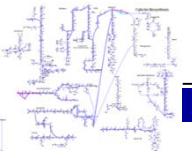


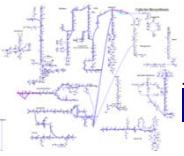
# The Cobra Toolbox



# Learning Objectives

Each student should be able to:

- Explain the purpose of the *Cobra Toolbox*,
- Demonstrate basic operation of the *Cobra Toolbox*,
- Explain the *BIGG Database*,
- Explain the capabilities of *Paint4Net*.

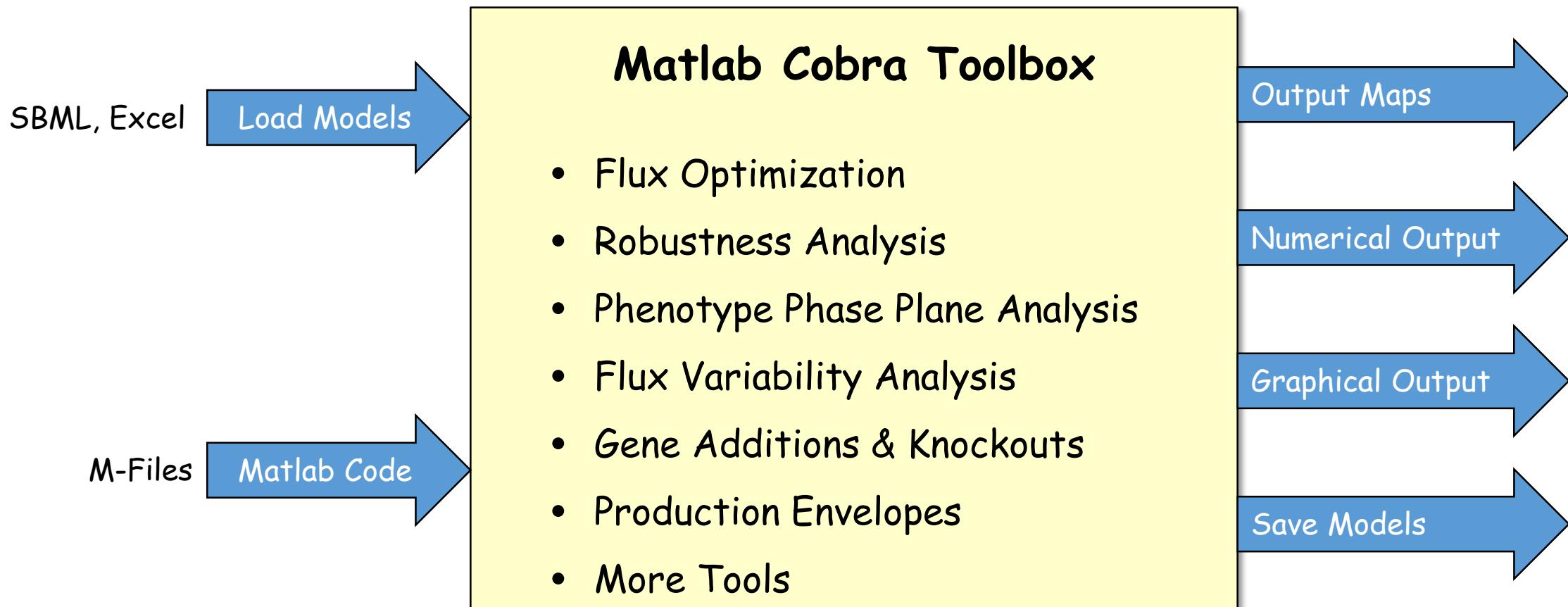


## Cobra Toolbox

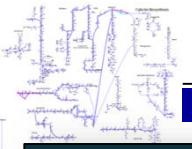
- Cobra Toolbox Overview
- Cobra Toolbox Fundamentals
- Cobra Examples
- BIGG Database
- Paint4Net



# Cobra Toolbox Overview



Schellenberger J, Que R, Fleming RMT, Thiele I, Orth JD, Feist AM, Zielinski DC, Bordbar A, Lewis NE, Rahmanian S, Kang J, Hyduke DR, Palsson BØ. 2011 Quantitative prediction of cellular metabolism with constraint-based models: the COBRA Toolbox v2.0. *Nature Protocols* 6:1290-1307.



## COBRA Toolbox

Dr Fleming's letter   Download   Document   Developer   Collaborator

# COBRA Toolbox

## COOnstraints Based Reconstruction and Analysis Toolbox

Version 2.X



<http://opencobra.github.io/cobratoolbox/#top>

# Cobra Toolbox Website

<http://opencobra.github.io/cobratoolbox/>



## Developer



Dr. Ronan  
Fleming  
Lead Developer



Prof. Dr.  
Ines Thiele  
Developer



Prof. Dr.  
Nathan  
Lewis  
Developer



Dr. Hulda  
Haraldsdottir  
Developer



Dr. Longfei  
Mao  
Developer



# Version of Cobra Toolbox Used in the Course

<http://opencobra.github.io/cobratoolbox/>

*A letter from the lead developer*

Dear All,

As of Jan 30, 2015, I am now the Lead Developer for the COBRA Toolbox. I and others have already begun bringing the COBRA Toolbox up to date and the latest version is and will henceforth be available here: (<https://github.com/opencobra/cobratoolbox/>). Older, but more stable versions are available here (<https://github.com/opencobra/cobratoolbox/archive/master.zip>)

Everyone is encouraged to work together to improve the latest version of the COBRA Toolbox by cloning the master version available here:  
<https://github.com/opencobra>

Instructions on how to use git are available here (<http://gitref.org/> for \*nix users and <https://windows.github.com> for windows users).

Please raise any new issues you wish to have fixed here: <https://github.com/opencobra/cobratoolbox/issues> That is also a good place to tell the community what you want from the next version of the cobra toolbox and why.

I am looking for committed developers to actively engage in the project in particular domains especially where I am not sufficiently active to maintain the code on my own, e.g., bioengineering, but also where people are interested to develop new and maintain existing parts of the COBRA toolbox.

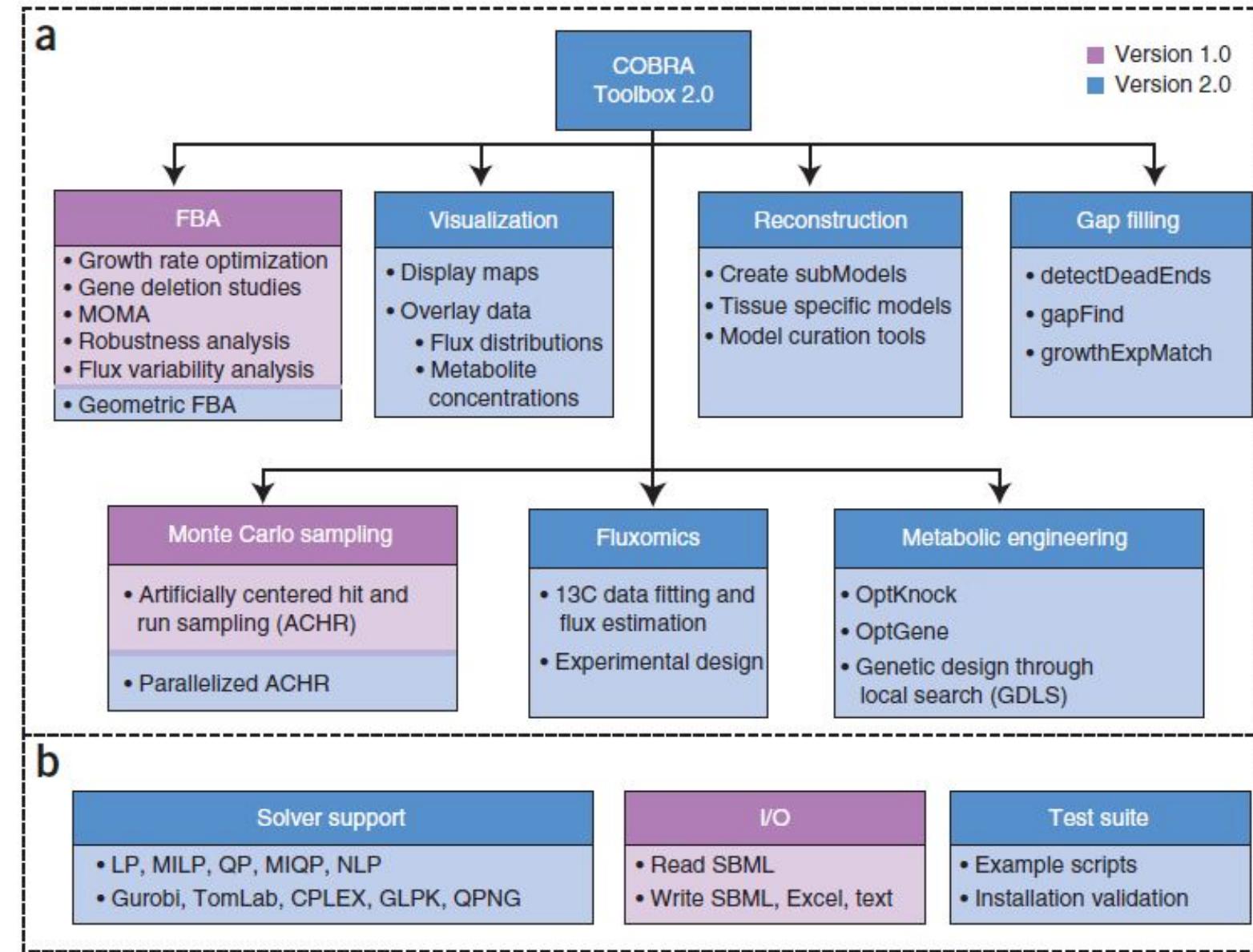
I am delighted to be supporting Nathan Lewis, Department of Pediatrics in the UC San Diego School of Medicine in his role as chief steward of the The openCOBRA Project (<http://opencobra.github.io/>), which we agree should be a welcome home to COBRA code in a language agnostic manner.

