A + 2 B -> C, the stoichiometric coefficient of A is 1, that of B is 2 and that of C is 1. To denote whether a molecule is a reactant or a product of a reaction, we add a sign to its stoichiometric coefficient. Thus, the stoichiometry of A is -1, the stoichiometry of B is -2 and that of C is 1.

Metabolic reaction networks can be very large and comprise hundreds to thousands of reactions. For such large networks, it is useful to represent the network's structure in the form of a stoichiometric matrix. Flux balance analysis (FBA) uses this representation, which is compact and simple. Consider the simple metabolic reaction 'network' consisting of the following three reactions: R1, R2 and R3.

R1: $A + 2\,B \rightarrow C$
R2: $B + C \rightarrow D$
R3: $A + D \rightarrow 2\,E$

This network can be represented in a stoichiometric matrix of m = 5 rows and n = 3 columns, where each row corresponds to one metabolite and each column corresponds to one reaction. The structure of this matrix is shown in the table below.

	R1	R2	R3
A	-1	0	-1
B	-2	-1	0
C	1	-1	0
D	0	1	-1
E	0	0	2

Reactions R1-R3 are irreversible reactions, as indicated by the unidirectional arrows in their stoichiometric equations. To represent a reversible reaction in a stoichiometric matrix, it can be split into two unidirectional reactions. For example, if R1 is a reversible reaction, it can be split into two reactions R1f (forward) and R1b (backward). That is, the stoichiometric equation

R1: A + 2 B ↔ C

would translate into

R1f: A + 2 B → C
R1b: C → A + 2B

Exercise 1.1. Exploring a simple reaction network (see figure 1 below)

a) Draw the stoichiometric matrix manually (feel free to show non-zero entries only).
b) Think of three alternative ways to produce X when A, B and C are present in the extracellular compartment (and when the metabolic network is in a steady state).
c) Now imagine the situation where B and C are not present in the extracellular compartment, but A is. If the flux through reaction R2 takes a value of 10 mmol/gDW/hr, what is the maximum rate at which X can be produced, assuming steady state?

Note that metabolic network analysis often uses different symbols for the same metabolite in two different biological compartments, such as the cytoplasm and the extracellular space. Specifically, X_c and X_e is usually used to denote that metabolite X is in intra- or extracellular space, respectively.

