MetaFlux in Pathway Tools

Mario Latendresse
Markus Krummenacker

SRI International

Oct 17 – 18, 2013

Outline

The FBA Tool in Pathway Tools

1. The FBA Tool, MetaFlux, was introduced in version 15.0 of Pathway Tools (Feb 2011)
2. MetaFlux has three modes: solving, development, and gene knockout
3. Solving mode: compute the fluxes of reactions to produce the biomass
4. Development mode: trying different biomass, nutrients, secretions, and reactions to create a model
5. Gene knockout: deactivating gene(s) from the model and see the effect on growth (testing a model)

What is a Flux of a Reaction?

- Fluxes are rate of reactions typically expressed as mmol per gram dry weight per hour, denoted mmol/gDW/hr
- Other units could be used (eg, mol instead of mmol)
- The FBA tool does not assume any unit for the fluxes as this is not needed to get valid results
- Solving an FBA model gives the fluxes of all reactions that are needed to create a non-zero flux for the biomass (set of metabolites necessary for growth)

Creating an FBA Model vs Solving an FBA Model (1)

Creating a Flux Balance Analysis (FBA) Model

Creating an FBA model consists in curating the description of an organism such that it represents as accurately as possible the *in vivo* reaction fluxes under certain conditions.

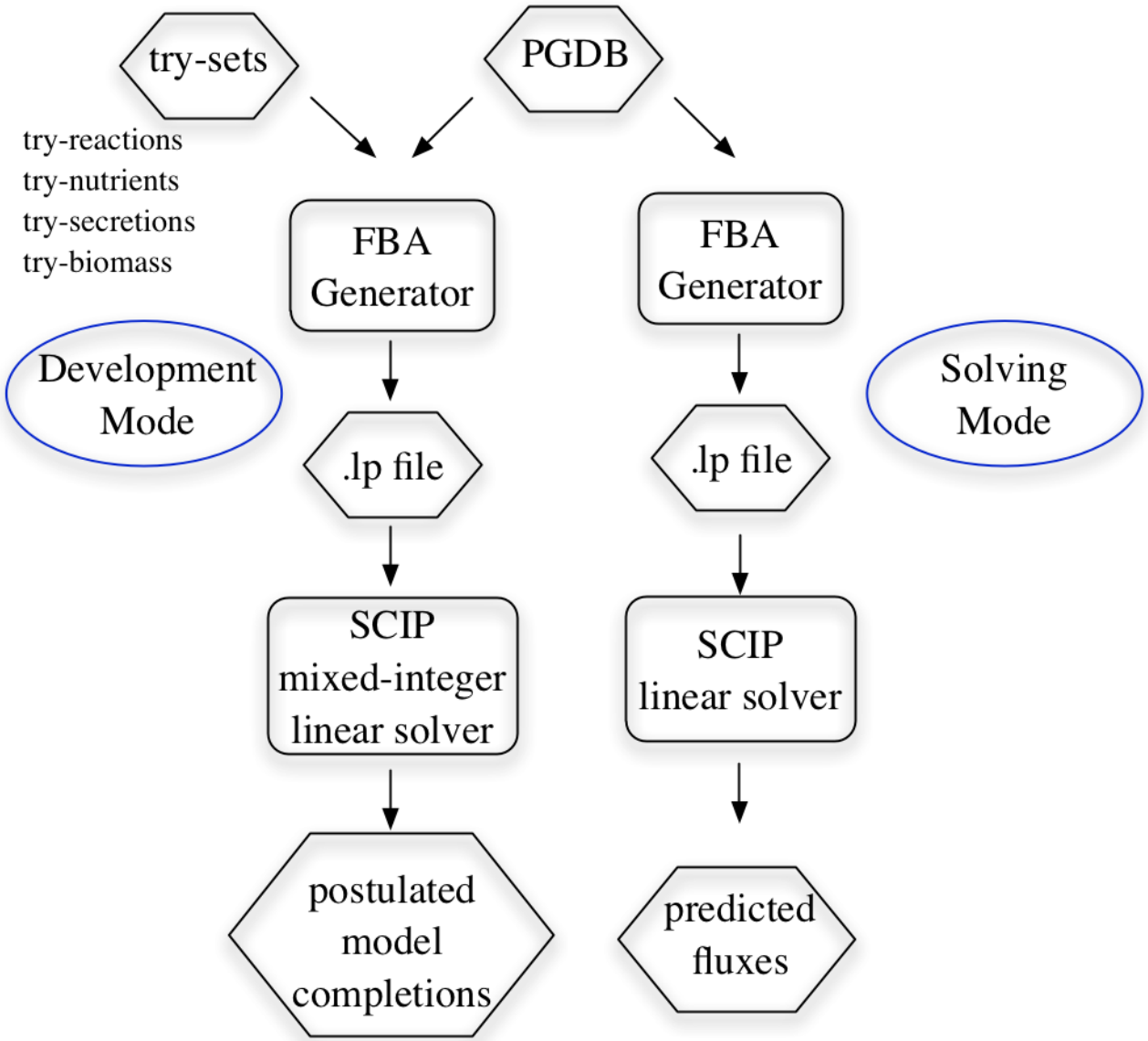
Solving an FBA Model

Solving an FBA model computes the reaction fluxes under certain conditions. The model could be infeasible: no biomass produced.

Gene Knockout

A gene knockout deactivates the reactions catalyzed by a gene and solving such a model. We can verify an FBA model by comparing the results (growth/no growth) of a gene deletion with experimental data.

Creating an FBA Model vs Solving an FBA Model (2)



What is Flux Balance Analysis (FBA)?

Computing Fluxes of Reactions for Organism Growth

Given a network of biochemical reactions, nutrients and secretions, assign a flux (a numerical value) to every reaction to produce a set of biomass metabolites for growth. Maximize the biomass.

Main Assumptions

- The system is in a steady state (metabolite concentrations do not vary)
- Regulation is ignored
- Cofactors are ignored
- Compartments are not completely taken care of
- Some transport reactions must be explicitly specified (e.g., ATP synthase)

Standard FBA Mathematical Formulation

Main Formulation

$$\text{Max } b_{\text{biomass}}$$

