Analyzing metabolic networks with the cobra toolbox

Katja Tummler & Jannis Uhlendorf

February 22, 2017

1 Getting Started[1]

The cobra-toolbox for Matlab is an open-source software provided from the opencobra community.
It is also available for python as cobrapy.
The development is currently hosted on github `github.com/opencobra/cobratoolbox`
(here we will use the the latest stable release).

Step 1: Initializing the COBRA Toolbox in MatLab

1. Copy all the course files to a new directory, best name it with your name.
2. Open MatLab and go to the new directory (`cd`),
   run `initCobraToolbox` to load and test the toolbox.
3. We use the open source linear programming solver `glpk` (`www.gnu.org/software/glpk/`).
   To set it in the cobra environment use `changeCobraSolver('glpk')`.

Step 2: Loading the Model

In the following we want to test general functionalities of the constraint based modeling using the example of a reduced network of the central metabolism of *E. coli* (Thiele et al. 2010, Figure 1).

1. You can load the model in SBML format using the function
   `M = readCbModel('EColiCore_SBML_export.xml');`
2. The model is now stored in the MatLab structure `M`. Have a look at the different fields of the structure using Matlabs built in variable viewer (double click on variable):
   Do you recognize the fields corresponding to the SBML entities?
   How are genes linked to the reactions in the model?
   What are the fields `M.c`, `M.lb` and `M.ub`?

[1] The tutorial is based on the COBRA introcudction provided in the supplementary material of:
What is flux balance analysis? Orth JD, Thiele I, Palsson BO; Nat Biotechnol. 2010 28(3): 245-248

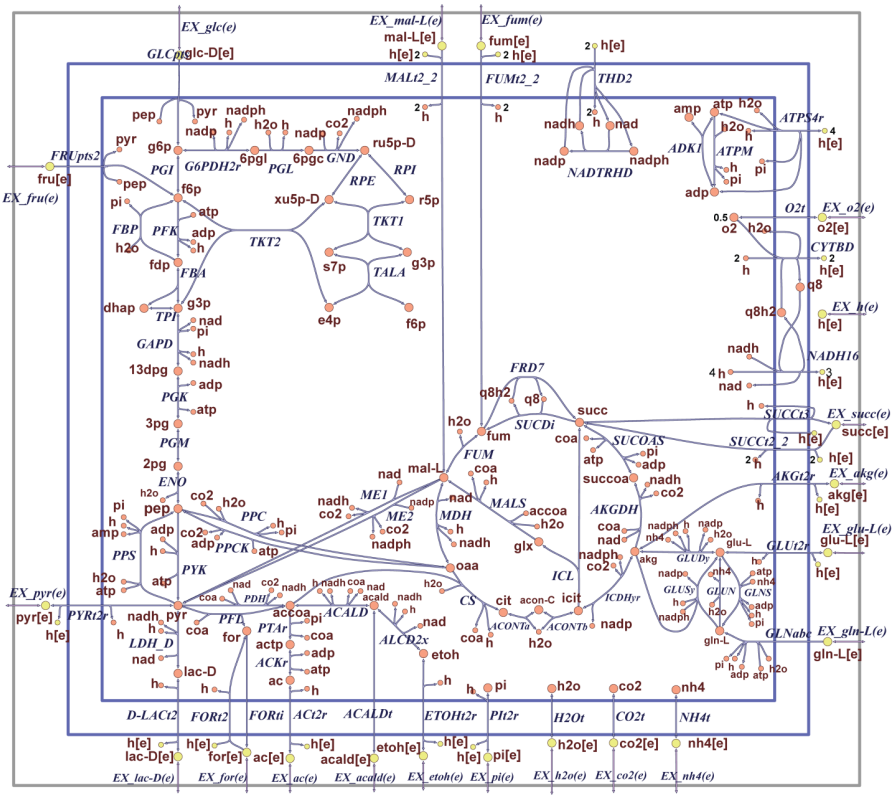

Figure 1: Schematic representation of the core model.

2 Structural Model Analysis

Step 1: Visualize the Stoichiometric Matrix

Have a look at the stoichiometric matrix of the system using the function `spy(M.S)`.

