

Tutorial 3 – Apply KOs, Run MOMA and Integrate External Data

This exercise shows how to run FBA and minimization of metabolic adjustment (MOMA) simulations and how one can use GEMs as a scaffold for interpreting microarray data. A simplified model of yeast metabolism is used in this approach as an example.

Open tutorial3.m in MATLAB to begin this exercise. This script contains all the functions needed to complete this exercise.

The solutions to this exercise can be found in tutorial3_solutions.m file.

Figure 2 shows the system, this time it is a central carbon metabolism. Note that there are slight differences in naming of metabolites; particularly all metabolites have their compartment in their name.

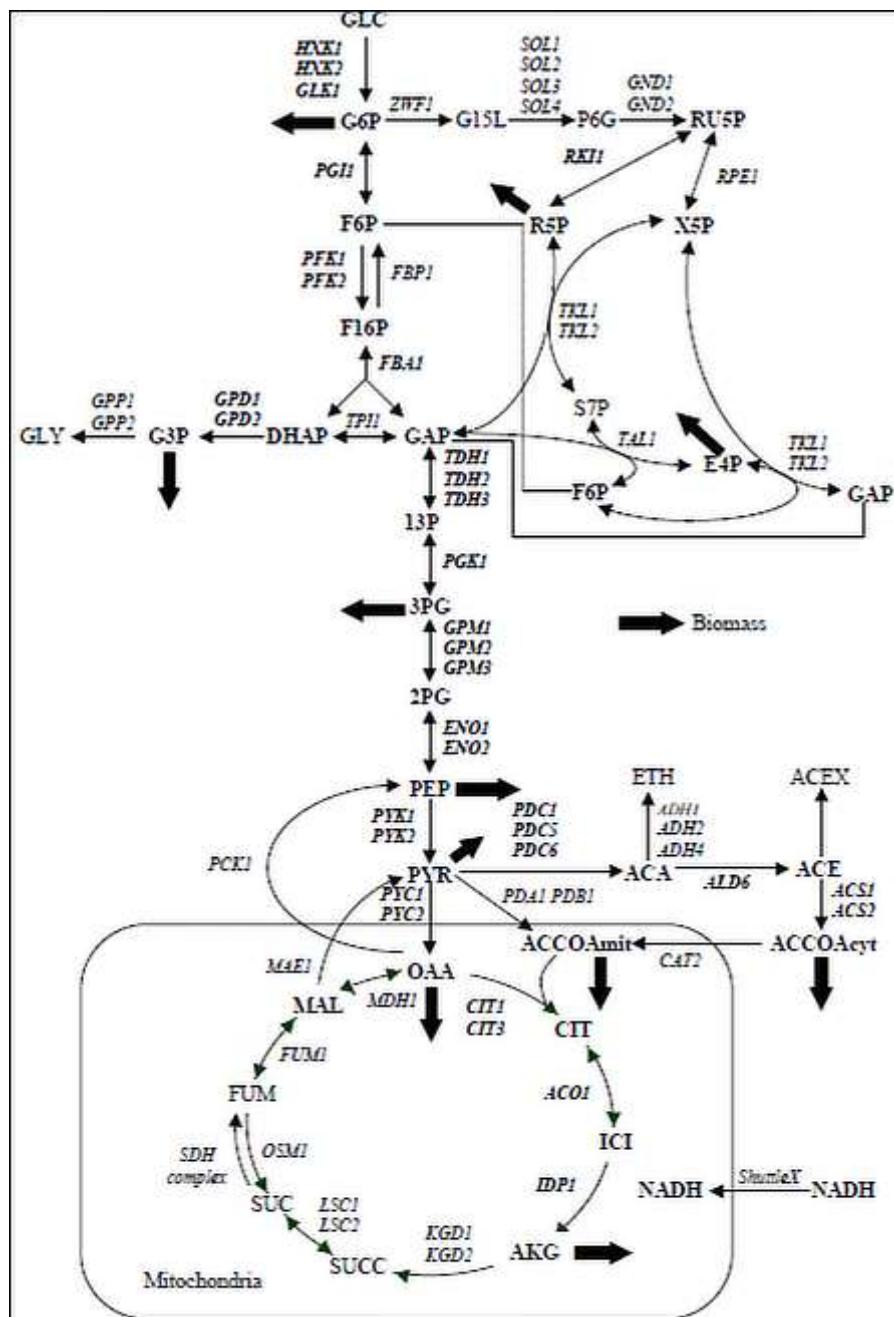


Figure 2. Central carbon metabolism in yeast (adapted from Förster, 2003).

