Homework updates / hints

5/9/2025

Packages (till May 2025)

Analysis:

	Install	documentation	Python version
cobrapy	pip install cobra	https://cobrapy.readthedocs.io/en/latest/	3.8, 3.9, 3.10, 3.11
pipeGEM	pip install pipegem	https://pipegem.readthedocs.io/en/latest/	3.8, 3.9, 3.10, 3.11, 3.12

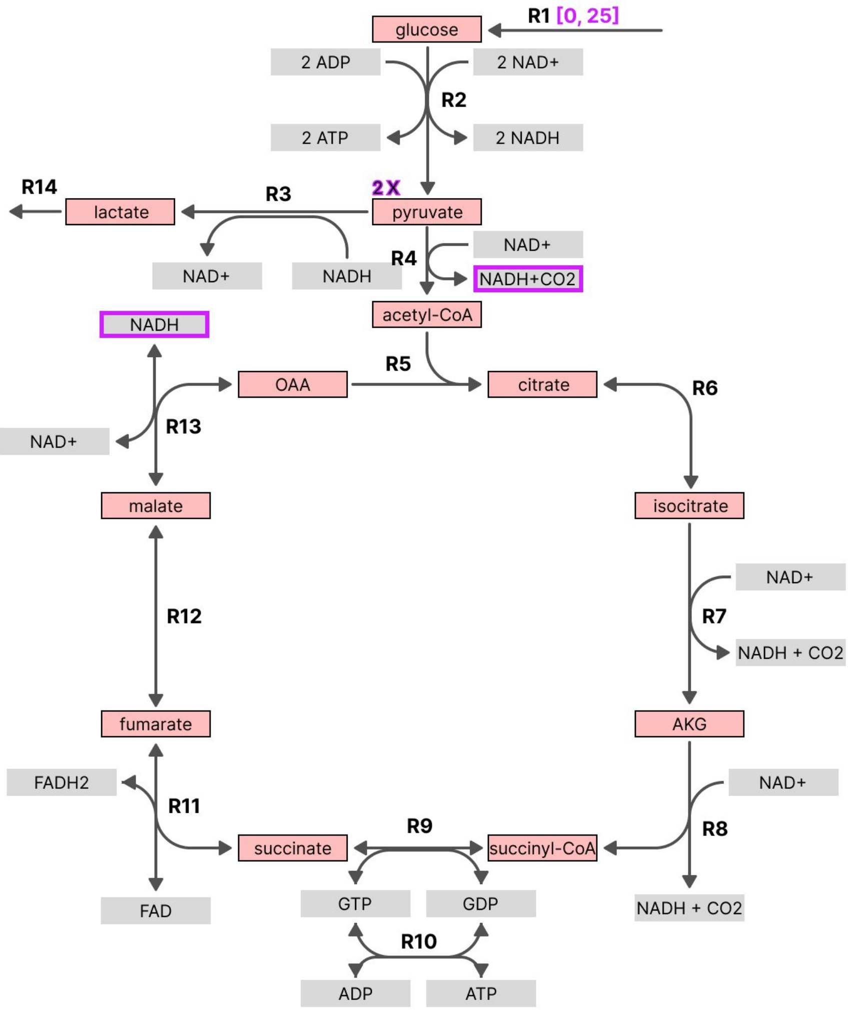
Plotting:

	Install	documentation	Python version
matplotlib	pip install matplotlib	https://matplotlib.org/	3.10, 3.11, 3.12, 3.13
seaborn	pip install seaborn	https://seaborn.pydata.org/	3.8, 3.9, 3.10, 3.11, 3.12