$$\mathbf{S}_{ij}\mathbf{v} = 0$$

where $\mathbf{S}$ is the stoichiometric matrix

S is a matrix where each row represents a metabolite and each column represents a reaction.
b_{biomass} is the flux for the biomass reaction.
v is a vector of variables representing the fluxes (real numbers).
Reactions must be mass balanced.

An Example of an LP Formulation

The Reactions, Biomass, Nutrients C, D, E

R_1 : $A + 2B \rightarrow Biomass$
R_2 : $C + D \rightarrow 2A$
R_3 : $D + E \rightarrow B$

The Linear Program (LP)

A : $2R_2 - R_1 = 0$
B : $R_3 - 2R_1 = 0$
C : $R_C - R_2 = 0$
D : $R_D - R_2 - R_3 = 0$
E : $R_E - R_3 = 0$
$Maximize\ R_1$
$0 \leq R_C, R_D, R_E \leq 100$

The Biomass Reaction

It is a virtual reaction representing a set of metabolites to be produced to enable growth.

$$c_1 B_1 + c_2 B_2 + \cdots + c_n B_n \rightarrow Biomass$$

The coefficients c_i are integers (positive or negative).
That reaction is in the **S** matrix (as any other reaction).
Note: **all B_i metabolites must be produced by some other reactions in the organism to satisfy the LP formulation with a non-zero biomass**.

Solving the FBA Formulation

- Solving such a formulation is done by a Linear Programming (LP) solver
- There are many open source and commercial LP solvers: CPLEX, GLPK, SCIP, Gurobi, and more
- Pathway Tools 15.0 uses SCIP
- Even with thousands of reactions, typical FBA/LP formulation can be solved in a few seconds

Syntax of the MetaFlux Input File

- The input to MetaFlux is a text file (a `.fba` file)
- Metabolites are specified as list of names or frame-ids or use `all-compounds`
- Reactions are specified as frame-ids (common) or reaction equations (rare)
- The keyword `metab-all` is the set of all metabolic reactions in the PGDB
- The keyword `metacyc-metab-all` is the set of all metabolic reactions in MetaCyc
- Also, keywords `transport-all` and `metacyc-transport-all`
- Full documentation in the FBA Chapter of the Pathway Tools' User Guide
- Let's look at the `bsubcyc1.fba` file

- A graphical user interface (GUI) is used to submit an FBA input file
- The output files (e.g., `.sol`) are displayed as a text file via a browser
- The Cellular Omics Viewer can be invoked via one click to show the resulting fluxes
- The invocation of the MetaFlux GUI is under the Tools menu
- GUI demo using the `bsubcyc1.fba` file

Four Files Generated

Four files are generated when submitting a model.