1. Open the file smallYeast.xlsx. One will hopefully recognize the model structure from Tutorial 2, even though some more stuff is added this time. The most important difference is the composition for metabolites and gene associations for reactions. Metabolite composition information is useful for finding errors in the GEMs and knowing gene associations allows to link the GEMs to other types of data, such as microarrays. By default, the production of acetate, biomass, glycerol, CO₂, and ethanol is

allowed and the consumption of O₂ and glucose is not allowed. Try to validate the model by confirming the results from Tutorial 2 (the maximal ATP production in glycolysis). One can set the simulation parameters in the Excel file like in Tutorial 2 or try to use the setParam function instead (see tutorial3.m). Just remember that sucrose was used in the first example while glucose is used here.

```
In [ ]: setRavenSolver('gurobi')
```

```
In [ ]: % Load model
model = importExcelModel('./tutorial_data/smallYeast.xlsx');
printModelStats(model, true, true)
```

```
NOTE: DEFAULT LOWER not supplied. Uses -1000
NOTE: DEFAULT UPPER not supplied. Uses 1000
Network statistics for smallYeast: Central carbon metabolism for yeast
Genes*                      61
NOTE: DEFAULT UPPER not supplied. Uses 1000
Network statistics for smallYeast: Central carbon metabolism for yeast
Genes*                      61
    cytosol 52
    mitochondria     17

Reactions*                  53
    cytosol 45
    mitochondria     19
Unique reactions**          53

Metabolites                 52
    cytosol 35
    mitochondria     17
Unique metabolites           45
```

* Genes and reactions are counted for each compartment if any of the corresponding metabolites are in that compartment. The sum may therefore not add up to the total number.

** Unique reactions are defined as being biochemically unique (no compartmentalization)

Short model quality summary for smallYeast: Central carbon metabolism for yeast
 Reactions which could not be elementally balanced 2

```
    biomassOUT
    GROWTH
```

Reactions which are elementally unbalanced	19
acOUT	carbon, oxygen, hydrogen
co2OUT	carbon, oxygen
ethOUT	carbon, oxygen, hydrogen
glyOUT	carbon, oxygen, hydrogen
glcIN	carbon, oxygen, hydrogen
o2IN	oxygen
ethIN	carbon, oxygen, hydrogen
FBP	oxygen, hydrogen
ENO	oxygen, hydrogen
PGL	oxygen, hydrogen
GPP	oxygen, hydrogen
ALD6	oxygen, hydrogen
ACS	oxygen, hydrogen
PYC	oxygen, hydrogen
CIT	oxygen, hydrogen
FUM1	oxygen, hydrogen
NADHX	oxygen, hydrogen
FADHX	oxygen, hydrogen
ATPX	oxygen, hydrogen

```
In [ ]: % get objective
fprintf('Current objective function: %s\n\n', strjoin(model.rxns(model.c == 1)))
% show transport reactions
fprintf('Transport reactions:\n\t- %s\n', strjoin(model.rxns(getTransportRxns(model
```

```
Current objective function: biomassOUT
```

```
Transport reactions:
```

- CAT2
- CO2TRANS
- GROWTH
- ShuttleXbiomassOUT

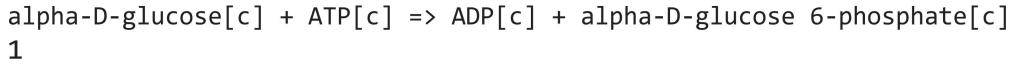
```
Transport reactions:
```

- CAT2
- CO2TRANS
- GROWTH
- ShuttleX

```
In [ ]: % validate ATP from glucose in aerobic conditions
validation_model = setParam(model, 'obj', {'ATPX'}, 1);
validation_model = setParam(validation_model, "ub", {"glcIN", "o2IN"}, [1, 1000]);
validation_model_sol = solveLP(validation_model);
printFluxes(validation_model, validation_model_sol.x, false, 10^-7, [], '%rxnID (%r
```

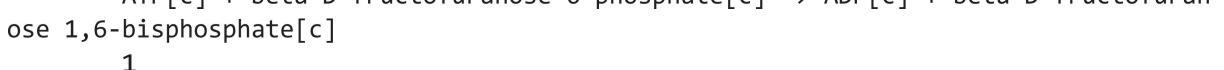
```
FLUXES:
```

```
HXK (Hexokinase):
```



```
1
```

```
PFK (Phosphofructokinase):
```



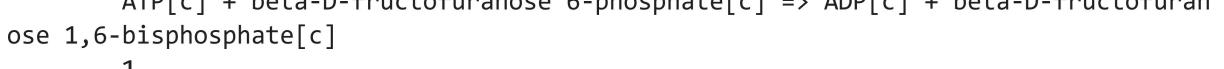
```
1
```

```
HXK (Hexokinase):
```



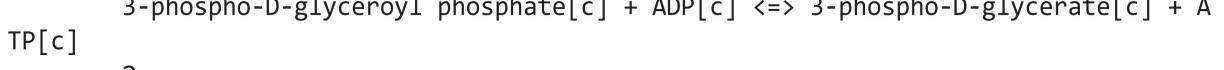
```
1
```

```
PFK (Phosphofructokinase):
```



```
1
```

```
PGK (Phosphoglycerate kinase):
```



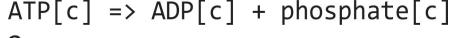
```
2
```

```
PYK (Pyruvate kinase):
```



```
2
```

```
ATPX (ATP hydrolysis):
```



```
2
```

```
In [ ]: % depure variables that will be never used
clear validation_model validation_model_sol
```

