**Assignment: Metabolic Modeling (Week 1)**

06/09/2025

This assignment builds on the Week 1 practical on metabolic modeling in COBRApy. We will use the same COBRApy package in Python and the *E. coli* core model, as well as the online metabolic map.

Please submit your answers per the general instructions for assignments in this course.

**Tasks:**

In this assignment, we will integrate sample gene expression data into the *E. coli* core model. As outlined in the lecture of Week 1, enzyme gene expression data can serve as a (crude) proxy for maximal reaction activity.

The data is contained in the file ‘e\_coli\_core\_expression.csv’ which you can find in Canvas next to these instructions The file contains for each reaction an estimated maximal activity in units of mmol/gDW/h (concentration per time). We discussed in the course how multiple enzymes can be involved in a given reaction; the numbers in the data file represent the effective activity for the reaction in question.

1. Visualize the reaction maximal activity data on the ESCHER map of the *E. coli* core model. For this we use the non-interactive version:

<https://escher.github.io/#/app?map=e_coli_core.Core%20metabolism&tool=Builder&scrollToZoom=true&model=e_coli_core>

In order to load the data, go to “Data 🡪 “Load reaction data” and load the file. The data will then be visualized in the same way that we inspected actual flux predictions in the interactive session in Week 1.

a) In the interactive session, we saw reaction fluxes in a linear pathway being equal to each other due to mass balance constraints. Do you observe the same thing now for the maximal reaction activities? Explain your observations. [5 pt]

a) Some reactions have no maximal reaction activity data. Identify at least two different kinds of such reactions, and explain why gene expression-derived data would not be applicable for these kinds of reactions. [5 pt]

2. We will now establish an enzyme activity-constrained metabolic model in COBRApy. Please implement the maximal reaction activity data in the *E. coli* core model from the practical per the below instructions.

- For reversible reactions, set the lower and upper flux bound to -value and +value, respectively (“value” as the maximal reaction activity).

- For irreversible reactions, only set the maximal flux bound to value and leave the minimal flux bound at zero.

- For reactions without data, leave their default constraints.

- For the glucose exchange reaction (see practical), please remove the maximal absolute flux bound and use the high absolute default bound instead. (We assume the subsequent transporter maximal activity accounts for maximal glucose uptake.)

- For the ATPM energy maintenance reaction (see practical), leave the lower flux as-is, since it describes cellular energy requirements separate from any maximal reaction activities.

Please print a table listing each reaction’s lower and upper flux bound after implementing the above (no formatting requirements, simple printout sufficient). [30 pt]

3. In addition to maximizing reaction rate (“flux”) through some reaction of biological interest, like the biomass reaction, we can use Flux Balance Analysis (FBA) optimizations also to ask: What are the smallest and largest fluxes possible per reaction, given all present constraints in the network? This is known as “Flux Variabiliry Analysis” or FVA. In FVA, we loop over every reaction of interest and once maximize and once minimize its flux via FBA; thus obtaining the permissible flux range for each reaction in the model.

a) Carry out an FVA for all reactions in the model and print out the resulting minimal and maximal fluxes per reaction (simple printout sufficient). [5 pt]

b) Identify any reactions with gene expression-imposed maximal reaction activities, whose permissible flux range in forward direction is nonzero yet comes out less than its upper flux bound. State their number and explain why a set of reactions behaves this way. (You can ignore reaction FORt here which as seen in the practical has a nonconventional direction definition.) [15 pt]

c) How many reactions have a positive minimal flux in the FVA? State their number and explain why a set of reactions behaves this way. [15 pt]

4. a) Carry out an FBA optimization of biomass and report the maximal biomass production rate in the presence of the implemented constraints. [5 pt]

b) Determine the “bottleneck(s)” – i. e. those reaction(s) whose flux reaches its maximum. [10 pt]

c) The model does not use (have non-zero flux for) all reactions in the model constrained with maximal reaction activities. Pick a reaction with maximal reaction activity constraints and zero flux in the optimal solution, and explain from the network context why it carries not carry flux. *(Hint: Use the online demo again.)* [10 pt]

**Command Reference:**

*model = cobra.io.load\_json\_model('e\_coli\_core.json'):* Load model file (in json format).

*model.[hit tab]:* Get context menu with objects and functions in model.

*model.reactions.get\_by\_id():* Access a reactions using its ID as a string, e. g.:

*model.reactions.get\_by\_id(‘PFK’)*

*len(array):* Get number of elements in array.

*solution = model.optimize():* Carry out Flux Balance Analysis optimization on model with current objective function (biomass reaction per default). *‘solution’* is an object containing relevant information, like the flux values obtained in the FBA solution.

*model.reactions.[reaction name].knock\_out():* Knock out a given reaction.

*model.reactions.[gene name].knock\_out():* Knock out a given gene.

*model.reactions.[reaction name].gene\_reaction\_rule:* Boolean GPR rule for a given reaction.

*print(item):* Print item to screen.

*for item in items:*

*[commands]*

Carry out commands for each item in items.

*if [condition]:*

*[commands]*

Carry out commands if condition is true.

*import csv*

*filename = 'e\_coli\_core\_expression.csv'*

*with open(filename, mode='r', newline='') as file:*

*reader = csv.reader(file)*

*for row in reader:*

*[…]*

Code snippet to read in data from csv file.

*FVA\_results = cobra.flux\_analysis.variability.flux\_variability\_analysis(model = model, reaction\_list = model.reactions, loopless = False, fraction\_of\_optimum = 0.0)*

Flux Variability Analysis (FVA) on all reactions in the model, stored in a pandas DataFrame. *‘loopless’* is a parameter not relevant here. The *‘fraction\_of\_optimum’* argument serves to tell the function not to impose any minimal objective (biomass) flux (which is another possible application of FVA).

Please note: The calculation might take a few minutes.