FBA analysis

Exercise 1:

1)

* Define reactions
* Define metabolites
* Add metabolites to reaction
* Add reactions to model
* Print model reactions and info
* Save model

2)

* Add external reactions (exchange reactions)
* Set objective function

4)

* Set exchange reaction to 0
* Create fba and see results (round values)

5)

* Find essential reactions (single\_reaction\_deletion(model))

6)

* Find active reactions (after optimize()) solution.fluxes

7)

* Flux variability analysis (cobra.flux\_analysis\_variability\_analysis(model)), fixed reactions, blocked reactions

8)

* Maximal yield (rate of synthesize of the product to the rate of utilization or uptake of the substrate)

Exercise 2:

1)

* Load in data, list reactions, list metabolites
* Which is the objective function? (model.objective.expression)
* Why maximize biomass function

2)

* Use single\_reaction\_deletion to see if essential (essential reactions are often fixed)

3)

* Access summary of each metabolite like (cobra\_model.metabolites.nadh.summary())

4)

* Fva with growth flux constraint to maximum value (can be set to any number)

5)

* Look at external reactions (model.exchages)

6)

* Find growth limiting nutrients. Compute growth with different values of glucose uptake
* Plot values for different uptake rates

7)

* Plot uptake rates of other ressources as function of glucose uptake

8)

* Compute growth rate with acetate as only carbon source

9)

* Look which metabolites contain carbon, possible carbon sources

Exercise 3)

1)

* Identify which molecular component of a cell need greatest amounts of resources

2)

* Find subsystem of all essential reactions

Exercise 4)

1)

* Number of reactions and metabolites
* Active reactions, active external reactions

2)

* Essential reactions for glucose minimal environment
* Fraction of essential reactions

3)

* Essential reaction on acetat minimal
* Fraction of essential reaction

4)

* Reactions essential unique to acetat

5)

* Use csv file to look at subsystem
* Count subsystem appearances