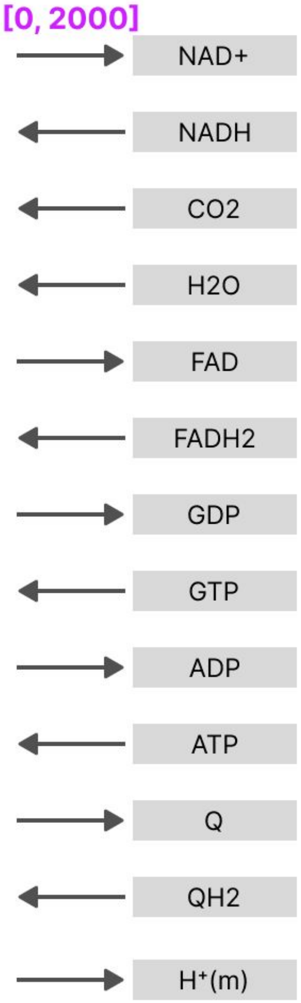
Data:

	Install	documentation	Python version
pandas	pip install matplotlib	https://matplotlib.org/	3.10, 3.11, 3.12, 3.13

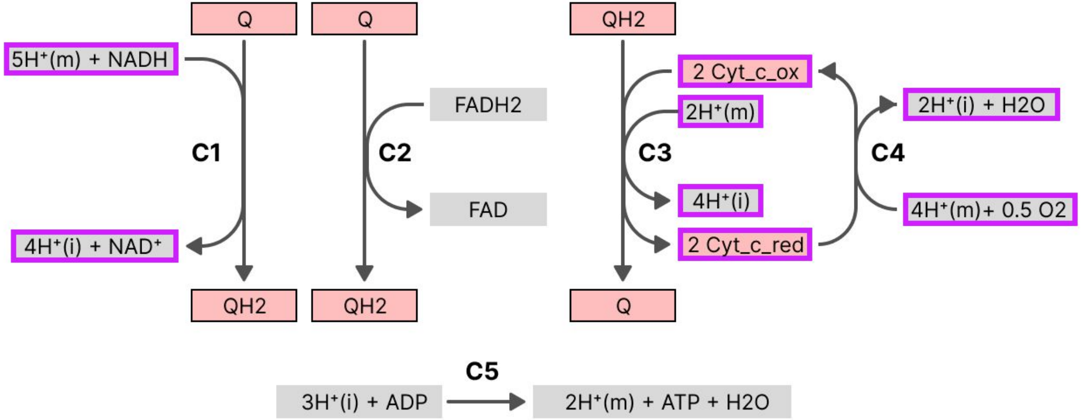
Q1



R1 [0, 25]
glucose
2 ADP
2 NAD+
R2
2 ATP
2 NADH
R14
lactate
R3
2X
pyruvate
NAD+
NADH
R4
NAD+
NADH+CO2
acetyl-CoA
NADH
R5
OAA
citrate
R6
NAD+
R13
malate
isocitrate
NAD+
R12
R7
NADH + CO2
fumarate
AKG
FADH2
R11
NAD+
R9
R8
succinate
succinyl-CoA
GTP
GDP
NADH + CO2
FAD
R10
ADP
ATP



[0, 2000]
NAD+
NADH
CO2
H2O
FAD
FADH2
GDP
GTP
ADP
ATP
Q
QH2
H⁺(m)


Q
Q
QH2
5H⁺(m) + NADH
FADH2
2 Cyt_c_ox
2H⁺(m)
2H⁺(i) + H2O
C1
C2
C3
C4
4H⁺(i)
FAD
4H⁺(m)+ 0.5 O2
4H⁺(i) + NAD⁺
2 Cyt_c_red
QH2
QH2
Q
C5
3H⁺(i) + ADP
2H⁺(m) + ATP + H2O

Q1

Steps:

1. Create the metabolites
2. Create the reactions and connect them to the metabolites (beware of the constraints)
3. Create a model and add the reactions to the model
4. Perform pFBA

Output:

1. A DataFrame storing the pFBA fluxes

Hint:

1. https://colab.research.google.com/drive/1IrYmyCXr61KTQimFn2-JUJontVG4Nn6V

Q2

Steps:

1. Perform FVA on the wild model
2. Calculate the flux range (max - min) of C5
3. Knock out each reaction one by one and do step 1 and step 2
4. If the range is less than the half of C5’s range, record it.