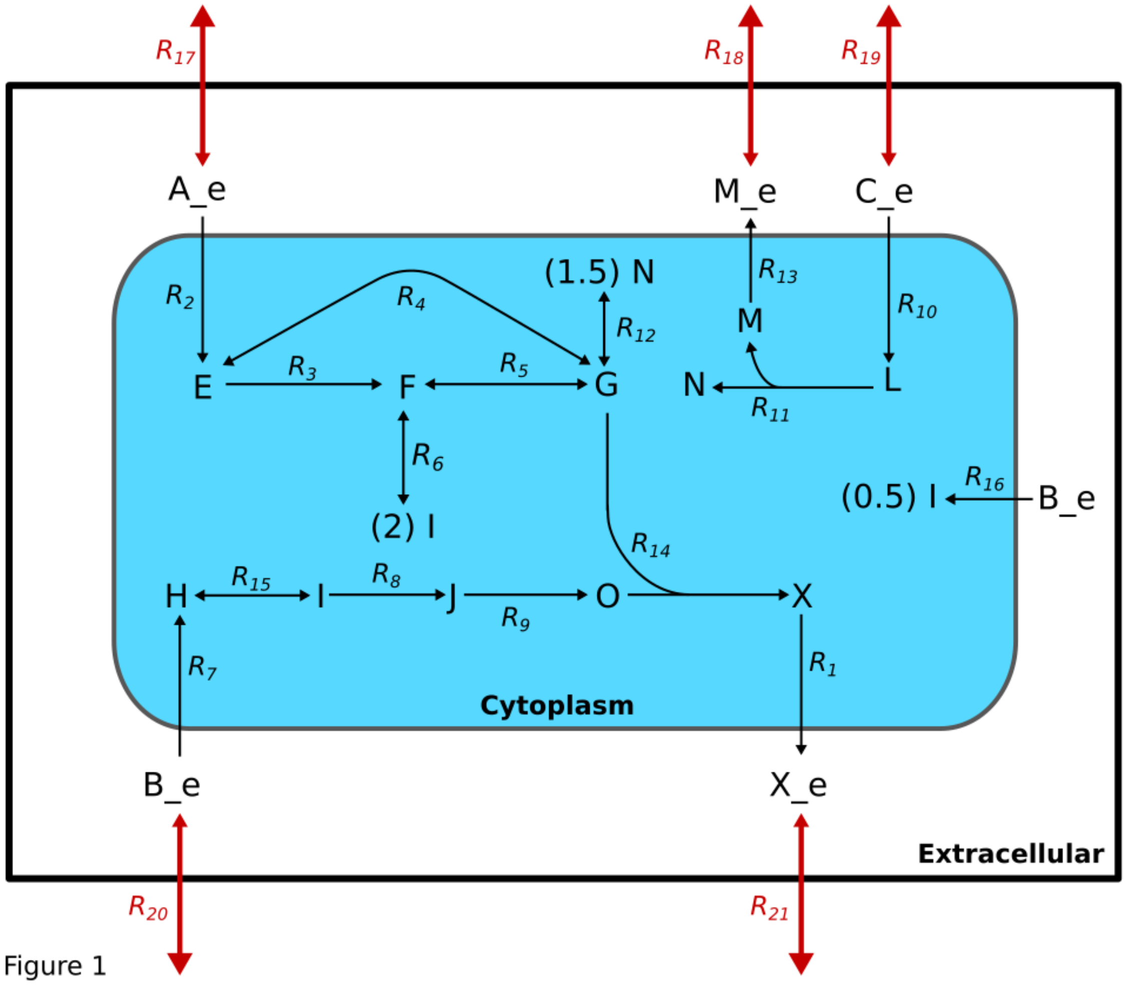

Figure 1: A simple toy metabolic network. Black arrows show internal reactions, red arrows show external reactions. Arrowheads show the direction of possible fluxes; for the external reactions, flux out of the system is the default (large arrowheads), but you can change the lower bound of an external reaction to allow a metabolite to enter the system (small arrowheads).

You have so far performed a metabolic optimization similar to FBA, but manually and with a simple network. An FBA problem can be solved by inspection if the network is as simple as this one, but as a network gets more complex it becomes impossible to solve manually. Let's now try to perform FBA with the assistance of computers. COBRApy is a python package to model biological networks computationally (opencobra.github.io/cobrapy). It allows building and reading models in various formats as well as performing FBA. The basics of how to use COBRApy (version 0.17.1) will be introduced as they are needed in this course. You can find more detailed information in the COBRApy reference provided in this course or you can take a look at the official documentation (cobrapy.readthedocs.io).

Before FBA can be performed, a computational model of a metabolic network must be created in the computer. Models can be easily created and written in standard formats with the aid of COBRApy. The following paragraphs explain how to build a model and *save it in SBML format. SBML (Systems Biology Markup Language) is an XML-based standard format for representing and distributing computational models of biological systems. There is a section explaining how to build a metabolic model in the COBRApy documentation as well.*

Let's build the model for the network used in our initial example, which has 3 reactions and 5 metabolites. For the present example, we will consider R3 to be a reversible reaction:

R1: A + 2 B → C
R2: B + C → D
R3: A + D ↔ 2 E

The python script must start by loading the required packages and defining the model:

```
# Import the cobra package and the required objects:
import cobra
from cobra import Model, Reaction, Metabolite

# Define the model:
cobra_model = Model('ExampleModel_building')
```

Then we need to define the reactions. As stated earlier, if a reaction is reversible, it can be split into two unidirectional reactions. When building a model, however, the reaction is typically defined in one direction only and the directionality is then indicated by setting limits on the flux value. Genome-scale metabolic models (metabolic models including all reactions that can take place in an organism) indicate the directionality of reactions by giving a lower and an upper bound for the flux of each reaction. We will follow the same convention for building our example:

```
# Define the reactions:
reaction1 = Reaction('R1')
reaction1.lower_bound = 0.
reaction1.upper_bound = 1000.

reaction2 = Reaction('R2')
reaction2.lower_bound = 0.
reaction2.upper_bound = 1000.

reaction3 = Reaction('R3')
reaction3.lower_bound = -1000.
reaction3.upper_bound = 1000.
```

The statements above create three reactions with reaction IDs R1, R2 and R3. The reversibility or irreversibility of a reaction is given by the flux boundaries. Reaction 3 can take a positive or negative flux value, which indicates that the reaction is reversible.