- A `.lp` file: the input to the SCIP solver
- A `.log` file: a trace of the output of the generation phase and of the SCIP solver. It contains the quality of the solution, the reactions that were filtered out (e.g. unbalanced or might be unbalanced).
- A `.sol` file: a summary of the solution found as a text file. **This is the file to look at first**. Contains metabolites produced, nutrients and secretions used, added reactions, fluxes for all reactions, and reactions with zero flux.
- A `.dat` file: an omics data file for the Pathway Tools Cellular Overview. Contains only reactions with non-zero flux. Gap-filled reactions are not included.

The Solution File

The solution files list the:

- fixed- and try-biomass metabolites that could be produced
- fixed- and try-nutrients used
- fixed- and try-secretions produced
- added reactions from MetaCyc (reversed or not)
- added reversed reactions from the PGDB
- reactions with their non-zero flux
- remaining reactions that have zero flux

The `.log` File

- Contains warnings and possibly error messages about unbalanced and instantiated reactions
- Each reaction that is unbalanced or might be unbalanced is not included in the model. The list of these reactions are given in that file.
- The process of instantiation of reactions is summarize in that file
- More on instantiation of reactions in Markus' presentation

The `.dat` File

- Each reaction having a flux is listed in that file
- The file can be used as is by the Cellular Omics Viewer of Pathway Tools
- The file can be used on the Web or the Desktop mode
- The Cellular Omics Viewer is directly accessible from the MetaFlux GUI

The `.lp` File

- The `.lp` file is the input of the solver. The complete specification of the model is in that file.
- The file has three parts: the objective function (`maximize obj`), the constraints (`Subject to`), the list of variables with their lower and upper bounds (`bounds`).
- There are comments in that file
- Each metabolite produces one constraint: the constraint contains all reactions where that metabolite is either produced or consumed. Each constraint says that production and consumption of a specific metabolite must balance
- For MILP, the objective function can be very large (over 10000 terms); for LP, this is a single term

Testing a Model Using Genes Knockout

Knocking Out One Gene

- Knocking out a gene means to deactivate the reactions catalyzed by that gene
- Isozymes are taken into account

Multiple Kockouts

More than one gene might be knocked out simultaneously

Batch Knockouts

Typically, MetaFlux is used to run a batch of gene knockouts (e.g., all genes)

Knockout Parameters in FBA Input File (1)

- The parameter `knockout-genes` gives the genes (names or frame ids) to knockout, not necessarily simultaneously
- We can specify `metab-genes` for all genes that are involved in metabolic reactions of the PGDB
- The parameter `knockout-nb-genes` gives the number of genes to knockout from the set `knockout-genes`
- If `knockout-nb-genes` is 1, it is a single-gene knockout, if it is 2, it is a double-gene knockout, and so on
- Note that a double-gene knockout for all genes is over 500 thousands knockout experiments if you have, say, 1000 genes

Knockout Parameters in FBA Input File (2)

- Parameter `knockout-summary-only` controls the number of `.sol` files generated
- If it is `yes` then only one `.sol` is generated summarizing the gene knockout solutions
- If it is `no` then a `.sol` file is generated for each gene knockout run, plus the summary
- Each file has the suffix 'knockout-n' where n is an integer starting with 0
- Saying `no` could generate thousands of `.sol` files when using `metab-genes` for `knockout-genes`

Knockout Parameters in FBA Input File (3)

- The parameter `knockout-reactions` can be used to specify explicitly the reactions to deactivate without specifying genes
- The parameter `knockout-nb-reactions` is used to specify the number of reactions to deactivate simultaneously
- These two parameters can be combined with the parameters for knocking out genes

Knockout Parameters in FBA Input File (4)

- Examples of gene knockout run on EcoCyc for
  1. A few genes: cysN, cysD, gltX
  2. All metabolic genes with summary solution file only (takes about one minute)
  3. All metabolic genes with all solution files generated (takes more than one minute)

Minimal Nutrient Sets

Definition of a Minimal Nutrient Set

A minimal nutrient set is a sufficient set of nutrients to have growth and from which any nutrient removed results in no growth. It is likely that there are many minimal nutrient sets, of various sizes, for one organism and one biomass reaction.