Output:

1. A list of reaction. e.g., [R1, R2, R4, …]

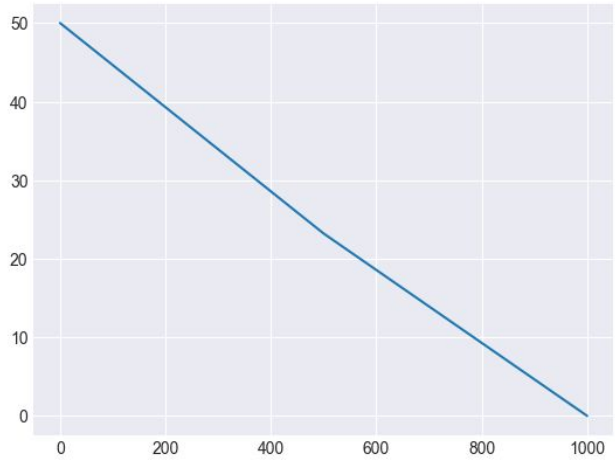
Hint:

1. Use context manager (with model: …) to restore KO quickly (for step 3)
    a. https://cobrapy.readthedocs.io/en/0.19.0/deletions.html
2. Use rxn.knock_out() to KO reactions

Q3

Steps:

1. Create a upper bound list X.
2. Iterate the list X and assign the upper bound to the adjusted reaction (C1, C3, or C5).
3. For each iteration, perform pFBA, obtain R14's flux, and store the value in a list Y.
4. Plot the X and Y in a graph.



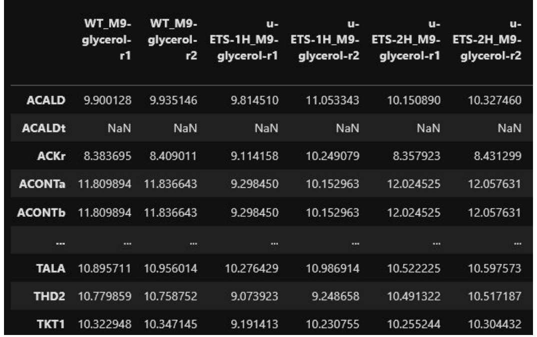
Output:

1. Four figures (C1, C3, C1+C3, and C5)

Hint:

1. See the figure on the right

Q4



	WT_M9-glycerol-r1	WT_M9-glycerol-r2	u-ETS-1H_M9-glycerol-r1	u-ETS-1H_M9-glycerol-r2	u-ETS-2H_M9-glycerol-r1	u-ETS-2H_M9-glycerol-r2
ACALD	9.900128	9.935146	9.814510	11.053343	10.150890	10.327460
ACALDt	NaN	NaN	NaN	NaN	NaN	NaN
ACKr	8.383695	8.409011	9.114158	10.249079	8.357923	8.431299
ACONTa	11.809894	11.836643	9.298450	10.152963	12.024525	12.057631
ACONTb	11.809894	11.836643	9.298450	10.152963	12.024525	12.057631
...	...	...	...	...	...	...
TALA	10.895711	10.956014	10.276429	10.986914	10.522225	10.597573
THD2	10.779859	10.758752	9.073923	9.248658	10.491322	10.517187
TKT1	10.322948	10.347145	9.191413	10.230755	10.255244	10.304432

Steps:

1. Download the GSE291717_sysETS_TPM_File
2. Convert the value to log2 (TPM + 1) (using numpy)
3. Load the textbook model and use it to initiate pipeGEM.Model
4. Create a GeneData for each sample (column) and add it to the model (add_gene_data method)
5. Find reaction score via model.gene_data[c].rxn_scores and concat them into a data frame.