Now it is time to introduce the metabolites and linking the metabolites to the reactions (the compartments define where the metabolite is located e.g. 'e' for extracellular or 'c' for intracellular. For the moment, let's not care too much about them and set them all to 'c'):

```
# Define the metabolites:
A = Metabolite('A', compartment='c')
B = Metabolite('B', compartment='c')
C = Metabolite('C', compartment='c')
D = Metabolite('D', compartment='c')
E = Metabolite('E', compartment='c')

# Link ('add') the metabolites to reactions with their respective
stoichiometric coefficients:
reaction1.add_metabolites({A: -1.0,
                           B: -2,
                           C: 1.0})
reaction2.add_metabolites({B: -1.0,
                           C: -1.0,
                           D: 1.0})
reaction3.add_metabolites({A: -1.0,
                           D: -1.0,
                           E: 2.0})
```

The model is still empty; we now must include the reactions in the model:

```
# Add the reactions to the model, which will automatically add all
associated metabolites:
cobra_model.add_reactions([reaction1, reaction2, reaction3])
```

The model is now built. You can check the number of reaction and metabolites included in the model with `len(cobra_model.reactions)` *and* `len(cobra_model.metabolites)`*, respectively. You can also inspect your model further by running:*

```
for x in cobra_model.reactions:
    print(f"{x.id} : {x.reaction} / lb: {x.lower_bound} / ub:
{x.upper_bound}")
```

Once you have checked that the model contains everything you wanted, it is time to save it. This will allow you to work with it later without having to build it again. With the following line, you can save your model in SBML format:

```
# Save the model in SBML format:
cobra.io.write_sbml_model(cobra_model, 'ExampleModel_building.xml')
```

The file is saved in your current working directory. If you want to change the location, just add the specific path before the file name.

In the file ExampleModel_building.py, you have a script with all these commands that create and save the example model.

Exercise 1.2. Building a model including external reactions and an objective function

Now you know the basics of creating a model, it is time to introduce some terminology and conventions that will facilitate working with more complex models in the following exercises.

Metabolic models include artificial reactions that allow the flux of metabolites in and out of the system: ***external*** *aka* ***exchange reactions****. They act on external metabolites, that is, a metabolite that is outside a cell that can be transported in or out of a cell. By convention, the role of an external reaction is to remove metabolites from the metabolic network, so an influx of a given metabolite into the network carries a negative flux. You may have already noticed that external reactions are necessary for FBA to work – metabolites have to be introduced somewhere and taken out of the system somewhere else if there is to be flux with unchanging metabolite concentrations.*

For that very reason, external reactions are a convenient way to specify the kind of environment in which we want to simulate cell growth. By changing the lower bounds of an exchange reaction (remember influx carries a negative flux!) you can define what metabolites are available to a cell. Note that external reactions are different from transport reactions as the latter allow import/export of metabolites into the cell. As you get to more complex metabolic networks, you will see that cells often have multiple ways of taking up or excreting metabolites. For example, in the toy model (Figure 1) you may have noticed that there are two pathways for the uptake of metabolite B_e. If you wanted to change how much of B_e the cell takes up, you don't need to change the bounds of both transport reactions R7 and R16; you can simply change the bounds of the external reaction and leave it to FBA to decide which pathway is used for the uptake of B_e.

Warning: *There is almost never a need to change the upper bound of an external reaction. By default, these are set to 1000 mmol/gDW/hr, which means that any metabolites that are excreted by the cell are removed from the system.*

Modify the script ExampleModel_building.py to include the transport of metabolites A, B and E to the extracellular space and the external reactions. The whole model is now:

R1: A + 2 B → C
R2: B + C → D
R3: A + D ↔ 2 E
R4: A ↔ A_e
R5: B ↔ B_e
R6: E ↔ E_e

R7: A_e ↔
R8: B_e ↔
R9: E_e ↔

Hint: You may want to use `compartment='e'` here.

