Genome-scale metabolic model

Table of Contents

0- Installation cobrapy 2

1- Introduction to Genome-scale metabolic model with Escher 2

A- Observation 2

B- First modification 2

2- Structure of a genome-scale metabolic model 3

3- Analysis and hypothesis with a genome-scale metabolic model 3

PART A 3

PART B 3

USED FUNCTION 4

STRUCTURE 5

0- Installation cobrapy

- Install python
- Open cmd
- Type: pip install cobra

1- Introduction to Genome-scale metabolic model with Escher

A- Observation

1- Go to the web page: https://escher.github.io/#/
2- Open « **Escher-FBA** interactive, web-based flux balance analysis »

This page contains the instruction to use Escher FBA, you can always read this page. We will use the different options on the next page.

3- Launch Escher-FBA
4- Can you describe what you are seeing? Whose map is it?
5- What is the objective function here? What does 0.874 mean?
6- We describe different types of reactions. Transport, metabolic and biomass reaction are internal reactions. Exchange, demand and sink reactions are external reactions. Could you identify one reaction from the present types?
7- Which metabolites are consumed by the model?

B- First modification

Click reset between each step to come back to the initial state.

Case 1:

You want to know if *E. coli* can grow without oxygen. What is your conclusion? Compare the result with the previous one.

Case 2:

You want to know if there is a better carbon source for *E. coli* than glucose. Between these compounds:

- Fructose
- Formate
- Ethanol

Can you change the bound of carbon sources (lb = -10)? Write down the objective function flux for each and conclude.

Case 3:

You want to produce a maximum of formate from an *E. coli* strain (you need to change the objective function). We want at least a biomass flux value of 0.305 (lb = 0.305). what do you think of this result?

What do you think of the result you obtain are we sure the model will produce the formate?

What can you conclude about the limit of genome-scale metabolic models and especially the flux balance analysis?

2- Structure of a genome-scale metabolic model

Open file: II-Structure.py

Answer the questions using your knowledge and the list of functions given in this file.

Model: study_model.xml

3- Analysis and hypothesis with a genome-scale metabolic model

Open file: III-Analysis.py

Answer the questions using your knowledge and the list of functions given in this file.

PART A

Model: e_coli_core (to download from BiGG database)

PART B

Select one of the three subjects.

Models:

- Saccharomyces cerevisiae (to download from BiGG)
- Lactococcus lactis (to download from BiGG)
- Klebsiella pneumoniae (to download from BiGG)
- Blautia hydrogenotrophica (to download from VMH)
- Roseburia intestinalis (to download from VMH)

Have fun 😊

USED FUNCTION

deleteNull(*matrix*): Delete all null row in an array. Useful to show only specific values.

getRXNSFromMet(*model, met*): Return reaction id where met is involved.

io.read_sbml_model(*model*) : Import a model from a xml.

info(model): Print information about the model.

medium(*model, medium*): Change the bound of the model to match the given medium.

reactionsType(*model*): Return a list with the reaction type for each reaction in the model. ('trans', 'sink', 'demand', 'exch', 'biomass')

reactionType(*aReaction*): Return the type of reaction from a reaction.

stoicMatrix(*model*): Return the stoichiometric matrix from a model.

Reaction related to a model

aModel.optimize(): Function related to a model to run the flux balance analysis. Result contains:

- objective_value (flux for the objective function)
- fluxes (fluxes for all reactions in the solution)
- status
- shadow_price

aModel.summary(): Function related to a model to show relevant result from flux balance analysis.

aModel.copy(): Create a copy of the 'self' model.

Reaction associated to a reaction or a metabolite

metabolite.summary(): Print information about a metabolite, show how it is produced and consumed (sometime incomplete).

metabolite||reaction.get_by_id(*anID*): return the metabolite|| reaction defined by anID

Common reaction useful with list

***list*.index**(*value*) : return the position of the value in the list.

char.join(*list*) : return a char containing all element from list join by char.

Write an objective: **{reaction1: coeff1, reaction2: coeff2}**