2. Calculate the maximal yields of biomass, ethanol, glycerol, and acetate on glucose and fully aerobic (biomass is in 1/h and the other fluxes are in mmol/gDW/h).

```
In [ ]: % set conditions
aerobic_conditions_model = setParam(model, "ub", {"glcIN", "o2IN"}, [1, 1000]);
reactions_to_evaluate = ["ethOUT", "acOUT", "glyOUT", "biomassOUT"];
reactions_to_names_map = dictionary(reactions_to_evaluate, ["ethanol", "acetate", "glycerol", "biomass"]);
reactions_to_units_map = dictionary(reactions_to_evaluate, ["mmol/gDW/h", "mmol/gDW/h", "mmol/gDW/h", "mmol/gDW/h"])
```

```
In [ ]: for reaction = reactions_to_evaluate
    current_model = setParam(aerobic_conditions_model, 'obj', {reaction}, 1);
    current_solution = solveLP(current_model);
    fprintf('- Maximum yield of %s is %5.4f %s.\n', ...
        reactions_to_names_map(reaction), current_solution.f*-1, reactions_to_units_map(reaction));
end
```

- Maximum yield of ethanol is 2.0000 mmol/gDW/h.
- Maximum yield of acetate is 0.6519 mmol/gDW/h.
- Maximum yield of glycerol is 0.5470 mmol/gDW/h.
- Maximum yield of biomass is 0.1222 /h.

```
In [ ]: clear aerobic_conditions_model reactions_to_evaluate reactions_to_names_map reactions_to_units_map
```

```
clear reaction current_model current_solution
```

3. Assume that one wants to predict how the flux distribution would change if the conditions are changed. The user should know by now how to import models and set parameters. Import the model twice and generate one model which has access to 1 mmol/gDW/h of glucose and unconstrained oxygen, and one which has access to 1 mmol/gDW/h of glucose and 0.5 mmol/gDW/h of O₂. Optimize for growth under both conditions. What were the outputs? Look in the tutorial and try to run drawMap for the two models. Open the resulting PDF file and see if one can understand (in broad terms) how metabolism changes between the two conditions.

```
In [ ]: % set conditions
aerobic_model = setParam(model, "ub", {"glcIN", "o2IN"}, [1, 1000]);
anaerobic_model = setParam(model, "ub", {"glcIN", "o2IN"}, [1, 0.5]);
models_to_check = [aerobic_model, anaerobic_model];
messages = ["aerobic", "anaerobic"];
stored_solution_distribution = [];
```

```
In [ ]: for index = 1:2
    current_model = models_to_check(index);
    current_solution = solveLP(current_model);
    stored_solution_distribution = [stored_solution_distribution, current_solution];
    fprintf('- Maximum yield of biomass in %s conditions is %5.4f /h\n', messages(index));
end
```

- Maximum yield of biomass in aerobic conditions is aerobic conditions is 0.1222 /h
- Maximum yield of biomass in anaerobic conditions is 0.0746 /h

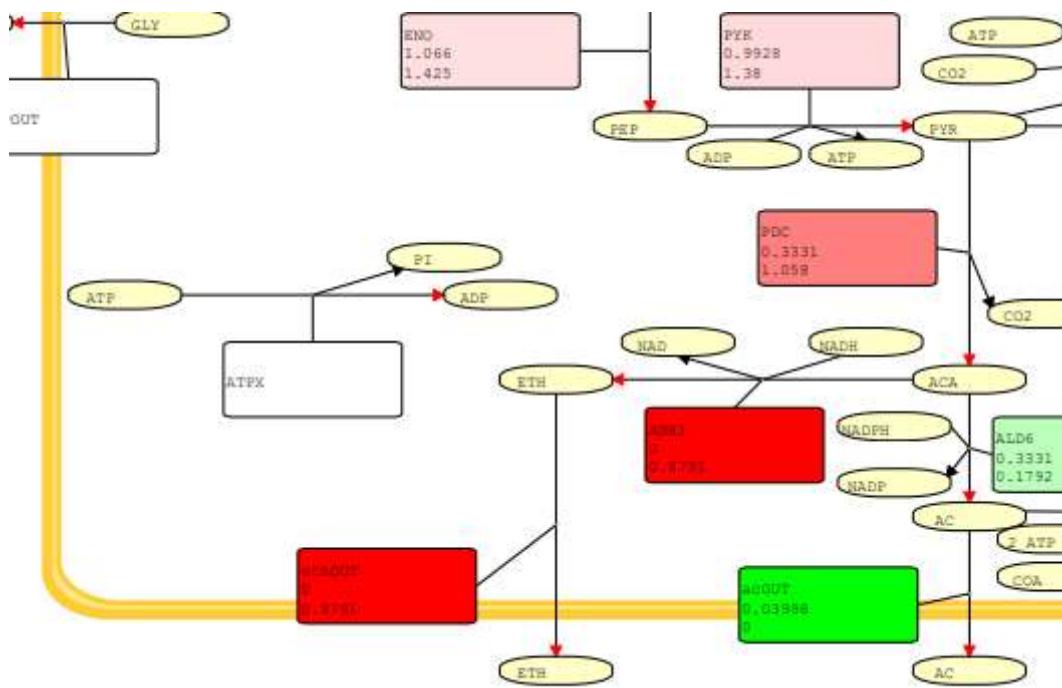
```
In [ ]: % Draw a map
% drawMap(title,pathway,modelA,conditionA,conditionB,modelB,filename,cutOff,supress
load("./tutorial_data/pathway.mat", "pathway");
drawMap('Aerobic vs. Anaerobic', pathway, ...
    aerobic_model, stored_solution_distribution(:, 1), ...
    stored_solution_distribution(:, 2), anaerobic_model, ...
    './output/tutorial_3_mapFBA.pdf', 10^-5);
```

File saved as ./output/tutorial_3_mapFBA.pdf
./output/tutorial_3_mapFBA.pdf

```
In [ ]: clear aerobic_model anaerobic_model models_to_check messages stored_solution_distribution
clear index current_model current_solution
```

- Alcohol is produced at anaerobic conditions (red).
- Acetate is produced at aerobic conditions (green).