Step 2: Calculate the Connectivity Distribution

The connectivity of a node (i.e a metabolite in our model) is defined as the number of reactions in which this node is included. Biological networks often show a typical **power-law** connectivity distribution, with few highly connected nodes (so-called hubs) and a large number of weakly connected nodes.
To gain insight into the connectivity of our network we have to

1. Find all entries in the stoichiometric matrix, representing substrates and products for each occurring reaction with a stoichiometric coefficient different from 0:
   `Sbin=(M.S~=0)`
   The binary matrix `Sbin` contains a `true` for all non-zero entries of the stoichiometric matrix `S` and `false` otherwise.

2. Sum up all occurring reactions for each metabolite (in the 2nd dimension (columns) of `S`):
   `metConnectivity= sum(Sbin,2)`

3. Plot the connectivity of each metabolite in descending order:
   `plot(sort(metConnectivity))`

3 Flux Balance Analysis

Step 1: Set the Objective Function

We will now optimize the metabolic flux through the network for maximum growth. The FBA will use the maximization of growth as objective. Hence, we need a defined reaction that accounts for the formation of biomass with the correct stoichiometry from the precursor-metabolites.

1. Find the predefined biomass reaction (`'Biomass_Ecoli_core_N(w/GAM)_Nmet2'`) in the model (`M.rxns`) and print the corresponding reaction formula using the function `printRxnFormula(M, M.rxn(ix))`.

2. Find the metabolites from the above reaction and list them next to their stoichiometric factors in the biomass producing reaction:

```
biomassComponents=find(M.S(:,M.c~=0)); % find corresponding metabolites
stoichiometry = M.S(biomassComponents,M.c~=0); % get their stoichiometry
table(M.metNames(biomassComponents),stoichiometry, ...
      'VariableNames',{'Metabolite', 'Stoichiometry'})
```

3. Now we can set the objective function to the biomass reaction using the COBRA function `M = changeObjective(M,'Biomass_Ecoli_core_N(w/GAM)_Nmet2')`. The field `M.c` should now contain only 0's, except from the position of the biomass reaction, where a 1 should be entered.

Step 2: Define the Medium

Next, we have to set the boundary fluxes of the model, i.e. the substrate uptake and excretion rates. By convention, exchange reactions are written as export reactions (e.g. `'glc[e] <==>'`), so import of a metabolite is a *negative* flux.
Within COBRA, boundaries of reactions can be changed using the `changeRxnBounds` function, specifying lower (`'l'`), upper (`'u'`) or both boundaries (`'b'`).

1. We start with growth on glucose and set the glucose uptake rate to $18.5 \frac{mmol}{gDW \cdot h}$:
   `M = changeRxnBounds(M,'EX_glc(e)',-18.5,'l');`.

2. If we want an input flux to be non restrictive or a substrate to be available in excess, we can un-limit its uptake rate to a high value such as 1000. For an aerated *E. coli* culture we can apply this to the oxygen uptake rate:
   `M = changeRxnBounds(M,'EX_o2(e)',-1000,'l');`

Step 3: Run the FBA

1. Run the FBA with the aim of maximizing the specified objective function:
   `FBAsolution = optimizeCbModel(M,'max');`

2. Have a look of the results of the FBA:
   - The optimum growth rate in `FBASolution.f`
   - One possible optimum flux distribution in `FBASolution.x`

4 Altering growth conditions - Phenotypic phase plane analysis

Step 1: Analyze oxygen limiting conditions

Re-run the FBA for anaerobic conditions (i.e. the oxygen uptake rate is 0) and compare the resulting flux distributions. How large is the growth rate, how is the flux redistributed, which fluxes have changed?

Step 2: Systematically test how growth depends on glucose and oxygen availability

We now want to predict the growth rates of the bacterial cultures in dependency of the available nutrients. As started above we will focus on the influence of glucose and oxygen:

1. **Glucose dependency:** Fix the value for the oxygen uptake (`EX_o2(e)`) to $-17\ \frac{mmol}{gDW \cdot h}$. In a `for` loop, run several FBAs with changing values for the glucose uptake rate `EX_glc(e)` between 0 and $-25\ \frac{mmol}{gDW \cdot h}$ and plot the resulting optimum growth rates.
2. **Oxygen dependency:** Redo the same analysis for a fixed glucose uptake rate `EX_glc(e)` $= -5\ \frac{mmol}{gDW \cdot h}$ and changing oxygen uptake rates (between 0 and $-25\ \frac{mmol}{gDW \cdot h}$).
3. **Interdependency:** Combine the first two analyses and systematically test combinations of the oxygen and glucose uptake rate within the given bounds. You can visualize the result using the Matlab function `surf`.