Minimum Set of Nutrients

A minimal nutrient set is not necessarily the smallest set of nutrients. A minimum set of nutrients is a minimal set having the smallest number of nutrients among all minimal sets.

Verifying Minimal Sets for EcoCyc

- Let's assume a simple biomass reaction, and the current set of reactions in EcoCyc: we will not add reactions to EcoCyc
- We can infer from biological knowledge some small sets of nutrients
- We will need to find out which secretions are necessary to use these minimal sets

Finding the Right Secretions

- The `try-secretions` parameter can specify a set of secretions to try. We can specify `all-compounds` to try them all.
- The `try-secretions-weight` should be negative, say -10, so that it *cost* something to add a secretion to the model
- MetaFlux will report which secretions are needed to have growth, given the biomass reaction, the nutrients, and the set of reactions of EcoCyc
- MetaFlux could have no feasible solution, that is no sets of secretions that it can add to generate growth

Infeasible Formulation

- FBA/LP formulation is infeasible: no solution or the only solution is a zero flux for the biomass
- Infeasibility is likely due to some metabolites in the biomass that cannot be produced
- This might be due to missing reactions, nutrients, secretions, or a combination of these
- **Gap-filling proposes model modifications of minimal cost**

If Infeasible with All Secretions Tried

- Assuming that even with all secretions tried, no feasible solution is found
- We could find out which subset of biomass metabolites can be produced
- Specify all the biomass metabolites as `try-biomass`
- Specify a large positive weight (a gain) for `try-biomass-weight`, say 1000
- **MetaFlux can apply multiple gap-filling simultaneously**

Controlling the Weight Parameters

- A weight is a positive or negative integer
- A positive weight is a gain, whereas a negative weight is a cost
- MetaFlux always try to maximize an objective function: if something (e.g. secretions) has a cost, MetaFlux minimizes their use, if it has a gain, it maximizes their use or production (biomass)
- The absolute value of a weight is not very important, but the relative weight values are important
- Consider a gain of 1000 for one biomass metabolite vs the cost of 10 (weight -10) for a secretions. Consider a gain of 10 for one biomass. What is the difference in possible solutions?
- Let's look at a .lp MILP file to better see what is mathematically happening

Single and Multiple Gap-Filling

Typically "Gap-Filling" Means "Completing the Reaction Network"

- Gap-filling adds reactions from a reference database (e.g., MetaCyc) to the FBA model to produce missing biomass
- Model might still be infeasible due to a lack of reactions in MetaCyc, or lack of nutrients, or secretions

Solution: Gap-Filling Extended to Important Metabolites

Nutrients, secretions, and biomass metabolites can also be added or removed. For biomass metabolites, we try to include as many as possible while still getting a feasible solution.

Multiple Gap-Filling

Multiple Gap-Filling

Multiple gap-filling is done on reactions, nutrients, secretions, and biomass metabolites **at the same time**.

Objective

Try to add as many biomass metabolites as possible by adding a minimum number of nutrients, secretions, and reactions; and still get a feasible solution.

Usage

Speeds curation of a PGDB. It is a technique to complete a PGDB to do standard FBA analysis.

Our multiple gap-filling extends the reaction gap-filling idea developed by Costas Maranas.

Linear Programming Becomes Mixed-Integer Linear Programming (MILP)

The LP formulation becomes a Mixed Integer Linear Program (MILP): binary variables control the addition of reactions, nutrients, biomass, and secretions.
A constraint to control the flux r_i of a reaction with a binary variable s_i:

$$r_i - s_i 1000 \leq 0$$

When s_i is 1, the reaction r_i can have a non-zero flux. And that s_i add a cost or gain in the objective function to maximize.
The biomass, secretions and nutrients can be converted into virtual reactions.
Each biomass metabolite is controlled by one reaction: **no more major constraint as in the LP formulation**.

Fixed Sets for Multiple Gap-filling

The user provides fixed sets of reactions and metabolites “at no cost or gain”.