FBA can be used to maximize the rate at which any given reaction proceeds. Before performing FBA, we must indicate which reaction will be the ***objective function*** *of the FBA problem, that is, the reaction whose rate FBA is going to maximize. The objective function can be set with the command:*

```
cobra_model.objective = 'REACTION ID'
```

For this exercise, we are interested in maximizing the production of E_e. Write the python/cobra statement that indicates that this is the objective function.

Exercise 1.3. Building the model shown in figure 1.

In the file ToyModel_building.py, part of the model has already been created. Complete the missing information and save it in SBML format.

Exercise 1.4. Explore how the synthesis rate of X_e changes as you change the uptake rate of A_e.

The synthesis rate of X_e depends on the rate of substrate uptake, in this case A_e. Increasing the uptake rate of A_e would result in a proportional increase in the synthesis of X_e. Perform FBA to get the maximal production of X_e when A_e is imported into the cell (“taken up”) at a maximal rate of 10 mmol/gDW/hr. Make sure to set the right boundaries on all reactions and to select an objective function. For this exercise, the objective function is the reaction R1 that synthesizes X_e. Changes you make to a model after loading it into a new script will not be saved in the respective file.

For this exercise, the following functions will be useful (for more details, find them on the COBRApy reference):

```
model = cobra.io.read_sbml_model('model_file.xml')

model.reactions.get_by_id('R1').lower_bound = 0.

fba_solution = model.optimize()
print(fba_solution.objective_value)
print(cobra_model.summary())
```

Compare the result to what you got in Exercise 1.1 (c). Now change the uptake rate of A_e and monitor the rate of synthesis of X_e. Is the increase in X_e linear with the increase in A_e? Why or why not?

Essential reactions: Finding reactions that must be there

A reaction is essential for the synthesis of a molecule from a specific substrate if its removal makes the synthesis of that molecule from the substrate impossible. As an example, consider the reactions involved in the synthesis of metabolite O from substrate B_e in figure 1. Synthesis of O from B_e requires reactions R8 and R9, but not R7 and R15, as these can be replaced by R16. If we were to delete either R8 or R9, it would be impossible to synthesize O from B_e. We therefore say that reactions R8 and R9 are essential for the production of O from substrate B_e.

Note: *Due to the way the solver works, COBRApy will at times return numerical results/fluxes through a reaction that are tiny, for example 1.5x10^{-16} (*`1.5e-16`*). This means that there is no flux. You can include (ex 1.5.) or exclude (ex 1.6.) these values by using a threshold of your own choosing, for example* `0.001` *or* `1.e-3`*. Consider using the functions* `round(flux, 3)` *or* `abs(flux)` *instead of simply checking for zero (*`flux != 0`*) – but it is not too important and we will not take away marks if you do the latter. This is also the case for the active reactions below.*

Exercise 1.5. List the reactions that are essential for the production of X from input substrate A_e.

Go over figure 1 and list all reactions that you think are essential for the synthesis of X_e from substrate A_e (assuming that the synthesis rate of X_e is to be maximized.).
Here are a few reactions worth paying special attention to: Is reaction R3 essential? Why or why not? And what about reaction R6?

The following lines allow you to calculate the effect of deleting a single reaction on the maximum possible growth rate after deletion (for each reaction individually):

```
from cobra.flux_analysis import single_reaction_deletion
deletion_results = single_reaction_deletion(model)
```

`single_reaction_deletion()` returns a pandas DataFrame. You can access values in the DataFrame in different ways (see COBRApy reference). Have a look at the full DataFrame, then try to select by condition to get an output which only includes the rows of the essential reactions of the network. This will be particularly useful when we use larger networks.

Now compare your answer with the prediction from flux balance analysis. What do you see?

Note: Depending on your setup, functions from the `cobra.flux_analysis` module will throw an error, in which case try to move your entire script under the if-statement `if __name__ == '__main__':`.