Regards,

*Dr. Ronan M.T. Fleming*  
Luxembourg Centre for Systems Biomedicine,  
University of Luxembourg



# Overview of the Cobra Toolbox



Schellenberger, J., R. Que, et al. (2011). "Quantitative prediction of cellular metabolism with constraint-based models: the COBRA Toolbox v2.0." Nature protocols 6(9): 1290-1307.



# Cobra Toolbox Documentation

## Matlab Index

### Matlab Directories

- Cobra
- Cobra/design
- Cobra/external
- Cobra/external/BuildMPS
- Cobra/external/m2html
- Cobra/fluxomics
- Cobra/fluxomics/c13solver
- Cobra/gapFilling
- Cobra/gapFilling/KEGG
- Cobra/gapFilling/growthExpMatch
- Cobra/io
- Cobra/maps
- Cobra/maps/tools
- Cobra/rFBA
- Cobra/reconstruction
- Cobra/reconstruction/mass\_balance
- Cobra/reconstruction/mass\_balance/basicPhysicochemicalData
- Cobra/sampling
- Cobra/solvers
- Cobra/tools

Generated by [m2html](#) © 2003

<http://opencobra.github.io/cobratoolbox/docs/index.html>

## optimizeCbModel

### PURPOSE

**optimizeCbModel** Solve a flux balance analysis problem

### SYNOPSIS

```
function FBAsolution = optimizeCbModel(model,osenseStr,minNorm,allowLoops)
```

### DESCRIPTION

**optimizeCbModel** Solve a flux balance analysis problem

Solves LP problems of the form: max/min  $c^*v$   
subject to  $S^*v = b$   
 $lb \leq v \leq ub$   
FBAsolution = optimizeCbModel(model,osenseStr,minNormFlag)

#### INPUT

model (the following fields are required - others can be supplied)  
S Stoichiometric matrix  
b Right hand side = dx/dt  
c Objective coefficients  
lb Lower bounds  
ub Upper bounds

#### OPTIONAL INPUTS

osenseStr Maximize ('max')/minimize ('min') (opt, default = 'max')

minNorm {(0), 'one', > 0, n x 1 vector}, where [m,n]=size(S);  
0 Default, normal LP  
'one' Minimise the Taxicab Norm using LP.  
min |v|  
s.t.  $S^*v = b$   
 $c^*v = f$   
 $lb \leq v \leq ub$

-----  
The remaining options work only with a valid QP solver:  
-----

> 0 Minimises the Euclidean Norm of internal fluxes.  
Typically 1e-6 works well.



# Matlab GUI

MATLAB R2014b

The screenshot shows the MATLAB R2014b interface. The Current Folder browser on the left lists several M-files related to metabolic reconstruction, including `enumerateOptimalSolutions.m`, `extractCompModel.m`, and `initCobraToolbox_Tomlab_CPLEX.m`. The Command Window in the center displays the output of running the `initCobraToolbox_Tomlab_CPLEX` function, which initializes the COBRA Toolbox and sets up various solver paths. The Workspace browser on the right shows variables like `files` (a 1x5 cell), `fname` (the path to the toolbox), `i` (the value 5), and `NumericDisplay` and `NumericFormat` (both set to empty strings).

Current Folder

- Name
- enumerOptimalSolutions.m
- extractCompModel.m
- extractMetModel.m
- extractSubNetwork.m
- extractSubSysModel.m
- findBlockedReaction.m
- findEpistaticInteractions.m
- findExcRxns.m
- findGenelDs.m
- findGenesFromRxns.m
- findMetabolicJunctions.m
- findMetIDs.m
- findNeighborRxns.m
- findRxnIDs.m
- findRxsnsFromGenes.m
- findRxsnsFromMets.m
- findSubSysGen.m

initCobraToolbox\_Tomlab\_CPLEX.m (Function)

initCobraToolbox Initialize COnstraint-Based Reconstruction and Analysis Toolbox

initCobraToolbox\_Tomlab\_CPLEX()

MATLAB R2014b

Command Window

```
THE CURRENT DIRECTORY IS C:\tomlab
Found the base path as C:\tomlab\base
Found the cgo path as C:\tomlab\cgo
Found the testprob path as C:\tomlab\testprob
Found the examples path as C:\tomlab\examples
Found the optim path as C:\tomlab\optim

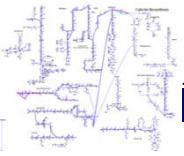
Warning - optimization toolbox drop-in replacements
Remove tomlab\optim from path to disable

Found the mex path as C:\tomlab\mex
Found the splines path as C:\tomlab\splines
Found the quickguide path as C:\tomlab\quickguide
Found the MAD path as C:\tomlab\mad
Found the lib path as C:\tomlab\lib
Found the usersguide path as C:\tomlab\usersguide
Found the ampl path as C:\tomlab\ampl
Found the common path as C:\tomlab\common
Found the finance path as C:\tomlab\finance
Found the modellib path as C:\tomlab\modellib
MAD successfully installed
Found the PROPT path as C:\tomlab\propt
Found the TOMSYM path as C:\tomlab\tomsym
Found the modellib path as C:\tomlab\modellib

Run tomblablic to display license information
Done with setting TOMLAB paths
>> initCobraToolbox_Tomlab_CPLEX
LP solver set to tomlab_cplex successful
MILP solver set to tomlab_cplex successful
QP solver set to tomlab_cplex successful
MIQP solver set to tomlab_cplex successful
CB map output set to svg successful
fx >>
```

Workspace

Name	Value
files	1x5 cell
fname	'C:\Program Files...
i	5
NumericDisplay	"
NumericFormat	"

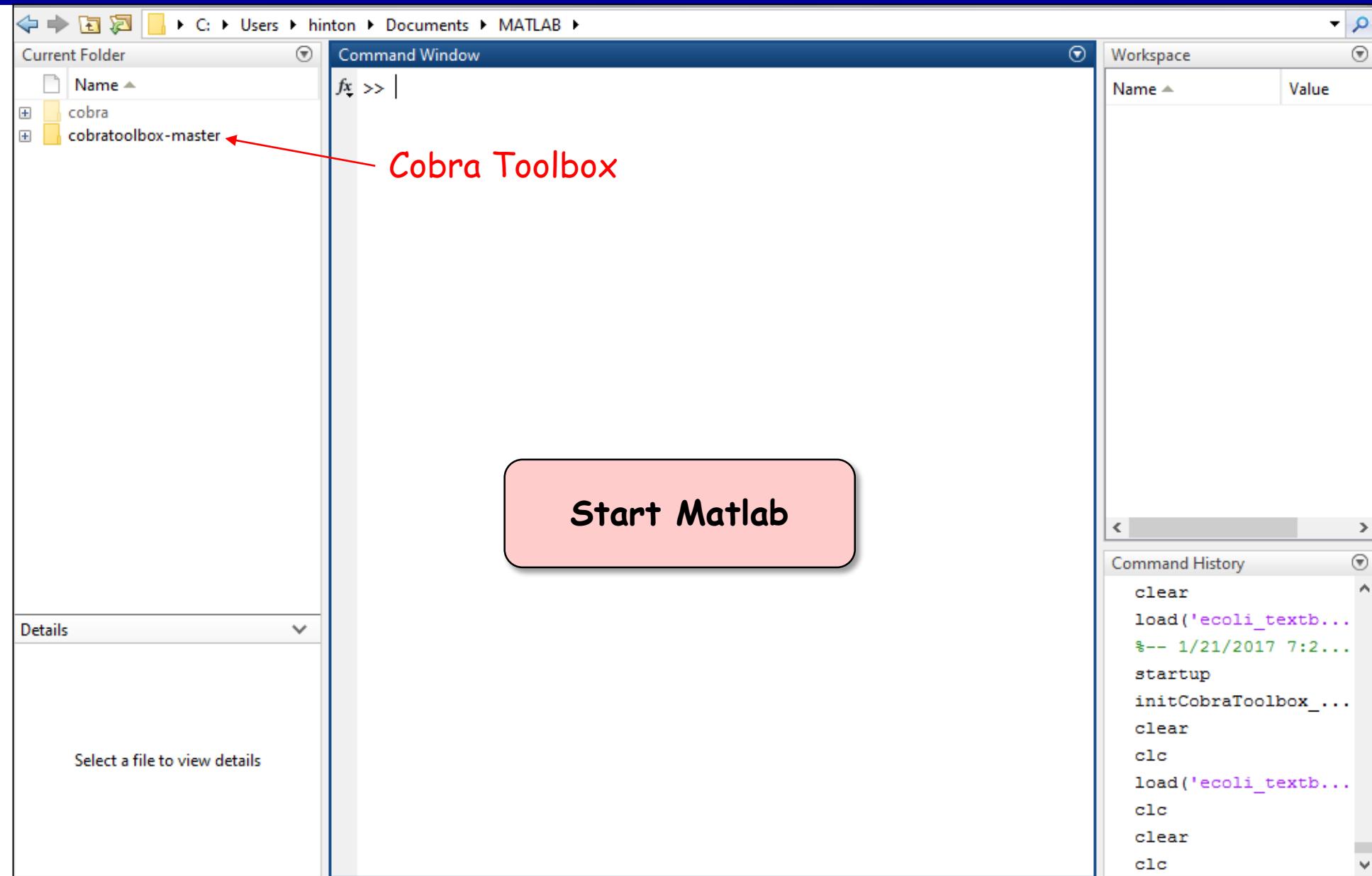


## Cobra Toolbox

- Cobra Toolbox Overview
- • Cobra Toolbox Fundamentals
- Cobra Examples
- BIGG Database
- Paint4Net