- Set of fixed reactions to use at no cost: typically all metabolic reactions of the PGDB are used
- Sets of nutrient and secreted metabolites that can be used at no cost
- Set of metabolites that must be produced in the biomass and for which no gain is given
- Any or all of these sets might be empty
- **It is recommended to start with an empty set of fixed biomass metabolites.**

Try-Sets and Weights for Multiple Gap-filling

The user provides four try-sets and weights to control the generation of the model.

- Set of reactions to try to add at a cost: typically all metabolic reactions of MetaCyc
- Sets of nutrients, secretions and biomass metabolites to try to add to the model
- Weights, as integers for gain and cost, for the reactions, nutrients, secretions and biomass metabolites
- Typically, adding a biomass metabolite is a gain, but adding a reaction or a nutrient is a cost. We have different weights for different type of reactions (e.g., spontaneous, in the taxonomic range, etc.)

The Objective Function for Multiple Gap-Filling (1)

The objective function to maximize is

$$\sum_i w_t R_i + \sum_i w_o R_i + \sum_i w_u R_i + \sum_i w_s R_i + \sum_i w_r R_i + \sum_i w_{rm} R_i$$
$$\sum_i w_b B_i + \sum_i w_s S_i + \sum_i w_n N_i - \sum_j F_j$$

where $w_t, w_o, w_u, w_s, w_r, w_{rm}$ are weights for reactions in taxonomic range, outside taxonomic range, unknown taxonomic range, spontaneous, reversed from PGDB, and reversed from Metacyc

where w_b, w_s, w_n are weights for biomass, secretions, and nutrients

where B_i, R_i, S_i, N_i are binary variables

where F_j are the fluxes of reactions.

The Objective Function for Multiple Gap-Filling (2)

- Different weights $w_t, w_o, w_u, w_s, w_r, w_{rm}$ for reactions
- The w_t are for reactions from MetaCyc being in the taxonomic range of the PGDB whereas w_o are for reactions outside the taxonomic range and w_u for reactions of unknown taxonomic range
- The weight w_s for spontaneous reaction should be a low negative number and not zero to avoid bringing them all in the model
- The w_r are for reversed reactions from the PGDB that are not reversible
- Similarly for w_{rm} for reversed reactions from MetaCyc
- The term $-\sum_j F_j$ forces removal of the high-flux loops
- At that stage, the real fluxes of the reactions are not optimized. Solving the generated FBA model would give the optimized fluxes.

Mixed Integer Linear Programming (MILP)

- A MILP formulation is typically more difficult to solve exactly than a LP formulation due to the integer and binary variables. Essentially, the integer and binary variables require the solver to try to solve many (e.g., thousands) of LP cases.
- The solver might take forever to find the optimal solution
- We typically set a time limit to the solver, say 5 minutes
- MILP solvers vary widely in their performance and capabilities

The Weights: Costs and Gains

Typical Weights

- Adding a biomass metabolite to the model is a **gain**.
- Adding any reaction, secretion, or nutrient has a **cost**.
- That corresponds to the usual goal: generating as many biomass metabolites as possible with the minimum number of nutrients, secretions, and added reactions

Variations

But other scenarios are useful: use as many nutrients and secretions as possible

Selecting the Right Weights for Reactions

There are many different weights for the reactions: taxonomic range, reversed, and more

Balancing Out Costs and Gains (1)

- Example: 1000 for a biomass metabolite, -10 for a reaction, -2 for a nutrient, -1 for a secretion
- The solver could add as many as 100 reactions to produce one biomass metabolite
- The solver would not add 101 reactions since there would be a net lost
- Setting the gain at 10^6 for one biomass metabolite would certainly add as many reactions as possible to produce every biomass metabolite since MetaCyc will have less than 100,000 reactions for the foreseeable future

Balancing Out Costs and Gains (2)

- The ratio between the weight (cost) for a reaction and a nutrient should also be wisely selected
- Example: -10 for a reaction and -6 for a nutrient would bring in some reactions capable of replacing any two nutrients. This is typically undesirable. But you might be interested to see if this is possible.