This makes sense since the alcohol group (R-OH) is more reduced than the carboxylic group (R-COOH). In anaerobic conditions will be more favored fermentation.



4. Say that the goal is to increase glycerol production in the strain (e.g., to make sweeter wine). FBA is useful for suggesting gene deletions which can couple the set goal with the overall objective for the cell, which is to grow as fast and efficient as possible. Set the parameters to maximize for growth under anaerobic conditions and run the code for making a single gene deletion using FBA. How large was the difference in glycerol production? Can one come up with a hypothesis as to how the deletion leads to higher glycerol production? One can also use the function followChanged, which compares two flux distributions and let the user search between them (see tutorial3.m)

- `findGeneDeletions`

```
function [genes, fluxes, originalGenes, details,
grRatioMuts]=findGeneDeletions(model,testType,analysisType,refModel,oeFactor)
Deletes genes, optimizes the model, and keeps track of the resulting fluxes. This is used for
identifying gene deletion targets.
```

```
In [ ]: anaerobicModel = setParam(model, "ub", {"glcIN", "o2IN"}, [1, 0]);
[genes, fluxes, originalGenes, details, grRatioMuts] = findGeneDeletions(anaerobicM
```

- `getIndexes`

```
function indexes=getIndexes(model, objects, type, returnLogical)
Retrieves the indexes for a list of reactions or metabolite.
```

```
In [ ]: indexes = getIndexes(anaerobicModel,{'biomassOUT', 'glyOUT'},'rxns');
biomassOutIndex = indexes(1);
glycerolOutIndex = indexes(2);
```

- `find`

```
[row,col] = find(__)
```

Returns the row and column subscripts of each nonzero element in array X using any of the input arguments in previous syntaxes.

- `max`

```
[M,I] = max(__)
```

Also returns the index into the operating dimension that corresponds to the first occurrence of the maximum value of A.

```
In [ ]: % set the threshold from which the value is considered equal to zero
tolerance = 10^-7;
% "fluxes" is a matrix of size {number of reactions} x {deleted genes}
% get biomassOut fluxes for all deletions and find those with growth
columnsWithGrow = find(fluxes(biomassOutIndex, :) >= tolerance);
% get glycerolOut fluxes for the chosen deletions
glycerolFluxes = fluxes(glycerolOutIndex, columnsWithGrow);
% get the maximum possible glycerol production
maximumGlycerolProduction = max(glycerolFluxes);
% First, remember that more than one deletion can give the maximum glycerol product
% Second, the appropriate way to compare float point values is using a delta. Do not
columnsWithMaximumGrow = columnsWithGrow(abs(glycerolFluxes - maximumGlycerolProduction) < tolerance);
targetGenes = originalGenes(columnsWithMaximumGrow);
fprintf('The maximum glycerol production is: %5.4f mmol/gDW/h.\nRelated gene single
full(maximumGlycerolProduction), strjoin(targetGenes, '\n\t- '));
```

The maximum glycerol production is: 0.37010.3701 mmol/gDW/h.

Related gene single deletions:

- YLR354C
- YNL241C

```
In [ ]: % related reactions
targetReactions = anaerobicModel.rxns(contains(anaerobicModel.grRules, targetGenes));
targetReactionNames = anaerobicModel.rxnNames(contains(anaerobicModel.grRules, targetGenes));
disp(table(targetGenes, targetReactions, targetReactionNames));
```

targetGenes	targetReactions	targetReactionNames
{'YLR354C'}	{'ZWF'}	{'Glucose-6-phosphate 1-dehydrogenase'}
{'YLR354C'} {'YNL241C'}	{'ZWF'} {'TAL1'}	{'Glucose-6-phosphate 1-dehydrogenase'} {'Transaldolase'}

```
In [ ]: wildTypeFluxes = solveLP(anaerobicModel).x;
electronCarriers = {'NADPH' 'NADH' 'NAD(+)''NADP(+)'};
% Notice that it is the transpose of targetReactions. MATLAB Loops go over the columns
for reaction = targetReactions'
    fprintf('Current reaction: %s\n', string(reaction));
    % set conditions
    mutatedModel = setParam(anaerobicModel,'eq', reaction, 0);
    mutatedFluxes = solveLP(mutatedModel).x;
    % export file
    title = reaction + " deletion vs. wildtype" ;
    fileName = strjoin(['./output/tutorial_3_map', reaction, '.pdf'], '');
    drawMap(title , pathway, ...
    mutatedModel, mutatedFluxes, ...
    wildTypeFluxes, anaerobicModel, ...
    fileName, 10^-5);
    % print some relevant changes related with electrons carriers (reduction power)
    followChanged(anaerobicModel, mutatedFluxes, wildTypeFluxes, ...
    20, 10^-2, 0, electronCarriers);
    fprintf('-----\n') ;
end
```