## Starting Matlab

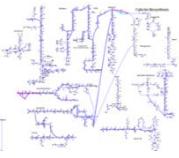




## Initializing the Cobra Toolbox

The screenshot shows a MATLAB interface with two main panes: 'Current Folder' and 'Command Window'.  
In the 'Current Folder' pane, a list of MATLAB files is shown, including 'initCobraToolbox.m' which is highlighted with a gray selection bar.  
In the 'Command Window' pane, the following text is displayed:  
>> initCobraToolbox  
Define LP solver...  
LP solver: gurobi6  
  
Define MILP solver...  
MILP solver: gurobi6  
  
Define QP solver...  
QP solver: gurobi6  
  
Define MIQP solver...  
MIQP solver: gurobi6  
  
Define CB map output...  
CB map output: svg  
  
TranslateSBML worked with the test .xml file: Ecoli\_core\_ECOSAL.xml  
fx >> |

**Run initCobraToolbox.m**



## Loading a Cobra-based Model

The screenshot shows the MATLAB environment with the following components:

- Current Folder:** Displays a list of files including `Cobra Toolbox Matlab Files`, `Paint4Net v1.3`, and several M-files related to metabolic modeling.
- Command Window:** Shows the command `>> load('ecoli_textbook.mat')` being run, followed by the output which describes the `model` variable as a `struct` with various fields.
- Workspace:** Shows the `model` variable is a `1x1 struct`.
- Annotations:** Red arrows point from the text labels to specific parts of the Command Window output:
  - Load Command:** Points to the first line of code in the Command Window.
  - Reactions:** Points to the `rxns` field.
  - Metabolites:** Points to the `mets` field.
  - S matrix:** Points to the `S` field.
  - Objective Function:** Points to the `rev` field.
  - Lower Bounds:** Points to the `lb` field.
  - Upper Bounds:** Points to the `ub` field.
- Text Box:** A pink box highlights the command `load('ecoli_textbook.mat');`



## Cobra Model Structure

Double click on Model to open window with the model variables

The screenshot shows the MATLAB environment with the following windows:

- Current Folder:** Shows files in the directory C:\Users\hinton\OneDrive\USU\Course\Cobra Toolbox\Cobra Toolbox Matlab Files - 2017. The file "ecoli\_textbook.mat" is selected.
- Variables - model:** A central window showing the contents of the "model" variable, which is a 1x1 struct with 27 fields. Fields include: modelVersion, rxns, mets, S, rev, c, metNames, metFormulas, lb, ub, metCharge, rules, genes, rxnGeneMat, grRules, subSystems, confidenceScores, rxnReferences, and ECN.
- Workspace:** Shows the "model" variable as a 1x1 struct.
- Command Window:** Displays the command "load('ecoli\_textbook.mat')".
- History:** Shows the command history including "clc", "load('ecoli\_textbook.mat')", "model", "clear", "load('ecoli\_textbook.mat')", "startup", "initCobraToolbox", "clear", "clc", and "load('ecoli\_textbook.mat')".



# Model Structure Used by the Cobra Toolbox

- **rxns**: A list of all of the reaction abbreviations in the same order they appear in the stoichiometric matrix
- **mets**: A list of all of the metabolite abbreviations in the model in the same order they appear in the stoichiometric matrix
- **S**: The stoichiometric matrix in sparse format
- **lb**: The lower bound corresponding to each reaction, in order
- **ub**: The upper bound corresponding to each reaction, in order
- **c**: The relative weight of each reaction in the objective function—often a single one in the position corresponding to the biomass reaction and zeros elsewhere
- **subSystem**: The metabolic subsystem for each reaction
- **rules**: Boolean rules for each reaction describing the gene-reaction relationship. For example 'gene1 and gene2' indicate that the two gene products are part of an enzyme whereas 'gene1 or gene2' indicate that the two gene products are isozymes that catalyze the same reaction.
- **genes**: The gene names of all the genes included in the model
- **rxnGeneMat**: A matrix with as many rows as there are reactions in the model and as many columns as there are genes in the model. The  $i^{th}$  row and  $j^{th}$  column contains a one if the  $j^{th}$  gene in **genes** is associated with the  $i^{th}$  reaction in **rxns** and zero otherwise.



# Model Spreadsheet

ecoli\_core\_model.xls [writeCbModel(model, 'xls')]

abbreviation	officialName	equation	subSystem
ACALD	acetaldehyde dehydrogenase (acetylating)	[c] : acald + coa + nad <==> accoa + h + nadh	Pyruvate Metabolism
ACALDt	acetaldehyde reversible transport	acald[e] <==> acald[c]	Transport, Extracellular
ACKr	acetate kinase	[c] : ac + atp <==> actp + adp	Pyruvate Metabolism
ACONTa	aconitase (half-reaction A, Citrate hydro-lyase)	[c] : cit <==> acon-C + h2o	Citric Acid Cycle
ACONTb	aconitase (half-reaction B, Isocitrate hydro-lyase)	[c] : acon-C + h2o <==> icit	Citric Acid Cycle
ACt2r	acetate reversible transport via proton symport	ac[e] + h[e] <==> ac[c] + h[c]	Transport, Extracellular
ADK1	adenylate kinase	[c] : amp + atp <==> (2) adp	Oxidative Phosphorylation
AKGDH	2-Oxoglutarate dehydrogenase	[c] : akg + coa + nad --> co2 + nadh + succoa	Citric Acid Cycle
AKGt2r	2-oxoglutarate reversible transport via symport	akg[e] + h[e] <==> akg[c] + h[c]	Transport, Extracellular
ALCD2x	alcohol dehydrogenase (ethanol)	[c] : etoh + nad <==> acald + h + nadh	Pyruvate Metabolism
ATPM	ATP maintenance requirement	[c] : atp + h2o --> adp + h + pi	Oxidative Phosphorylation
ATPS4r	ATP synthase (four protons for one ATP)	adp[c] + (4) h[e] + pi[c] <==> atp[c] + (3) h[c] + h2o[c]	Oxidative Phosphorylation
CO2t	CO2 transporter via diffusion	co2[e] <==> co2[c]	Transport, Extracellular
CS	citrate synthase	[c] : accoa + h2o + oaa --> cit + coa + h	Citric Acid Cycle
CYTBD	cytochrome oxidase bd (ubiquinol-8: 2 protons)	(2) h[c] + (0.5) o2[c] + q8h2[c] --> (2) h[e] + h2o[c] + q8[c]	Oxidative Phosphorylation
D_LACt2	D-lactate transport via proton symport	h[e] + lac-D[e] <==> h[c] + lac-D[c]	Transport, Extracellular
ENO	enolase	[c] : 2pg <==> h2o + pep	Glycolysis/Gluconeogenesis
ETOHt2r	ethanol reversible transport via proton symport	etoh[e] + h[e] <==> etoh[c] + h[c]	Transport, Extracellular
EX_ac(e)	Acetate exchange	[e] : ac <==>	Exchange
EX_acald(e)	Acetaldehyde exchange	[e] : acald <==>	Exchange
EX_akg(e)	2-Oxoglutamate exchange	[e] : akg <==>	Exchange



## Solving for Fluxes

The screenshot shows the MATLAB interface with the following components:

- Current Folder:** Displays a list of files including 'Cobra Toolbox Matlab Files' and 'Paint4Net v1.3'. The file 'ecoli\_textbook.mat' is selected.
- Command Window:** Shows the MATLAB code used to solve the metabolic model:

```
>> load('ecoli_textbook.mat')
>> solution=optimizeCbModel(model,'max',0,0)
solution =
stru_
    x: [95x1 double]
    f: 0.8739
    y: [72x1 double]
    w: [95x1 double]
    stat: 1
    origStat: 1
    solver: 'tomlab_cplex'
    time: 0.5270
```
- Workspace:** Shows the variables 'model' and 'solution' as 1x1 structures.
- Structure used to store solution**: A red arrow points from the text above to the 'solution' variable in the workspace.
- Command History:** Displays the command history at the bottom of the window.

A callout box highlights the line of code: **solution = optimizeCbModel(model, 'max', 0, 0);**



## Structure of Optimization Solution

The screenshot shows the MATLAB environment with several windows open:

- Current Folder:** Shows files related to the Cobra Toolbox, including various M-files and an Excel file.
- Variables - solution:** A table showing the contents of the `solution` struct.