5 Flux Variability Analysis

Flux Variability Analysis (FVA) can help to understand the flexibility and robustness in the metabolic system. First, think about how you would implement an FVA. Which steps are required?
For the analysis of our system we will use the cobratoolbox function for FVA, following the following steps:

1. Reset the medium to the initial state:
   Glucose uptake is restricted to $-18.5\ \frac{mmol}{gDW \cdot h}$ and oxygen is available in excess.
2. Run the FVA using the COBRA function
   `[minFluxAll,maxFluxAll] = fluxVariability(M)`.
3. Which reactions have a high variability? Find their corresponding genes (have a look at `M.rxnGeneMat`) and look up their function (e.g. in the KEGG database). What could be an explanation for the variability?

6 Gene Essentiality

Step 1: Double gene knock out

We can use FBA to simulate the effect of gene knock outs on the network flux.

1. Calculate growth rate changes for double knock-outs of all pairs of genes in the model. First think about how you would implement the task. If you want, you can try an own implementation or use the cobratoolbox function `[grRatio,grRateKO,grRateWT] = doubleGeneDeletion(M)`.
2. Visualize the relative changes in the growth rate for each double knock-out (`grRatio`) using the MatLab function `imagesc`.
3. Find a pair of genes that only causes a phenotype when knocked-out simultaneously. With the help of "the internet" try to understand why this happens.

Step 2 (advanced): Visualize impact of single gene knock-outs

For a quicker overview you can map FBA results to a graphic representation of a network. We can try this here for the impact of single gene knock-outs on the growth rate (as above for double knock outs). For that we have to extract a list containing gene identifiers and the growth rate ratio of the corresponding knock out. The numerical values then have to be translated to colors to be able to upload them for example on the KEGG database.

1. First run the gene deletion analysis using the cobratoolbox function `grRatio = singleGeneDeletion(M)`.
2. Convert the relative growth rates ($\in (0, 1)$) of the knock-outs strains to color values by interpolating them in the RGB space between red `[1 0 0]` and blue `[0 0 1]`: `rgb = interp1([1, 0],[1 0 0; 0 0 1],grRatio)`
3. Mapping on KEGG pathways requires HEX-encoded colors. Use the provides function `rgb2hex` to convert the RGB color values to the hex color space and save them in a text file:

```
t = table(M.genes, rgb2hex(rgb));
writetable(t,'SingleGeneKOResults.txt', 'delimiter', '\t')
```

4. Now that you have the data in the right format you can pick a pathway map on the KEGG homepage (KEGG PATHWAY database, for example `http://www.kegg.jp/kegg-bin/show_pathway?map=map01200`).
5. Select the reference *E. coli* strain K-12 MG1655 as organism, notice that now genes found in *E. coli* are highlighted in green.
6. Now use the option "User data mapping" to visualize your data: Paste the exported table in the pop-up window and click "pathway mapping". According to the strength of their knock-out effect, genes will now be highlighted in red to blue on the map.

7 A genome scale model (advanced)

With this bit of practice we can now avenge further to the analysis of genome scale networks. As an example we can look at some aspects of the *S. cerevisiae* model Yeast 7.

1. Load the preprocessed and annotated model file: `load('Y7_filled.mat')`.

2. Have a look at the model structure (e.g. annotations for reactions and metabolites), visualize as before the stoichiometric matrix and the connectivity distribution.
   How do they compare to the small core model?
   How is the biomass reaction defined?
   Which substances are to be found in the simulated medium?

3. Run an FBA with to maximize growth, try to map the resulting fluxes to KEGG, as above (you can use the provided KEGG identifiers of the reactions in the model). Note, that your fluxes do not lie between 0 and 1, as was the case for the relative growth rates. You can also try to use the KEGG MAPPER to explore several pathway maps (search against `map`).
   `http://www.kegg.jp/kegg/tool/map_pathway2.html`

4. Implement alternative objective function, as discussed in the lecture:
   - parsimonious FBA: Minimize the total flux and hence the enzyme cost
   - maximize the ATP yield of the cell

   Compare growth rate and fluxes between the different approaches.
   What would be the trivial solution of parsimonious FBA? How can you prevent it?