Current reaction: ZWF
File saved as ./output/tutorial_3_mapZWF.pdf
These reactions have flux values that differ by more than 20 percent, absolute values above 0.01, and a total difference above ZWF
File saved as ./output/tutorial_3_mapZWF.pdf
These reactions have flux values that differ by more than 20 percent, absolute values above 0.01, and a total difference above 0 (4 reactions)

Only prints reactions involving one or more of the following metabolites:
NADPH NADH NAD(+) NADP(+)

ZWF: alpha-D-glucose 6-phosphate[c] + NADP(+) [c] => 6-O-phosphono-D-glucono-1,5-lactone[c] + NADPH[c]
Glucose-6-phosphate 1-dehydrogenase
Flux: 0 Reference flux: 0.10432 Difference: -0.10432

GND: 6-phospho-D-gluconate[c] + NADP(+) [c] => CO2[c] + D-ribulose 5-phosphate[c] + NADPH[c]
6-phosphogluconate dehydrogenase, decarboxylating
Flux: 0 Reference flux: 0.10432 Difference: -0.10432

DAR: glycerone phosphate[c] + NADH[c] => glycerol monophosphate[c] + NAD(+) [c]
glycerol-3-phosphate dehydrogenase
Flux: 0.37288 Reference flux: 0.23556 Difference: 0.13732

ALD6: acetaldehyde[c] + NADP(+) [c] => acetate[c] + NADPH[c]
Aldehyde dehydrogenase
Flux: 0.25424 Reference flux: 0.094223 Difference: 0.16001

Current reaction: TAL1
File saved as ./output/tutorial_3_mapTAL1.pdf
These reactions have flux values that differ by more than 20 percent, absolute values above 0.01, and a total difference above 0 (4 reactions)

Only prints reactions involving one or more of the following metabolites:
NADPH NADH NAD(+) NADP(+)

ZWF: alpha-D-glucose 6-phosphate[c] + NADP(+) [c] => 6-O-phosphono-D-glucono-1,5-lactone[c] + NADPH[c]
Glucose-6-phosphate 1-dehydrogenase
Flux: 0 Reference flux: 0.10432 Difference: -0.10432

GND: 6-phospho-D-gluconate[c] + NADP(+) [c] => CO2[c] + D-ribulose 5-phosphate[c] + NADPH[c]
6-phosphogluconate dehydrogenase, decarboxylating
Flux: 0 Reference flux: 0.10432 Difference: -0.10432

DAR: glycerone phosphate[c] + NADH[c] => glycerol monophosphate[c] + NAD(+) [c]
glycerol-3-phosphate dehydrogenase
Flux: 0.37288 Reference flux: 0.23556 Difference: 0.13732

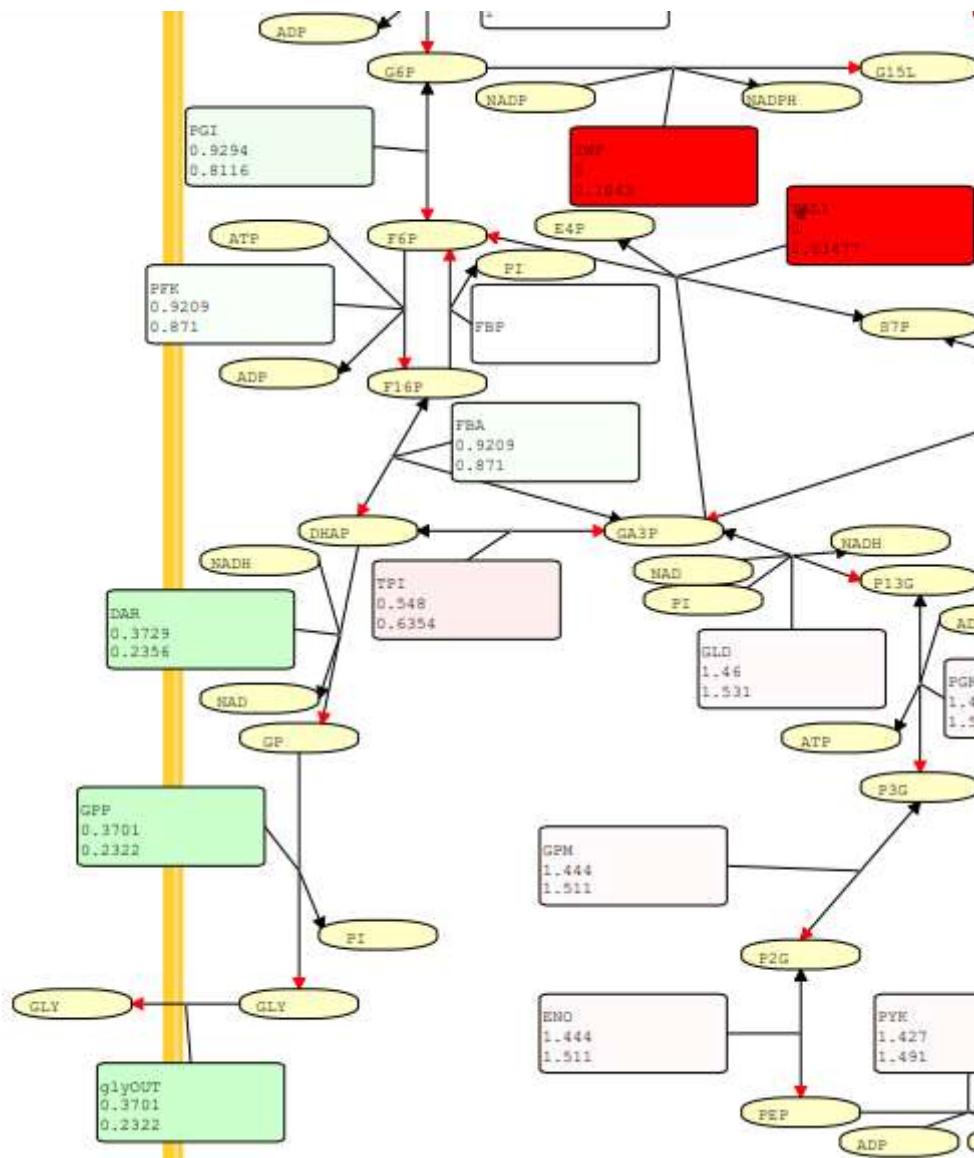
ALD6: acetaldehyde[c] + NADP(+) [c] => acetate[c] + NADPH[c]
Aldehyde dehydrogenase
Flux: 0.25424 Reference flux: 0.094223 Difference: 0.16001