Field	Value
x	95x1 double
f	0.8739
y	72x1 double
w	95x1 double
stat	1
origStat	1
solver	'tomlab_cplex'
time	0.5270
- Workspace:** Shows the `model` and `solution` variables.
- Command Window:** Displays the MATLAB code used to load the model and run the optimization.

Red arrows point from the labels to the corresponding fields in the `solution` struct table:

- Flux Vector → f
- Objective Function Value → f
- Shadow Prices → y
- Reduced Costs → w
- Solver → solver

```
>> load('ecoli_textbook.mat');
>> solution=optimizeCbModel(model,'max',0.0)
solution =
  struct with fields:
    x: [95x1 double]
    f: 0.8739
    y: [72x1 double]
    w: [95x1 double]
```



## Printing the Optimized Flux Solutions

The screenshot shows the MATLAB interface with the following details:

- Current Folder:** Displays files including `Cobra Toolbox Matlab Files`, `Paint4Net v1.3`, and `ecoli_textbook.mat`.
- Command Window:** Shows the command `>> printFluxVector(model, solution.x, true)` and its output. The output lists various metabolic reactions with their flux values. The last value listed is `0.873922`, which is annotated with a red arrow and the label "Growth-rate".
- Workspace:** Shows two variables: `model` (1x1 struct) and `solution` (1x1 struct).
- Command History:** Lists the commands entered, including `initCobraToolbox`, `clear`, `clc`, `load('ecoli_textb...`, `clc`, `clear`, `clc`, `load('ecoli_textb...`, `solution=optimize...`, and `printFluxVector(m...`.

Annotations in red text and arrows:

- An arrow points from the text "Flux Passing through the Reaction (mmols g<sub>DW</sub><sup>-1</sup>h<sup>-1</sup>)" to the value `-10` next to the reaction `EX_glc(e)`.
- An arrow points from the text "Reaction Name" to the reaction `EX_glc(e)`.

A pink callout box at the bottom contains the command:

```
printFluxVector(model, solution.x, true)
```



# Print Flux Values

```
printFluxVector(model, solution.x, true)
```

ACONTa	6.00725	FBA	7.47738	PGK	-16.0235
ACONTb	6.00725	FUM	5.06438	PGL	4.95998
AKGDH	5.06438	G6PDH2r	4.95998	PGM	-14.7161
ATPM	8.39	GAPD	16.0235	PIt2r	3.2149
ATPS4r	45.514	GLCpts	10	PPC	2.50431
Biomass	...0.873922	GLNS	0.223462	PYK	1.75818
CO2t	-22.8098	GLUDy	-4.54186	RPE	2.67848
CS	6.00725	GND	4.95998	RPI	-2.2815
CYTBD	43.599	H2Ot	-29.1758	SUCDi	5.06438
ENO	14.7161	ICDHyr	6.00725	SUCOAS	-5.06438
EX_co2(e)	22.8098	MDH	5.06438	TALA	1.49698
EX_glc(e)	-10	NADH16	38.5346	TKT1	1.49698
EX_h2o(e)	29.1758	NH4t	4.76532	TKT2	1.1815
EX_h(e)	17.5309	O2t	21.7995	TPI	7.47738
EX_nh4(e)	-4.76532	PDH	9.28253		
EX_o2(e)	-21.7995	PFK	7.47738		
EX_pi(e)	-3.2149	PGI	4.86086		

Growth  
Rate

Inputs & Outputs  
(Exchange  
Reactions)

The generation time,  $g$ , is the cell doubling time  
 $g = \ln(2)/\text{growth-rate} = \ln(2)/0.873922 = 0.7931$  hours  
which means the cells double every 47.6 minutes



# printFluxVector

```
printFluxVector(model, solution.x)
```

ACALD	-14.6749
ACALDt	0
ACKr	-15.1732
ACONTa	0.507693
ACONTb	0.507693
ACT2r	-15.1732
ADK1	0
AKGDH	0
AKGt2r	0
ALCD2x	-14.6749
ATPM	8.39
ATPS4r	-11.1879
Biomass...	0.470565
CO2t	0.840759
CS	0.507693
CYTBD	0
D_LACT2	0
ENO	35.0451
ETOHt2r	-14.6749
EX_ac(e)	15.1732
EX_acald(e)	-0
EX_akg(e)	-0
EX_co2(e)	-0.840759
EX_etooh(e)	14.6749
EX_for(e)	32.1194
EX_fru(e)	-0
EX_fum(e)	-0
EX_glc(e)	-18.5
EX_gln_L(e)	-0

```
printFluxVector(model, solution.x, true)
```

ACALD	-14.6749
ACKr	-15.1732
ACONTa	0.507693
ACONTb	0.507693
ACT2r	-15.1732
ALCD2x	-14.6749
ATPM	8.39
ATPS4r	-11.1879
Biomass...	0.470565
CO2t	0.840759
CS	0.507693
ENO	35.0451
ETOHt2r	-14.6749
EX_ac(e)	15.1732
EX_co2(e)	-0.840759
EX_etooh(e)	14.6749
EX_for(e)	32.1194
EX_glc(e)	-18.5
EX_h2o(e)	-12.0879
EX_h(e)	56.7321
EX_nh4(e)	-2.5659
EX_pi(e)	-1.73107

```
printFluxVector(model, solution.x, true,true)
```

Biomass...	0.470565
EX_ac(e)	15.1732
EX_co2(e)	-0.840759
EX_etooh(e)	14.6749
EX_for(e)	32.1194
EX_glc(e)	-18.5
EX_h2o(e)	-12.0879
EX_h(e)	56.7321
EX_nh4(e)	-2.5659
EX_pi(e)	-1.73107

Only prints Exchange Reactions that are nonzero

Only prints Reactions that are nonzero

[http://opencobra.sourceforge.net/openCOBRA/opencobra\\_documentation/cobra\\_toolbox\\_2/index.html](http://opencobra.sourceforge.net/openCOBRA/opencobra_documentation/cobra_toolbox_2/index.html)



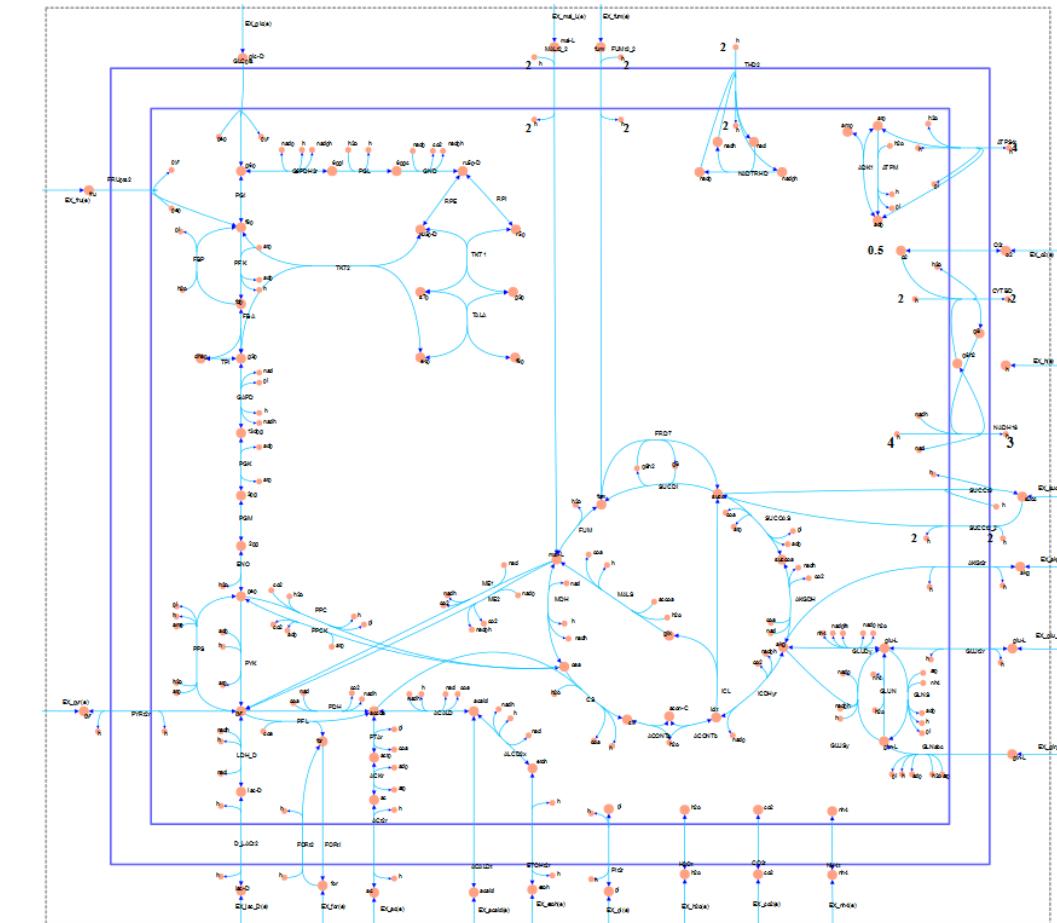
# Drawing Maps

- A map for the *E.coli* core, "ecoli\_Textbook\_ExportMap," model is available in the Matlab files folder available on the course website.
- Read the map into Cobra

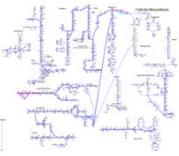
```
map=readCbMap('ecoli_Textbook_ExportMap');
```

