**Exercise 3**

2019-10-04

**Background**

The objectives of this exercise are:

* To get familiar with a MATLAB structure of genome-scale models.
* To realize how undetermined models of metabolism are.
* To be able to find an optimal flux distribution under specific conditions.
* To interpret flux distributions and find trends in metabolism.

To attain these objectives, you will be given a model of central carbon metabolism of *Saccharomyces cerevisiae* (smallModel.mat) and you will have to find the flux distributions of the model, using different objective functions and simulating different conditions. Additionally, you will analyze which secondary metabolites the model is producing and how is it satisfying energetic and charge balance requirements (ATP, NADH and NADPH).

**Book contents:** This exercise covers chapters 18 and 19, together with some topics from chapter 21.

**Additional references:** Nielsen, J., & Villadsen, J. (2011). *Bioreaction engineering principles Third Edition*. *Reactions*(3rd ed.). London: Springer. <https://doi.org/10.1002/ep.670150306>

**Chapter 5!**

**Report:** You must deliver through Canvas an individual report with all questions answered and a detailed analysis, no later than **Sunday 13th of October at 23.55**.

**Requirements:** You must use MATLAB for all growth simulations (and attach your code as an appendix to the final report), but you to use the plotting tool of your preference for results visualization.

**1.- Finding a flux distribution (10 p)**

As a first step, all necessary variables should be uploaded and set into the workspace. Then, attempt to find a flux distribution for *S. cerevisiae* growing aerobically with glucose as carbon source.

As this is a very reduced model of metabolism, essential processes should be captured and the model should be able to connect all the elements in the network (reactions, metabolites), therefore, some sanity checks should also be performed before trying to get numerical simulations.

**1.1.**  **Setting up the model (2p):** Load the model that will be used during this exercise, ‘smallModel.mat’ (representing central carbon metabolism in *Saccharomyces cerevisiae*), and define all relevant variables in the workspace based on the information inside the model (S, b, c, LB and UB). Inspect the various components in the model by what is contained in e.g. model.rxnNames; model.mets; model.metNames and other fields.

A) How many reactions does the model have?

B) How many metabolites?

C) Assuming that the metabolic network does not contain linear dependencies, how many measured fluxes should be required to compute a solution?

**1.2 Sanity checks and consistency checks (2p):**

A) The model contains a field called model.rev which contains information on reaction reversibility. When the model was generated, reversibility information was gathered from the literature and stored in the model.rev field. How many reversible reactions are there in the model? Even while they are indicated as reversible in model.rev, can you also confirm that they are *really*, mathematically reversible in the model?

B) Is this consistent with what has been gather from literature? Get the name of all the reversible reactions (take a look at glycolysis reactions in literature).

C) Introduce the necessary manual modifications in the network to correct the reversibility.

**1.3.  Determined problem (2p):**

A) Considering 1.1C, you should be able to solve as a determined system. Try to do this. What happens? Explain why.

B) Compute the rank of the matrix, what can be concluded?

**1.4.  Undetermined problem (4p):** Use the function linprog(), find the distribution of fluxes that the model predicts with the default objective function (maximizing the growth rate, this has already been set). Note that glucose and oxygen exchanges (‘glcEX’ and ‘o2EX’, respectively) are blocked for consumption. Change that by imposing maximum uptakes of 1 and 10 mmol/gDWh, respectively (both exchange reactions are defined as sinks, i.e. the flux will be negative when the substrate is being consumed by the model). With the obtained flux distribution perform the following analysis:

A) Show all exchange rates in the model as a bar plot (both uptake rates and production rates) and indicate the corresponding units. What can be observed?

B) Look at the ATP reactions. Comment on how much ATP is being produced/consumed in different pathways. Repeat the analysis for NADH and NADPH, and again comment on which pathways are involved in producing and/or consuming the different cofactors. You should include additional figures that support each of your statements.

**2.- Testing objective functions (15 p)**

In this section we will assess different flux distributions that can be computed under the same previous conditions (glucose limitation and aerobiosis), by using the following objective functions:

2.1. Maximizing growth rate.  
2.2. Maximizing acetate production.  
2.3. Maximizing ATP maintenance (reaction ‘ATPX’ in the model).

Use the same glucose uptake in all cases and show all exchange fluxes in a single bar plot. Include this plot in your report and compare the simulation results. In each case, are the results expected? Why?

**3.- Changing growth conditions (25p)**

Finally, we will simulate the organism both in aerobic and anaerobic conditions, and both with glucose and ethanol as carbon sources.

3.1. Adding additional reactions (5p): In order to be able to simulate growth in ethanol, additional irreversible reactions must be included:

1. Alternative cytosolic aldehyde dehydrogenase: Analogous to the already present aldehyde dehydrogenase in the model, but that uses NAD instead of NADP:

Acetaldehyde + NAD → Acetate + NADH

1. Glyoxylate cycle: A variation of the TCA cycle in mitochondria, needed given the irreversibility of pyruvate carboxylase. Consists of 2 reactions:

Isocitrate → Glyoxylate + Succinate

Glyoxylate + Acetyl CoA → Malate + CoA

Note that both reactions can be lumped together to avoid defining a new metabolite.

Update the relevant model components (S, b, c, LB and UB) to account for these extra reactions.

3.2. Comparing growth conditions (20p): With the updated model, simulate the following conditions:

* 1. Glucose as carbon source, aerobic.
  2. Glucose as carbon source, anaerobic.
  3. Ethanol as carbon source, aerobic.
  4. Ethanol as carbon source, anaerobic.

Perform all simulations with the same molar amount of carbon and optimizing growth. Display all exchange fluxes in a single bar plot. Compare simulation results among conditions: include in your discussion which fluxes are higher in each case and why do you think that is from a biological point of view.

- What are the values for the biomass yield [g biomass/g carbon source] in each case?

***4.-In-silico* genetic modifications(Additional 5 p)**

The basics for simulating gene knockouts in GEMs were explained in the lecture on genome-scale models. Metabolism is highly redundant because it needs to be robust enough to proliferate in a changing environment, however, there are some genes that encode for essential functions in cells and their deletion can induce observable global system responses (growth rates, exchange fluxes).

If you want to take this exercise further or to make sure that you get all of the assigned points for this (40 p), you can explore the effects of single gene deletions on the metabolic network.

**4.1** As this is a very simplified model of metabolism, the whole robustness of it might not be captured and essential biochemical functions should be present.

Explore the network and try to find out 3 essential genes (for growth). Explain the metabolic context for each of them and confirm your results in a numerical way.

**Appendix: Possible useful functions in MATLAB for this exercise**

* bar(): Plots a series of vectors (x1, x2, ...) as a bar plot. Usage: bar([x1,x2,...]);
* length(): Returns the length (n) of a vector (b). Usage: n = length(b);
* size(): Returns the size (m,n) of a matrix (s). Usage: (m,n) = size(s);
* find(): Return the indices of the non-zero values in a vector (c). Usage: (i) = find(c);
* linprog(): Used for solving linear optimization problems with an objective function (c), inequality constraints constraints (Aix ≤ bi), equality constraints (Aex = be) and lower/upper bounds (LB and UB, respectively). **Note** that this function always MINIMIZES the objective function. Usage: [x,f,flag] = linprog(c,Ai,bi,Ae,be,LB,UB);
* rank(): Returns the number of linearly independent rows (m) of a matrix (M). Usage: m = rank(M);
* \: Used for solving a determined system Ax = b. Usage: x = A\b;