- How large was the difference in glycerol production?

```
In [ ]: glycerolProductionIndex = getIndexes(anaerobicModel, 'glyOUT', 'rxns');
fprintf('Glycerol production:\n');
disp(table(mutatedFluxes(glycerolProductionIndex), ...
          wildTypeFluxes(glycerolProductionIndex), ...
          (mutatedFluxes(glycerolProductionIndex)/wildTypeFluxes(glycerolProductionIndex)) * 100));
VariableNames', {'mutant', 'wild-type', 'relative difference mutant vs. WT (%)'})
```

Glycerol production:		
mutant	wild-type	relative difference mutant vs. WT (%)
0.37006	0.23219	59.375
mutant	wild-type	relative difference mutant vs. WT (%)
0.37006	0.23219	59.375

- Can one come up with a hypothesis as to how the deletion leads to higher glycerol production?



Knock out of glucose-6-phosphate 1-dehydrogenase ([zwf](#)) increase the Glucose-6P available for conversion to Fructose-6P and Glycerone-P. [KEGG map](#).

```
In [ ]: clear anaerobicModel fileName maximumGlycerolProduction fluxes reaction
clear wildTypeFluxes genes modelMutatedZWF referenceFluxes biomassOutIndex
clear glycerolFluxes targetGenes columnsWithGrow glyceralOutIndex mutatedFluxes
clear targetReactionNames columnsWithMaximumGrow grRatioMuts mutatedModel
clear details indexes originalGenes title targetReactions glyceralProductionIndex
```

5. FBA assumes that cells have the same objective even after a perturbation; to grow as efficiently as possible. This is probably true after long evolution times, but shortly after, say, a gene deletion the cell might have other objectives. MOMA is based on that the perturbed cell would like to change its metabolism as little as possible. MOMA is normally used when one has the experimental data for the wild type of strain and would like to guess how the phenotype would change after a perturbation. These are some batch data for growth on glucose:

- acOUT (Production of acetate): 0

- biomassOUT (Production of biomass): 0.67706


```

- co2OUT (Production of CO2): 22.4122<br>
- ethOUT (Production of ethanol): 19.0946<br>
- glyOUT (Production of glycerol): 1.4717<br>
- glcIN (Uptake of glucose): 15<br>
- o2IN (Uptake of O2): 1.6<br>
- ethIN (Uptake of ethanol): 0<br>

```

Load one model with these constraints and one without any constraints on the exchange reactions but with the ZWF reaction turned off. Run the qMOMA function and plot the flux distributions. Is the glycerol production larger?

```
In [ ]: %%%%%% set up simulation %%%%%%
experimentalFluxNames = {"acOUT", "biomassOUT", "co2OUT", "ethOUT", "glyOUT", "glcI"
experimentalFluxValues = [0, 0.67706, 22.4122, 19.0946, 1.4717, 15, 1.6, 0];
```

```
In [ ]: % With this data the model is infeasible, so we need to relax a bit the constraints
wildTypeModel = setParam(model, "eq", experimentalFluxNames, experimentalFluxValues
fprintf('wildTypeModel optimization message:\n\t%', solveLP(wildTypeModel).msg);
```

wildTypeModel optimization message:
The problem is infeasibleThe problem is infeasible

```
In [ ]: wildTypeModel = setParam(model, "lb", experimentalFluxNames, experimentalFluxValues
wildTypeModel = setParam(wildTypeModel, "ub", experimentalFluxNames, experimentalFluxValues
fprintf('wildTypeModel optimization message:\n\t%', solveLP(wildTypeModel).msg);
```

wildTypeModel optimization message:
Optimal solution foundOptimal solution found

```
In [ ]: % now set up the mutant model
mutantModel = setParam(model, 'eq', {'ZWF'}, 0);
mutantModel = setParam(mutantModel, "lb", experimentalFluxNames, 0);
mutantModel = setParam(mutantModel, "ub", experimentalFluxNames, 1000);
```

qMOMA

```
function [fluxA,fluxB, flag]=qMOMA(modelA,modelB,fluxMinWeight)
```

Uses quadratic programming to minimize the sum((fluxAi - fluxBi)^2)

❗ Warning: Seems like it is necessary a MATLAB add-on:

'quadprog' requires Optimization Toolbox.