- Cobra saves a map into a file named "target.svg" that will be written into the current directory.

```
drawCbMap(map);
```

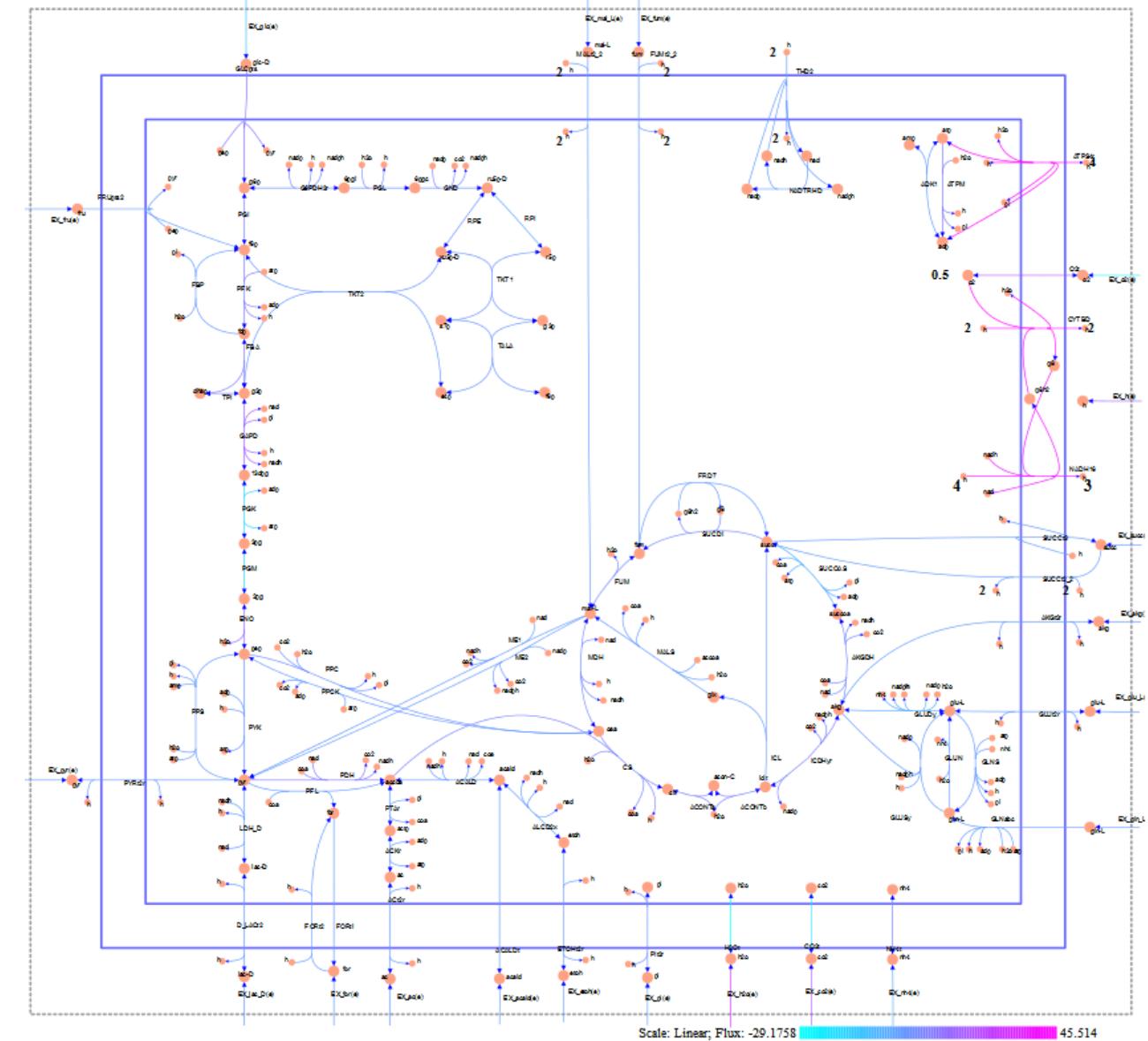


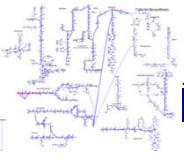
target.svg



# Drawing Flux Values onto a Map

`drawFlux(map, model, solution.x)`





# Matlab Editor

- The Matlab editor allows m-files to be created
  - m-files can contain a list of Matlab commands to be executed sequentially.

## Comments

The screenshot shows the MATLAB Editor interface with several windows open. The top menu bar includes 'EDITOR', 'PUBLISH', and 'VIEW'. The toolbar contains icons for 'New', 'Open', 'Save', 'Find Files', 'Compare', 'Go To', 'Comment', 'Breakpoints', 'Run' (with dropdown), 'Run Section', 'Run and Advance', and 'Run and Time'. Below the toolbar, tabs for 'FILE', 'NAVIGATE', 'EDIT', and 'BREAKPOINTS' are visible. The main workspace displays the code for 'AerobicGlucoseBioMass.m'. Red annotations highlight specific features:

- A red arrow points to the 'clear;' command in line 6 with the text "Clear the Workspace".
- A red arrow points to the 'Run' button in the toolbar with the text "Run m-file".
- A red arrow points to the 'AerobicGlucoseBioMass.m' tab with the text "Supports multiple m-file windows".
- A red arrow points to the bottom-left corner of the code window with the text "Large code area".

```
% Script to determine aerobic growth rate of E.coli on glucose
% Taken from "What is flux balance analysis? - Supplementary tutorial"
% by J. D. Orth, I. Thiele, & B. O. Palsson, Nature Biotechnology,
% Volume 28, Number 3, March 2010

clear; % Clear the Workspace

% Input the E.coli core model

load('ecoli_textbook.mat');

% The growth of E. coli on glucose can be simulated under aerobic conditions.
% To set the maximum glucose uptake rate to 18.5 mmol gDW-1 hr-1
% (millimoles per gram dry cell weight per hour, the default flux
% units used in the COBRA Toolbox), enter:

model = changeRxnBounds(model,'EX_glc(e)',-5,'l');

% This changes the lower bound ('l') of the glucose exchange reaction to
% -18.5, a biologically realistic uptake rate. By convention, the import
% of a metabolite is a negative flux. To allow unlimited oxygen uptake, enter:

model = changeRxnBounds(model,'EX_o2(e)',-10,'l');

% By setting the lower bound of the oxygen uptake reaction to such a
% large number, it is practically unbounded.

% Set optimization objective to Biomass_Ecoli_core_N(w/GAM)_Nmet2
```



# Cobra Toolbox

- Cobra Toolbox Overview
- Cobra Toolbox Fundamentals
- • Cobra Examples
- BIGG Database
- Paint4Net



# Aerobic *E.coli* Growth on Glucose

("What is flux balance analysis? - Supplementary tutorial")

Set the maximum glucose uptake rate to  $18.5 \text{ mmol gDW}^{-1} \text{ hr}^{-1}$  (millimoles per gram dry cell weight per hour, the default flux units used in the COBRA Toolbox), enter:

```
model = changeRxnBounds(model, 'EX_glc(e)', -18.5, 'l');
```

This changes the lower bound ('l') of the glucose exchange reaction to -18.5, a biologically realistic uptake rate (note that the import of a metabolite is listed as a negative flux). To allow unlimited oxygen uptake, enter:

```
model = changeRxnBounds(model, 'EX_o2(e)', -1000, 'l');
```

By setting the lower bound of the oxygen uptake reaction to such a large number, it is practically unbounded. Set the biomass reaction is set as the objective function, enter:

```
model = changeObjective(model, 'Biomass_Ecoli_core_N(w/GAM) Nmet2');
```

To perform FBA with maximization of the biomass reaction as the objective, enter:

```
FBAsolution = optimizeCbModel(model, 'max');
```

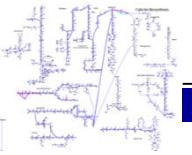
FBAsolution.f then gives the value of the objective function ( $Z$ ) as 1.6531 (the model predicts a growth rate of  $1.6531 \text{ hr}^{-1}$ ). The flux distribution vector FBAsolution.x shows that there is high flux in the glycolysis, pentose phosphate, TCA cycle, and oxidative phosphorylation pathways, with no secreted organic by-products. See **AerobicGlucoseBioMass.m**



# E.coli Textbook Spreadsheet

(ecoli\_textbook.xls)