Active reactions: Finding reactions in flux

Active reactions are reactions that have a non-zero metabolic flux, that is, reactions proceeding at a rate different from zero. It is NOT the upper and/or lower bound of a reaction that determines whether it is active, but the effective flux of the reaction in the FBA solution. Therefore, you can only know whether a reaction is active or not after performing FBA.

Exercise 1.6. Identify an active pathway for the synthesis of X_e from A_e in the reaction network of Figure 1.

Study figure 1 to find reactions that must be active for the synthesis of X_e from substrate A_e. Are these reactions different from the list of essential reactions in exercise 1.5? Why or why not?

Once you have run `solution = model.optimize()`, the results can be accessed within the solution object. The property `solution.fluxes` is a pandas Series containing the flux through each reaction, which allows you to find the active reactions.
Compare the list of active reactions obtained in this way with the list you obtained by hand.

Flux Variability Analysis (FVA): Finding alternative pathways

FBA solutions are rarely unique. For example, think about the active pathway you found in exercise 1.6 that produces X_e from A_e. Is this the only possible solution that maximizes production of X_e? Pay particular attention to reactions R3 and R4.
In general, we find that at least some reactions can take a whole range of fluxes without affecting the flux through the objective function. Metabolic networks are thus "flux variable", and FVA can provide some insights into the extent of the network's flux variability. FVA computes the minimum and maximum value of the flux through a reaction while keeping a given objective, such as biomass synthesis, unchanged, thus estimating the range of possible fluxes through a reaction (the reaction's flux variability).

Exercise 1.7. Flux variability analysis

We can use COBRApy to find out the reactions' flux variability.
What do you expect FVA to yield for the rates of essential reactions? What about active reactions?

To perform FVA with COBRApy, we can run:

```
fva_solution = cobra.flux_analysis.flux_variability_analysis(model)
```

The result is a pandas DataFrame containing the minimum and maximum allowable fluxes for each reaction.

*We have already introduced the concept of essential and active reactions, now we will introduce some additional terminology. In the FVA of our model, you can see that various reactions, e.g., R1 and R10, show the same maximum and minimum value. In those cases, we say that a **reaction is fixed**. Reaction R10 is fixed with a flux value of zero, in which case we say the **reaction is blocked**. It cannot be active if the production of X_e is to be maximized.*

With the tool of FVA in hand, revisit the uniqueness of the FBA solution. Is there only one way to produce X_e from A_e? Look at the flux values of reactions R3, R4 and R5. Does the solution make sense biologically?
Now pay attention to reaction R12. Why is this reaction blocked?

Note: Depending on your setup, functions from the `cobra.flux_analysis` module will throw an error, in which case try to move your entire script under the if-statement `if __name__ == '__main__':`.

Exercise 1.8. (OPTIONAL) Calculate the maximal yield of product X_e for each substrate A_e, B_e, and C_e in the network of figure 1.

The maximal yield of a product is defined as the ratio of the rate of synthesis of the product to the rate of utilization or uptake of the substrate. The maximal yield of X_e from substrate A_e can be expressed as Y_{X_e/A_e}.

To compute the synthesis rate of X_e from each substrate, use `model.optimize()`. Make sure to set the boundaries for each reaction appropriately before performing FBA.

Based on the yields Y_{X_e/A_e}, Y_{X_e/B_e}, and Y_{X_e/C_e}, which substrate is the best for the production of X_e? Why? (Hint: consider the roles of metabolites I and N carefully.)

Exercise 1.9. (OPTIONAL) Build the model with the reactions shown below, save it in SBML format and compute the maximal synthesis rate of E_e.

We have seen earlier that FBA can be used to maximize the rate at which any one given reaction proceeds. In the reaction system below, maximize the production of E_e.

R1: A → D
R2: A → E + F
R3: D → 2 E
R4: E → E_e
R5: A_e → A

Assume an uptake flux of A_e of 1 mmol/gDW/hr. List the reactions that must be active and explain why. (**Hint**: Consider which reactions will allow the complete conversion of A_e to E_e. Also, remember you will also need exchange reactions.)