Output:

1. A DataFrame, a dict of series, or a dict of reaction scores for each model.

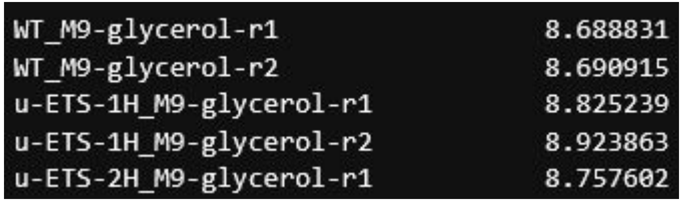
Hint:

1. https://pipegem.readthedocs.io/en/latest/tutorial/2_data.html#maping-gene-data-to-a-pipegem-model

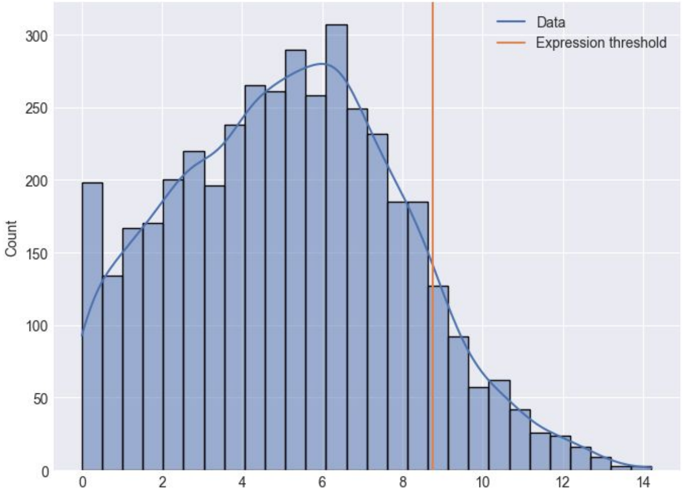
Q5

Steps:

1. Use GeneData.get_threshold(method='percentile') to get the threshold result
2. Summarize Result.exp_th into a Series like:



```
WT_M9-glycerol-r1              8.688831
WT_M9-glycerol-r2              8.690915
u-ETS-1H_M9-glycerol-r1        8.825239
u-ETS-1H_M9-glycerol-r2        8.923863
u-ETS-2H_M9-glycerol-r1        8.757602
```



Output:

1. A pandas series

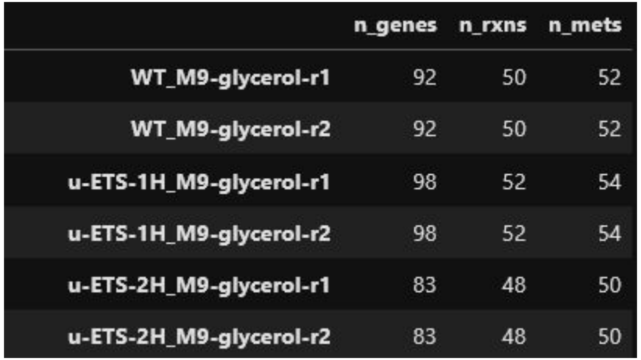
Hint:

1. https://pipegem.readthedocs.io/en/latest/tutorial/2_data.html#finding-data-thresholds

Q6

Steps:

1. For each sample in the TPM data, use model.integrate_gene_data to perform GIMME.
2. Summarize their numbers of reactions, genes, and metabolites, like this:



	n_genes	n_rxns	n_mets
WT_M9-glycerol-r1	92	50	52
WT_M9-glycerol-r2	92	50	52
u-ETS-1H_M9-glycerol-r1	98	52	54
u-ETS-1H_M9-glycerol-r2	98	52	54
u-ETS-2H_M9-glycerol-r1	83	48	50
u-ETS-2H_M9-glycerol-r2	83	48	50

Output:

1. A pandas dataframe

Hint:

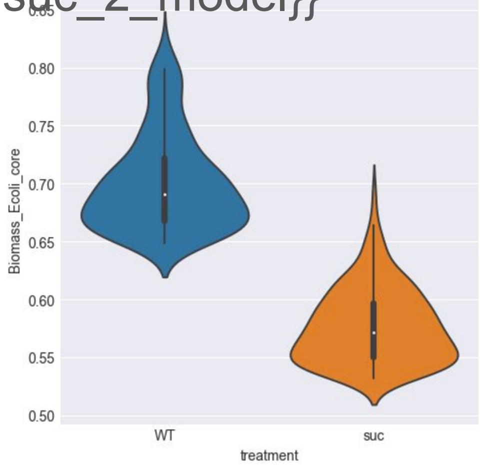
1. https://pipegem.readthedocs.io/en/latest/tutorial/4_MEM.html#gimme
2. https://pipegem.readthedocs.io/en/latest/tutorial/1_basic.html#create-group-object

Q7

Steps:

1. If you use pipeGEM, create a pipeGEM.Group specify the treatment groups:
   a. {‘WT’: {‘wt1’: wt1_model, wt2_model},
   ‘succinate’: {‘suc_1’: suc1, ‘suc_2_model’: suc_2_model}}
2. Do sampling and obtain a result obj
3. Plot the violin plot using the Result.plot

Output: A Figure like ------------------------------------->

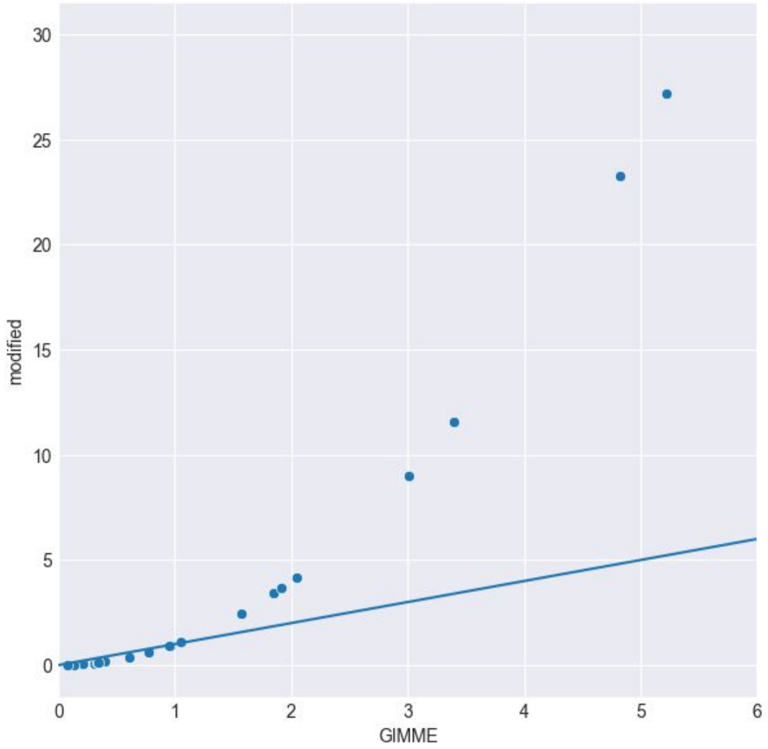


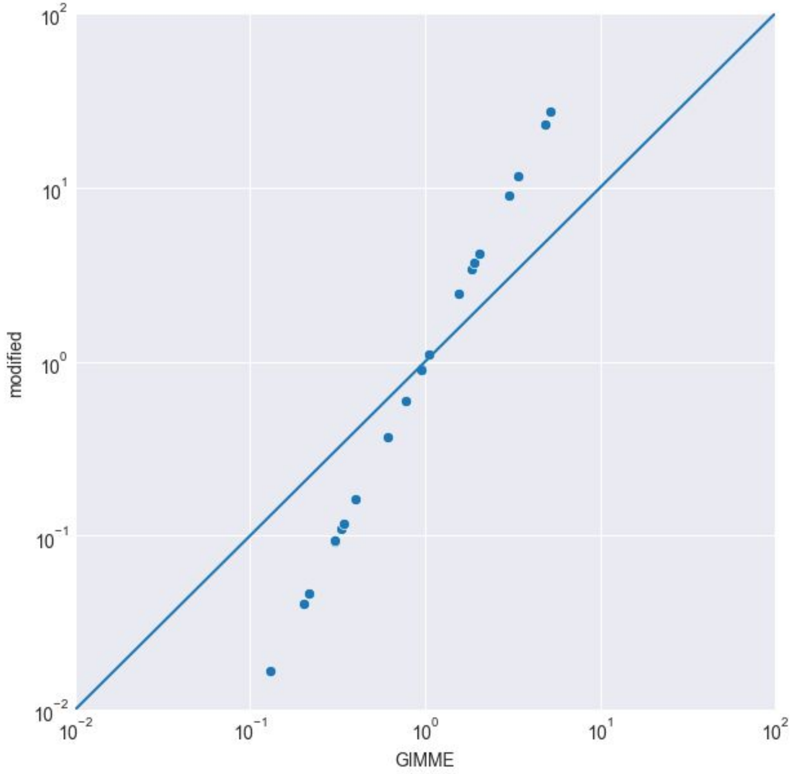
Hint:

1. https://pipegem.readthedocs.io/en/latest/tutorial/1_basic.html#create-group-object
2. https://pipegem.readthedocs.io/en/latest/tutorial/5_flux_simulation.html#randomized-sampling

Q8

Concept: modify the penalty coefficient of GIMME



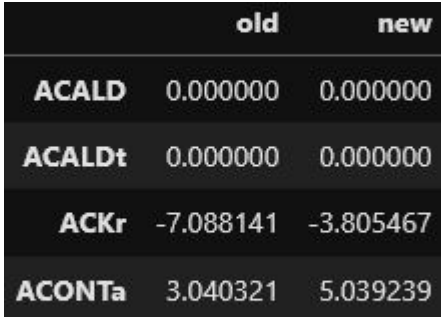


Q8

Steps:

1. Get WT_M9-glycerol-r1’s rxn scores and its p90 threshold (T)
2. For each reaction score S, check if it is not None and if it is less than a threshold T. If both conditions are true, compute the square of the difference between T and S, that is (T−S)^2, and store this value in a dictionary (obj_dic).
3. Use add_mod_pfba(model.cobra_model, weights=obj_dic) function to add pFBA constraint, the model is a pipeGEM model.
4. Minimize the total flux via model.optimize("minimize")
5. Compare the GIMME fluxes of the result model with the model you built in Q6

Output: A dataframe --->



	old	new
ACALD	0.000000	0.000000
ACALDt	0.000000	0.000000
ACKr	-7.088141	-3.805467
ACONTa	3.040321	5.039239

Hint:

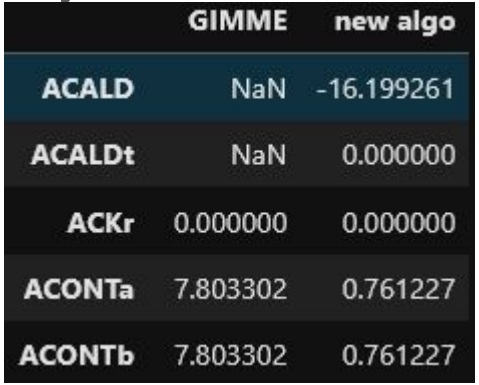
1. add_mod_pfba can be import from pipeGEM.analysis

Q9

Steps:

1. Identify candidate reactions for knockout by selecting those with a Reaction Activity Score (rxn_score) below the predefined expression threshold.
2. Iteratively attempt single knockouts, starting with the reaction that has the lowest rxn_score.
3. After each knockout attempt, test model consistency by verifying whether the model can still produce nonzero flux through the objective reaction.
4. Permanently remove the reaction only if this consistency is maintained.
5. Compare the pFBA fluxes of the result model with the model you built in Q6

Output: A dataframe --->



	GIMME	new algo
ACALD	NaN	-16.199261
ACALDt	NaN	0.000000
ACKr	0.000000	0.000000
ACONTa	7.803302	0.761227
ACONTb	7.803302	0.761227

Hint:

1. You can use mod.slim_optimize() to test if the flux is non-zero.