A	B	C	D	E	F	G
Abbreviation	Description	Reaction	GPR	Genes	Proteins	Subsystem
1 ACALD	acetaldehyde dehydrogenase (acetylating)	acald[c] + coa[c] + nad[c] <=> acco (b0351) or (b1241)	b0351 b1241			Pyruvate Metabolism
3 ACALDt	acetaldehyde reversible transport	acald[e] <=> acald[c]	(s0001)	s0001		Transport, Extracellular
4 ACKr	acetate kinase	ac[c] + atp[c] <=> actp[c] + adp[c]	(b2296) or (b1849)	b1849 b2296 b3115		Pyruvate Metabolism
5 ACONTa	aconitase (half-reaction A, Citrate hydro-lyase)	cit[c] <=> acon-C[c] + h2o[c]	(b0118) or (b1276)	b0118 b1276		Citric Acid Cycle
6 ACONTb	aconitase (half-reaction B, Isocitrate hydro-lyase)	acon-C[c] + h2o[c] <=> icit[c]	(b0118) or (b1276)	b0118 b1276		Citric Acid Cycle
7 ACt2r	acetate reversible transport via proton symport	ac[e] + h[e] <=> ac[c] + h[c]				Transport, Extracellular
8 ADK1	adenylate kinase	amp[c] + atp[c] <=> 2 adp[c]	(b0474)	b0474		Oxidative Phosphorylation
9 AKGDH	2-Oxoglutarate dehydrogenase	akg[c] + coa[c] + nad[c] -> co2[c] + (b0727) and (b0116 b0726 b0727)				Citric Acid Cycle
10 AKGt2r	2-oxoglutarate reversible transport via symport	akg[e] + h[e] <=> akg[c] + h[c]	(b2587)	b2587		Transport, Extracellular
11 ALCD2x	alcohol dehydrogenase (ethanol)	etoh[c] + nad[c] <=> acald[c] + h[c]	(b0356) or (b1241)	b0356 b1241 b1478		Pyruvate Metabolism
12 ATPM	ATP maintenance requirement	atp[c] + h2o[c] -> adp[c] + h[c] + pi[c]				Oxidative Phosphorylation
13 ATPS4r	ATP synthase (four protons for one ATP)	adp[c] + 4 h[e] + pi[c] <=> atp[c] + (b3739) and (b3731 b3732 b3733 b3734)				Oxidative Phosphorylation
14 Biomass_Ecoli_core_N(w/GAM)_Nmet2	core E. coli biomass equation (Neidhardt Based with	1.496 3pg[c] + 3.7478 accoa[c] + 59.81 atp[c] + 0.361 e4p[c] + 0.0709 f6p[c] + 0.129 g3p[c] + 0.205 g6p[c] + 0.2557				
15 CO2t	CO2 transporter via diffusion	co2[e] <=> co2[c]	(s0001)	s0001		Transport, Extracellular
16 CS	citrate synthase	accoa[c] + h2o[c] + oaa[c] -> cit[c]	(b0720)	b0720		Citric Acid Cycle
17 CYTBD	cytochrome oxidase bd (ubiquinol-8: 2 protons)	2 h[c] + 0.5 o2[c] + q8h2[c] -> h2o[ (b0979 and b0978)	b0973 b0974 b0978 b0979			Oxidative Phosphorylation
18 D_LACT2	D-lactate transport via proton symport	h[e] + lac-D[e] <=> h[c] + lac-D[c]	(b2975) or (b3603)	b2975 b3603		Transport, Extracellular
19 ENO	enolase	2pg[c] <=> h2o[c] + pep[c]	(b2779)	b2779		Glycolysis/Gluconeogenesis
20 ETOHt2r	ethanol reversible transport via proton symport	etoh[e] + h[e] <=> etoh[c] + h[c]				Transport, Extracellular



# "AerobicGlucoseBioMass.m" Matlab Script

```
clear; % Clear Workspace  
  
load('ecoli_textbook.mat'); % Load textbook model  
  
model = changeRxnBounds(model,'EX_glc(e)',-18.5,'l'); % Set lower bound of glucose  
model = changeRxnBounds(model,'EX_o2(e)',-1000,'l'); % Set lower bound of oxygen  
model = changeObjective(model,'Biomass_Ecoli_core_N(w/GAM)_Nmet2'); % Set objective function (Biomass)  
  
FBAsolution = optimizeCbModel(model,'max',0,0) % Find optimized flux values  
  
map=readCbMap('ecoli_Textbook_ExportMap'); % Input ecoli textbook map template  
options.zeroFluxWidth = 0.1;  
options.rxnDirMultiplier = 10;  
drawFlux(map, model, FBAsolution.x, options); % Draw Map  
  
printFluxVector(model, FBAsolution.x, true); % Print flux values
```



# "AerobicGlucoseBioMass.m" Matlab Output

```
>> AerobicGlucoseBioMass  
FBAsolution =  
    x: [95x1 double]  
    f: 1.6531  
    stat: 1  
    origStat: 1  
    solver: 'tomlab_cplex'  
    time: 0.1010  
  
Document Written  
ACONTa      10.3657  
ACONTb      10.3657  
AKGDH       8.5822  
ATPM        8.39  
ATPS4r      80.6069  
Biomass_Ecoli_core_N(w/GAM)_Nmet2  
CO2t        -40.6527  
CS          10.3657  
CYTBD       77.4832  
ENO         26.8391  
EX_co2(e)   40.6527  
EX_glc(e)   -18.5  
EX_h2o(e)   52.6943  
EX_h(e)     33.1606  
EX_nh4(e)   -9.01387  
EX_o2(e)    -38.7416  
EX_pi(e)    -6.08116
```

"optimizeCbModel"  
Output

Growth-rate

1.65307

Flux

Reactions

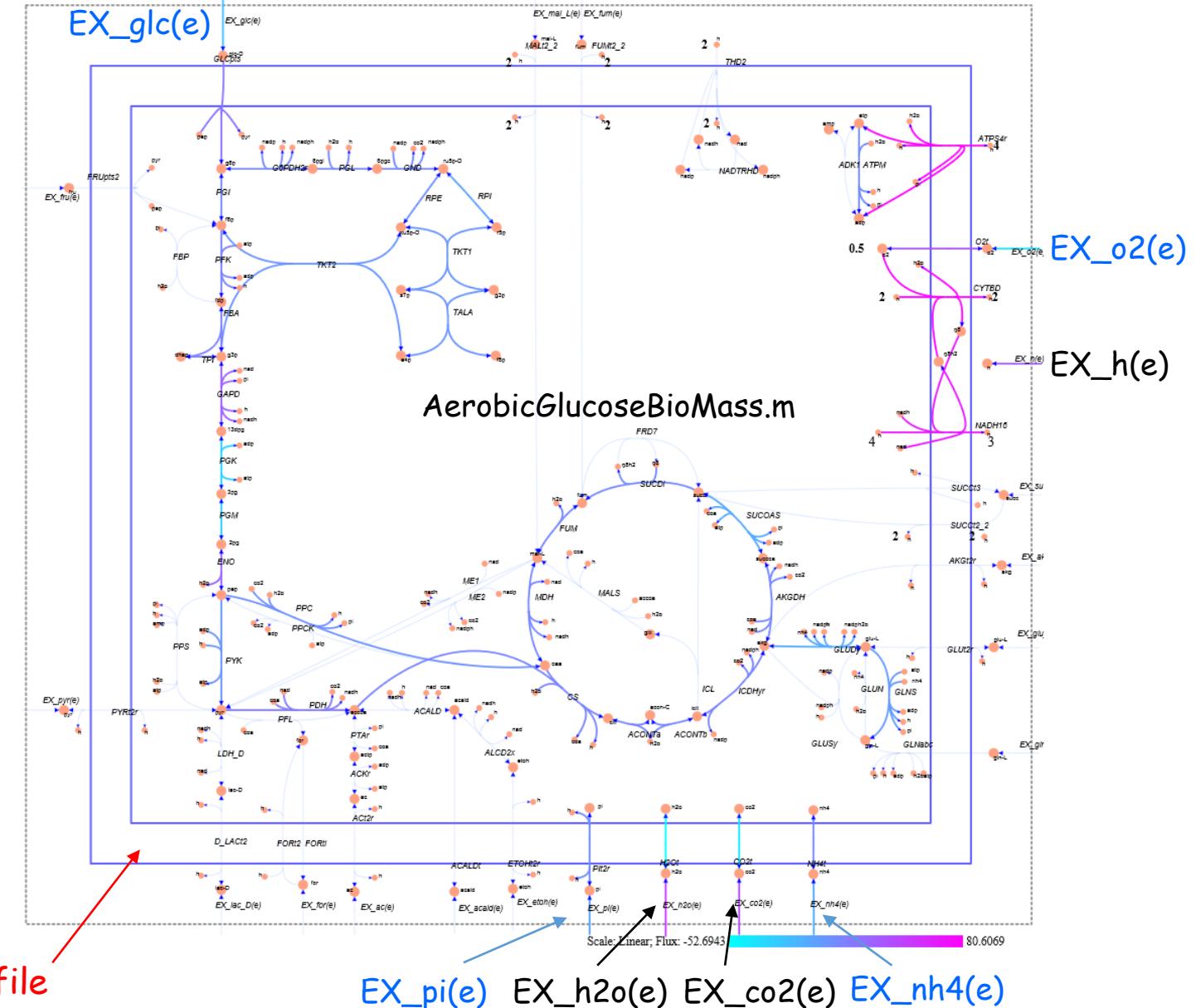
FBA	13.5621	SUCOAS	-8.5822
FUM	8.5822	TALA	2.99786
G6PDH2r	9.88077	TKT1	2.99786
GAPD	29.312	TKT2	2.4011
GLCpts	18.5	TPI	13.5621
GLNS	0.42269		
GLUDy	-8.59118		
GND	9.88077		
H2Ot	-52.6943		
ICDHyr	10.3657		
MDH	8.5822		
NADH16	68.901		
NH4t	9.01387		
O2t	38.7416		
PDH	16.5611		
PFK	13.5621		
PGI	8.28035		
PGK	-29.312		
PGL	9.88077		
PGM	-26.8391		
PIt2r	6.08116		
PPC	4.73704		
PYK	2.7439		
RPE	5.39895		
RPI	-4.48182		
SUCDi	8.5822		



# Aerobic Growth on Glucose

## Exchange Reactions

Biomass	1.65
EX_co2(e)	40.6527
EX_glc(e)	-18.5
EX_h2o(e)	52.6943
EX_h(e)	33.1606
EX_nh4(e)	-9.01387
EX_o2(e)	-38.7416
EX_pi(e)	-6.08116





# Anaerobic *E.coli* Growth on Glucose

("What is flux balance analysis? - Supplementary tutorial")

Set the maximum glucose uptake rate to 18.5 mmol gDW-1 hr-1 (millimoles per gram dry cell weight per hour, the default flux units used in the COBRA Toolbox), enter:

```
model = changeRxnBounds(model, 'EX_glc(e)', -18.5, 'l');
```

This changes the lower bound ('l') of the glucose exchange reaction to -18.5, a biologically realistic uptake rate (note that the import of a metabolite is listed as a negative flux). To prevent oxygen uptake, enter:

```
model = changeRxnBounds(model, 'EX_o2(e)', 0, 'l'); % The 0 sets the flux value of o2 to zero
```

Anaerobic operation is achieved by setting the lower bound of the oxygen uptake reaction to zero.