Error in qMOMA (line 74)

```
x=quadprog(H,zeros(size(H,1),1),[],[],fullS,fullB,fullLB,fullUB);
```

I am not sure why this add-on is needed if the environment already have a working solver (Gurobi) 😞

```
In [ ]: % IMPORTANT! remember that modelB is the reference case
[mutantFluxes, wildTypeFluxes, flag] = qMOMA(mutantModel, wildTypeModel);
```

Minimum found that satisfies the constraints.

Optimization completed because the objective function is non-decreasing in feasible directions, to within the value of the optimality tolerance, and constraints are satisfied to within the value of the constraint tolerance.

```
In [ ]: followChanged(wildTypeModel, mutantFluxes, wildTypeFluxes, 10, 10^-2, 10^-8, elect
```

These reactions have flux values that differ by more than 1010 percent, absolute values above 0.01, and a total difference above 1e-08 (10 reactions)

Only prints reactions involving one or more of the following metabolites:
NADPH NADH NAD(+) NADP(+)

ZWF: alpha-D-glucose 6-phosphate[c] + NADP(+) [c] => 6-O-phosphono-D-glucono-1,5-lactone[c] + NADPH[c]

Glucose-6-phosphate 1-dehydrogenase

Flux: 0 Reference flux: 2.0686 Difference: -2.0686

GND: 6-phospho-D-gluconate[c] + NADP(+) [c] => CO2[c] + D-ribulose 5-phosphate[c] + NADPH[c]

6-phosphogluconate dehydrogenase, decarboxylating

Flux: 0 Reference flux: 2.0686 Difference: -2.0686

DAR: glycerone phosphate[c] + NADH[c] => glycerol monophosphate[c] + NAD(+) [c]

glycerol-3-phosphate dehydrogenase

Flux: 3.0085 Reference flux: 1.5534 Difference: 1.455

ADH1: acetaldehyde[c] + NADH[c] <=> ethanol[c] + NAD(+) [c]

Alcohol dehydrogenase

Flux: 17.209 Reference flux: 19.1521 Difference: -1.9431

ALD6: acetaldehyde[c] + NADP(+) [c] => acetate[c] + NADPH[c]

Aldehyde dehydrogenase

Flux: 4.1697 Reference flux: 1.8955 Difference: 2.2742

IDP1: isocitrate[m] + NADP(+) [m] => 2-oxoglutarate[m] + CO2[m] + NADPH[m]

Isocitrate dehydrogenase

Flux: 0.04633 Reference flux: 0.085713 Difference: -0.039383

KGD1KGD2: 2-oxoglutarate[m] + coenzyme A[m] + NAD(+) [m] => CO2[m] + NADH[m] + succinyl-CoA[m]

Alpha-ketoglutarate dehydrogenase

Flux: 1.8215e-17 Reference flux: 0.018684 Difference: -0.018684

MDH1: (S)-malate[m] + NAD(+) [m] <=> NADH[m] + oxaloacetate[m]

Malate dehydrogenase

Flux: -0.97293 Reference flux: -1.3702 Difference: 0.39731

MAE1: (S)-malate[m] + NADP(+) [m] => CO2[c] + NADPH[m] + pyruvate[c]

NAD-dependent malic enzyme

Flux: 0.97293 Reference flux: 1.3889 Difference: -0.41599

GROWTH: 2-oxoglutarate[m] + 6 3-phospho-D-glycerate[c] + 24 acetyl-CoA[c] + 3 acetyl-CoA[m] + 25 alpha-D-glucose 6-phosphate[c] + 254 ATP[c] + 3 D-erythrose 4-phosphate[c] + 3 D-ribose 5-phosphate[c] + glycerol monophosphate[c] + 16 NAD(+) [c] + 6 NAD(+) [m] + 90 NADPH[c] + 22 NADPH[m] + 10 oxaloacetate[m] + 6 phosphoenolpyruvate[c] + 18 pyruvate[c] => 254 ADP[c] + 10 biomass[c] + 24 coenzyme A[c] + 3 coenzyme A[m] + 16 NADH[c] + 6 NADH[m] + 90 NADP(+) [c] + 22 NADP(+) [m] + 298 phosphate[c]

Growth

Flux: 0.04633 Reference flux: 0.067029 Difference: -0.020699

```
In [ ]: drawMap('ZWF deletion vs wild type', pathway, ...
    wildTypeModel, mutantFluxes, ...
    wildTypeFluxes, mutantModel, './output/tutorial_3_mapMOMA.pdf', 10^-5);
```

File saved as ./output/tutorial_3_mapMOMA.pdf
./output/tutorial_3_mapMOMA.pdf

- Is the glycerol production larger?

```
In [ ]: glycerolProductionIndex = getIndexes(wildTypeModel, 'glyOUT', 'rxns');
fprintf('Glycerol production:\n')
disp(table(mutantFluxes(glycerolProductionIndex), ...
    wildTypeFluxes(glycerolProductionIndex), ...
    mutantFluxes(glycerolProductionIndex)/wildTypeFluxes(glycerolProductionIndex),
    'VariableNames', {'mutant', 'wild-type', 'mutant/wildType'}))
```

Glycerol production:		
mutant	wild-type	mutant/wildType
mutant	wild-type	mutant/wildType
2.9621	1.4864	1.9928

```
In [ ]: clear flag experimentalFluxNames experimentalFluxValues glycerolProductionIndex
clear wildTypeModel mutantModel wildTypeFluxes mutantFluxes electronCarriers
```

6. GEMs can also be used to aid in interpretation of other types of data. The last lines load microarray data from batch growth on ethanol compared to glucose and perform a Reporter metabolites test. These are metabolites around which significant transcriptional changes occur (for such a small model it is not super interesting though, because there are changes almost everywhere). The very last lines plot all reactions involving any of those metabolites. This can be particularly useful when working with large models.