The Reaction Weights

- The basic weight for a reaction from MetaCyc
  - **outside the taxonomic range** of the PGDB is given by `try-reactions-weight`
  - **in the taxonomic range** of the PGDB is given by `try-reactions-in-taxa-weight`
  - **of unknown taxonomic range** is given by `try-reactions-unknown-taxa-weight`
- A reversed reaction from MetaCyc is added with the *additional* weight given by `try-reactions-reverse-try-weight`. This is an additional weight to the basic weight.
- A reversed reaction from the PGDB, but reversed, is added with weight given by `try-reactions-reverse-weight`

Suggested Simple `.fba` Settings

- The entire biomass reaction is specified as a list of metabolites in the try-biomass section. No metabolites specified in the biomass fixed set.
- Similarly for nutrients and secretions: all metabolites are specified in the try-sets
- Parameter try-biomass-weight is set to a high value, say 10000. All other weights are negative with small values (-1 to -100).
- No coefficients for the metabolites (biomass, nutrients, or secretions)
- The reaction section says `metab-all`. The try-reactions section is empty.
- We have `try-add-reverse-rxns:  no`
- Execute this `.fba` file and see how many try-biomass metabolites can be produced

Settings When not all Biomass is Produced

- You will rarely get all the try biomass metabolites produced the first time
- In this case, add the keyword `metacyc-metab-all` to section `try-reactions`
- Execute
- More biomass metabolites could be produced with suggested reactions to add
- Analyze the suggested reactions to add: are they in the same taxonomic range as the organism of the PGDB, do they form a pathway, etc.?
- If the number of suggested reactions to add is overwhelming, decrease the list of metabolites to try in the biomass reaction

Other Suggested `.fba` Settings

- If the previous settings do not provide more biomass produced, add `try-add-reverse-rxns:yes` and `try-add-reverse-try-rxns:yes`
- You could also try to only try reversing the reactions of your PGDB without considering the MetaCyc reactions
- Make sure the list of metabolites to secrete is not too small: add secretions, to the try-secretion set, that you think might be missing
- Execute
- Really add, to your PGDB, some of the suggested reactions to add that you think are indeed missing
- Really change the directionality of some of the reactions as suggested by MetaFlux

Controlling Which Reactions that Can Be Added

- Most of the reactions are added from the set given by `try-reactions`
- Reversed reactions from MetaCyc might be added only if `try-add-reverse-try-rxns` is `yes`
- Reversed reactions from the PGDB might be added only if `try-add-reverse-rxns` is `yes`

Move Try Metabolites into Fixed Sets

- When it is clear that some biomass metabolites can be produced, they can be listed in the `biomass` section, that is as fixed biomass metabolites
- Similarly for secretions and nutrients, they should eventually be listed as fixed metabolites in the `.fba` file

Iterative Tuning

- The preceding suggestions will have to be repeated until a satisfactory set of biomass metabolites is produced
- It will require a careful analysis of the suggested reactions to add
- Typically, this process requires from several days to several weeks of work

FBA Model

The goal of this entire process is to get only fixed sets so that it describes a desired FBA model

- At some point it might be useful to change the weights to increase the speed of the solver

Checking a Model

- A complete FBA model will require proper coefficients for the biomass reaction
- These coefficients could come from *in vivo* experiments
- The solving of a model will verify that the biomass flux is similar to *in vivo* experiments
- The completed model can also be verified against gene knockout experiments

Suboptimal Solutions

- If the SCIP solver terminates due to the time limit, a suboptimal solution has been found.
- An optimal solution is sometimes needed to have the correct reactions to add.
- The quality of the solution is given by the "gap" percentage given in the terminal output alongside all output of the SCIP solver.
- This gap value must be interpreted based on the weights used. Typically, we expect a gap of less than 5% to call it good enough.
- In the first stages of generating a model, we limit the solver to 5 minutes. At some steps it might useful to increase it: 20 minutes, 30 minutes, one hour, or more.

How To Interpret the Gap

- It is simpler to analyze the difference between the last two values under the headings "Dualbound" and "Primalbound" of the SCIP output
- Let's assume the gap is not 0%. And the suboptimal solution suggests three reactions to add.
- Assume that Primalbound is 319900, Dualbound is 320000. The difference is 100.
- Assuming that all the costs for adding a reaction is 50, it means that it is possible that SCIP will find an optimal solution removing two suggested reactions to add
- If you doubt that the suggested reactions to add are necessary, rerun with more time for the solver