Set the biomass reaction is set as the objective function, enter:

```
model = changeObjective(model, 'Biomass_Ecoli_core_N(w/GAM)_Nmet2', 0, 0);
```

To perform FBA with maximization of the biomass reaction as the objective, enter:

```
FBAsolution = optimizeCbModel(model, 'max');
```

FBAsolution.f then gives the value of the objective function ( $Z$ ) as 0.4706 (the model predicts a growth rate of 0.4706 hr-1). The flux distribution vector FBAsolution.x shows that oxidative phosphorylation is not used in these conditions, and that acetate, formate, and ethanol are produced by fermentation pathways. See **AnaerobicGlucoseBioMass.m**



# Anaerobic Growth on Glucose

## Exchange Reactions

Biomass 0.470565

EX\_ac(e) 15.1732

EX\_co2(e) -0.840759

EX\_etoh(e) 14.6749

EX\_for(e) 32.1194

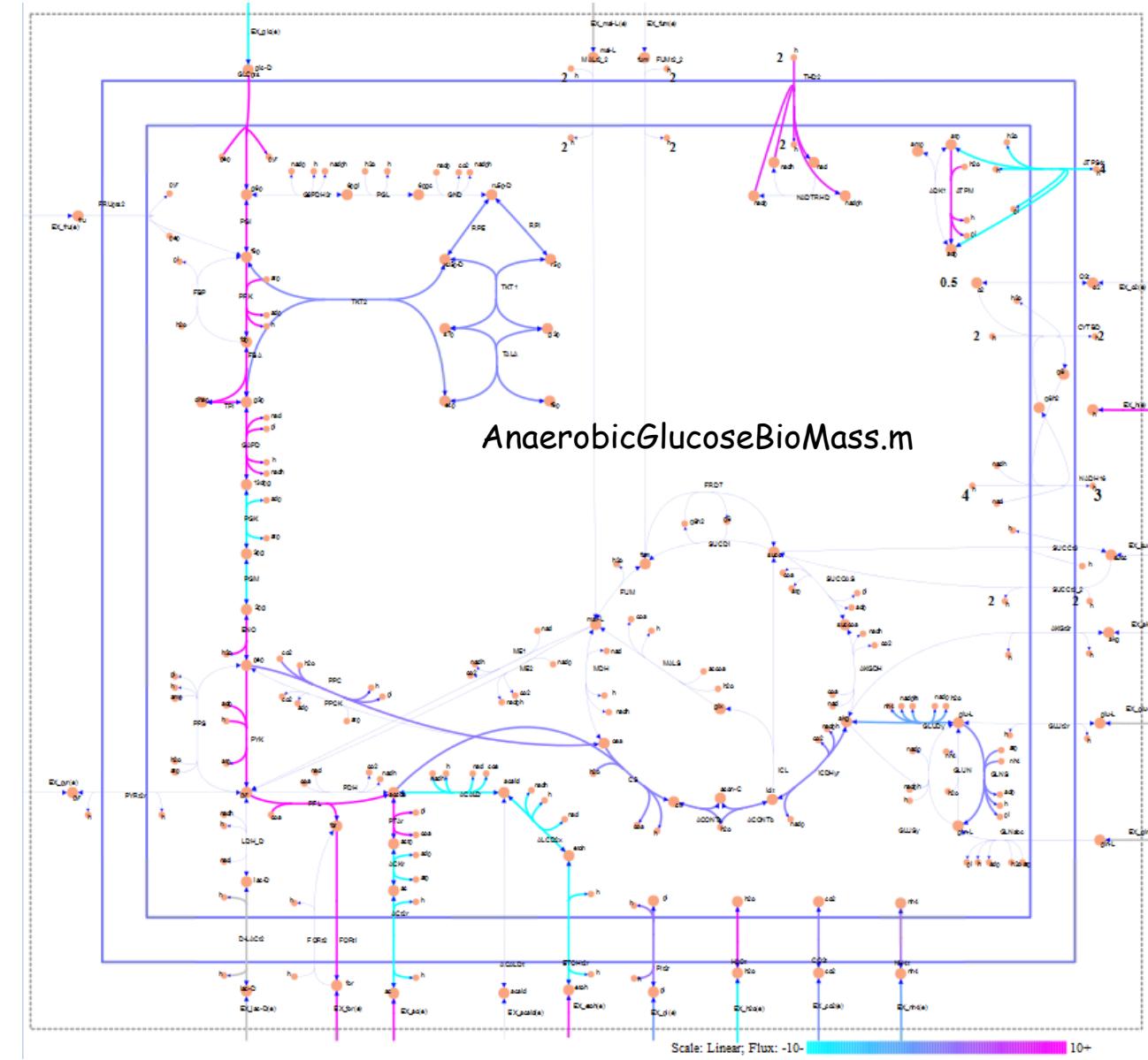
EX\_glc(e) -18.5

EX\_h2o(e) -12.0879

EX\_h(e) 56.7321

EX\_nh4(e) -2.5659

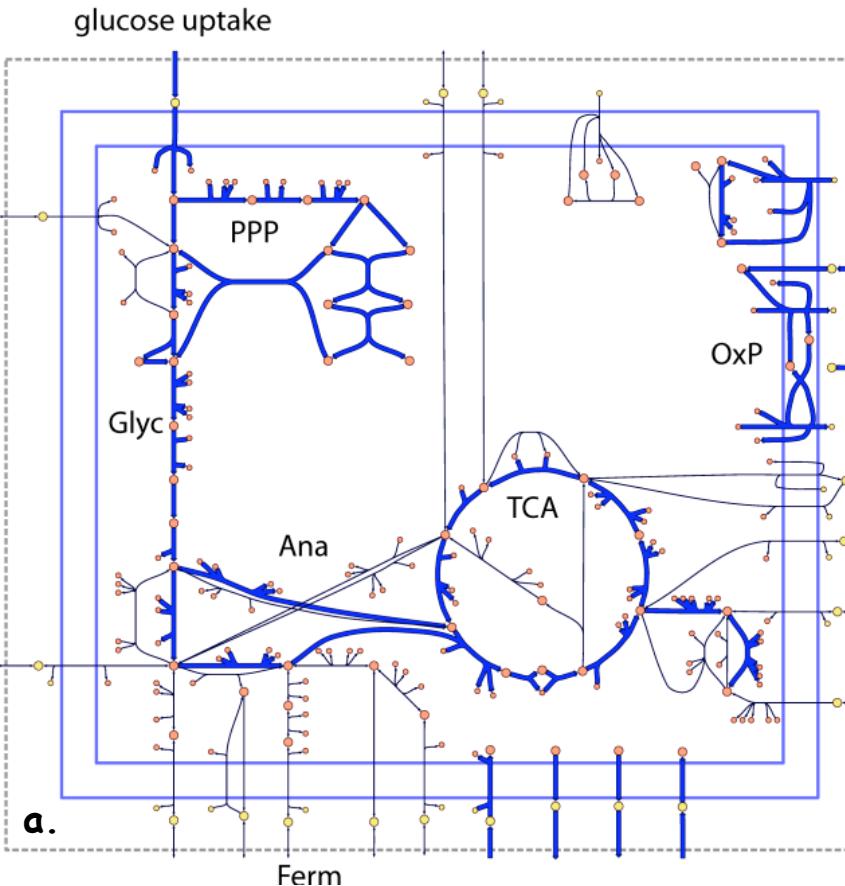
EX\_pi(e) -1.73107



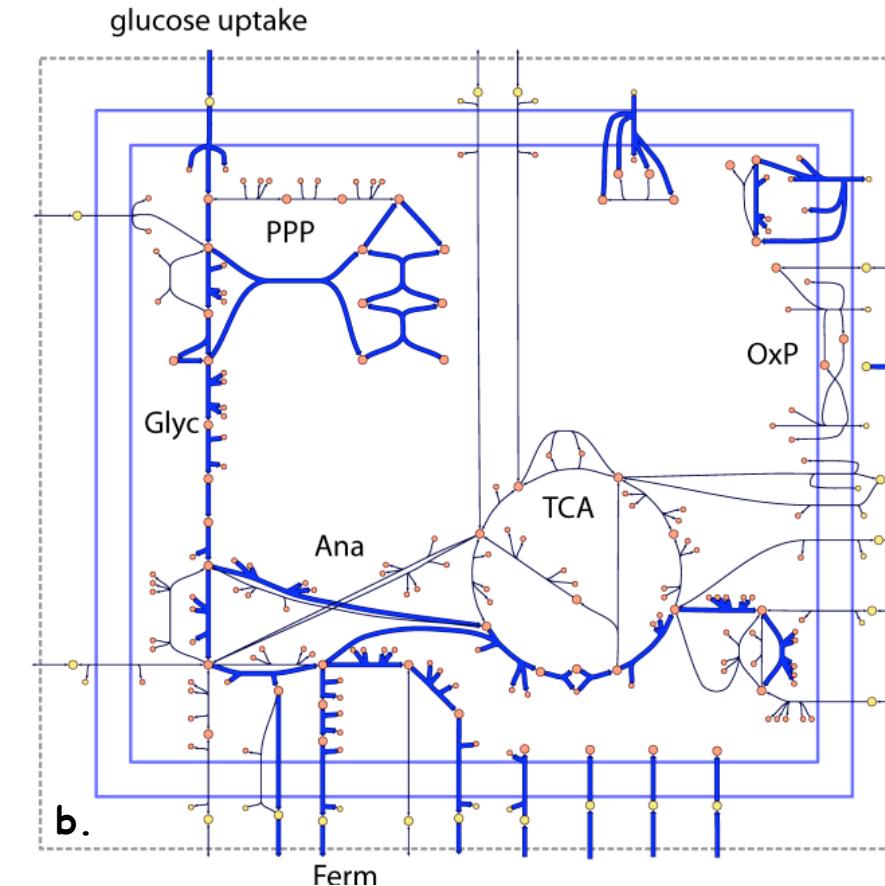


# Aerobic vs. Anaerobic Growth

Orth, J. D., I. Thiele, et al. (2010). "What is flux balance analysis?" Nature biotechnology 28(3): 245-248.



Aerobic Growth



Anaerobic Growth



# Growth on Alternate Substrates

("What is flux balance analysis? - Supplementary tutorial")

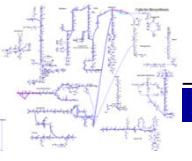
The core *E. coli* model contains exchange reactions for 13 different organic compounds, each of which can be used as the sole carbon source under aerobic conditions. For example, to simulate growth on succinate instead of glucose

```
model = changeRxnBounds(model, 'EX_glc(e)', 0, 'l'); % Required if glucose uptake is zero  
model = changeRxnBounds(model, 'EX_succ(e)', -20, 'l');  
FBAsolution = optimizeCbModel(model, 'max');
```

The growth rate, given by FBAsolution.f, will be  $0.8401 \text{ hr}^{-1}$

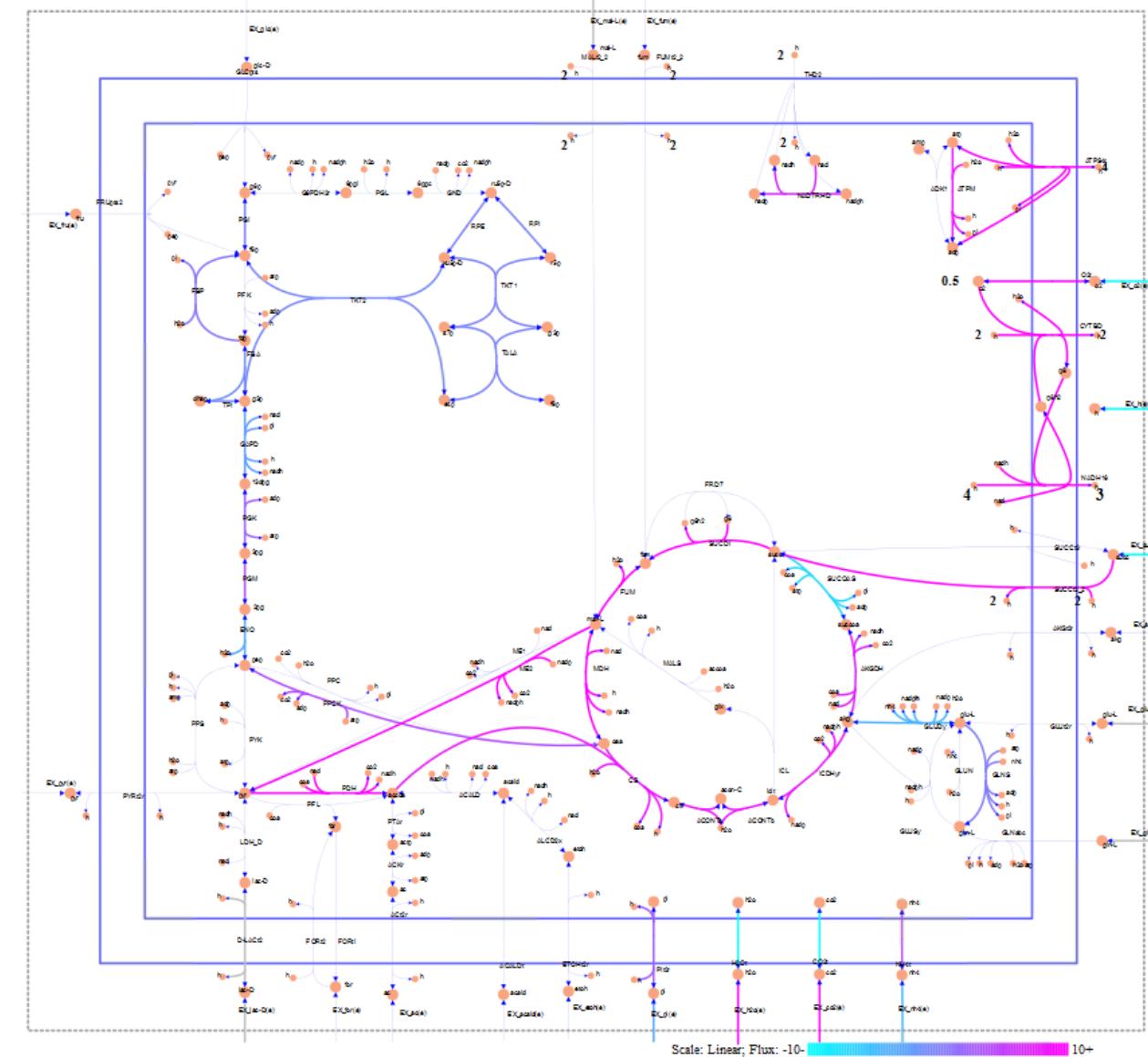