- [readtable](#)

```
T = readtable(filename,opts)
```

Creates a table using the import options opts. [delimitedTextImportOptions](#)

```
In [ ]: filename = './tutorial_data/expression.txt';
opts = delimitedTextImportOptions('VariableNames', {'openReadingFrame', 'pValue'},
                                  'VariableTypes', {'char', 'double'}, ...
                                  'Delimiter', '\t', ...
                                  'Whitespace', '\b');
disp(preview(filename, opts));
```

openReadingFrame	pValue
{'RTM1'}	0.52257
{'RTM1'}	0.52257
{'SUC4'}	1
{'YHR021W'}	0.36936
{'YOR008C'}	0.95133
{'YJL086C'}	0.52477
{'YOR387C'}	0.90097
{'YEL076W-C'}	0.19623
{'YGL158W'}	0.33086

```
In [ ]: reporterTable = readtable(filename, opts)
```

```
Out[ ]: reporterTable = 6383x2 table  
openReadingFrame pValue
```

openReadingFrame	pValue
{'RTM1' }	0.52257
{'SUC4' }	1
{'YHR021W' }	0.36936
{'YOR008C' }	0.95133
{'YJL086C' }	0.52477
{'YOR387C' }	0.90097
{'YEL076W-C'}	0.19623
{'YGL158W' }	0.33086
{'YOR199W' }	0.77312
{'YGL222C' }	0.19623
{'YDL163W' }	0.61951
{'YML084W' }	0.21644
{'YER044C' }	0.73427
{'YCR101C' }	0.10562
{'YKL036C' }	0.45677
{'YHR131W' }	0.6355

- reporterMetabolites

function

```
repMets=reporterMetabolites(model,genes,genePValues,printResults,outputFile,g  
The Reporter Metabolites algorithm for identifying metabolites around which transcriptional  
changes occur
```

! Warning: Seems like it is necessary another MATLAB add-on:

```
'norminv' requires Statistics and Machine Learning Toolbox.
```

```
Error in reporterMetabolites (line 78)  
geneZScores=norminv(genePValues)*-1;
```

It would be very helpful if those dependencies are more explicit in the documentation. 

```
In [ ]: repMets= reporterMetabolites(model,reporterTable.openReadingFrame, reporterTable.pV
```

sortrows

```
tblB = sortrows(tblA,vars)
```

sorts a table by the elements in the variables specified by vars. For example, sortrows(tblA, {'Var1','Var2'}) first sorts the rows of tblA based on the elements in Var1, then by the elements in Var2.

- If tblA is a table and it has row names, then vars can include the row names.
- If tblA is a timetable, then vars can include the row times.

```
In [ ]: % make a table from the reporter metabolites, next filter them by their p-value and  
sortedTable = sortrows(table(repMets.mets, repMets.metPValues, 'variableNames', {'m  
sortedTableTop10 = sortedTable(1:10,:);  
fprintf('Top 10 reporter metabolites (by p-value):\n\n');  
disp(sortedTableTop10);
```

Top 10 reporter metabolites (by p-value):

metaboliteID	pValue
{'F6P_c'}	0.044932
{'E4P_c'}	0.067485
{'S7P_c'}	0.067485
metaboliteID	pValue
{'F6P_c'}	0.044932
{'E4P_c'}	0.067485
{'S7P_c'}	0.067485
{'P6G_c'}	0.068213
{'ACCOA_c'}	0.089614
{'COA_c'}	0.089614
{'CI_m'}	0.10217
{'RU5P_c'}	0.11807
{'GLC_c'}	0.13644
{'NADP_c'}	0.14617

```
In [ ]: selectedMetsIndexes = getIndexes(model, sortedTableTop10.metaboliteID, 'mets');
% in which reactions those metabolites participate
[~, selectedReactionsIndexes] = find(model.S(selectedMetsIndexes, :));
```

trimPathway

```
function pathway=trimPathway(pathway, rxnsToKeep, deleteUnconnectedMets)
Delete reactions from a pathway object
```

```
In [ ]: updatedPathway= trimPathway(pathway, model.rxnss(selectedReactionsIndexes), true);
% notMapped=drawMap(title,pathway,modelA,conditionA,conditionB,filename,cuto
drawMap('Reactions involving the top 10 Reporter Metabolites', ...
    updatedPathway, model, ones(numel(model.rxnss),1), zeros(numel(model.rxnss),1), [
    './output/tutorial_3_mapRM.pdf', tolerance);
```

File saved as ./output/tutorial_3_mapRM.pdf
./output/tutorial_3_mapRM.pdf

```
In [ ]: clear
```