Growth can also be simulated under anaerobic conditions with any substrate by using changeRxnBounds to set the lower bound of the oxygen exchange reaction (EX\_o2(e)) to 0 mmol gDW $^{-1}$  hr $^{-1}$ , so no oxygen can enter the system. When this constraint is applied and succinate is the only organic substrate, optimizeCbModel returns a growth rate of 0 hr $^{-1}$ , and does not calculate a flux vector v (depending on which linear programming solver is used with the COBRA Toolbox, a growth rate may not be calculated at all). In this case, FBA predicts that growth is not possible on succinate under anaerobic conditions. Because the maximum amount of ATP that can be produced from this amount of succinate is less than the minimum bound of 8.39 mmol gDW $^{-1}$  hr $^{-1}$  of the ATP maintenance reaction, ATPM, there is no feasible solution.organic by-products.

See `AerobicSuccinateBioMass.m` and `AnaerobicSuccinateBioMass.m`



# Growth on Succinate

## AerobicSuccinateBioMass.m





# Substrate Maximum Growth Rate

The core *E. coli* model contains exchange reactions for 13 different organic compounds, each of which can be used as the sole carbon source under aerobic or anaerobic conditions.

For this table carbon source minimum uptake values should be set to -20 mmol/gDW·h. For aerobic conditions the minimum uptake for oxygen is  $EX_{o2}(e) = -1000$  mmol/gDW·h.

("What is flux balance analysis? - Supplementary tutorial")

Substrate	Aerobic (hr <sup>-1</sup> )	Anaerobic (hr <sup>-1</sup> )
acetate	0.3893	0
acetaldehyde	0.6073	0
2-oxoglutarate	1.0982	0
ethanol	0.6996	0
D-fructose	1.7906	0.5163
fumarate	0.7865	0
D-glucose	1.7906	0.5163
L-glutamine	1.1636	0
L-glutamate	1.2425	0
D-lactate	0.7403	0
L-malate	0.7865	0
pyruvate	0.6221	0.0655
succinate	0.8401	0



# Cobra Toolbox

- Cobra Toolbox Overview
- Cobra Toolbox Fundamentals
- Cobra Examples
- • BIGG Database
- Paint4Net



# Constraint-based Metabolic Reconstructions & Analysis

2017 H. Scott Hinton

38

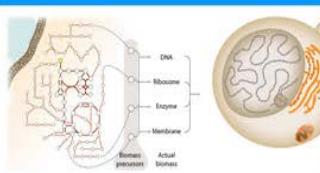
Systems Biology Research Group (<http://bigg.ucsd.edu>) About Advanced Search Web API

Welcome to the new BiGG! BiGG Models is a beta release, so please be patient if you encounter any issues. Learn more about BiGG Models beta.

# BiGG Models

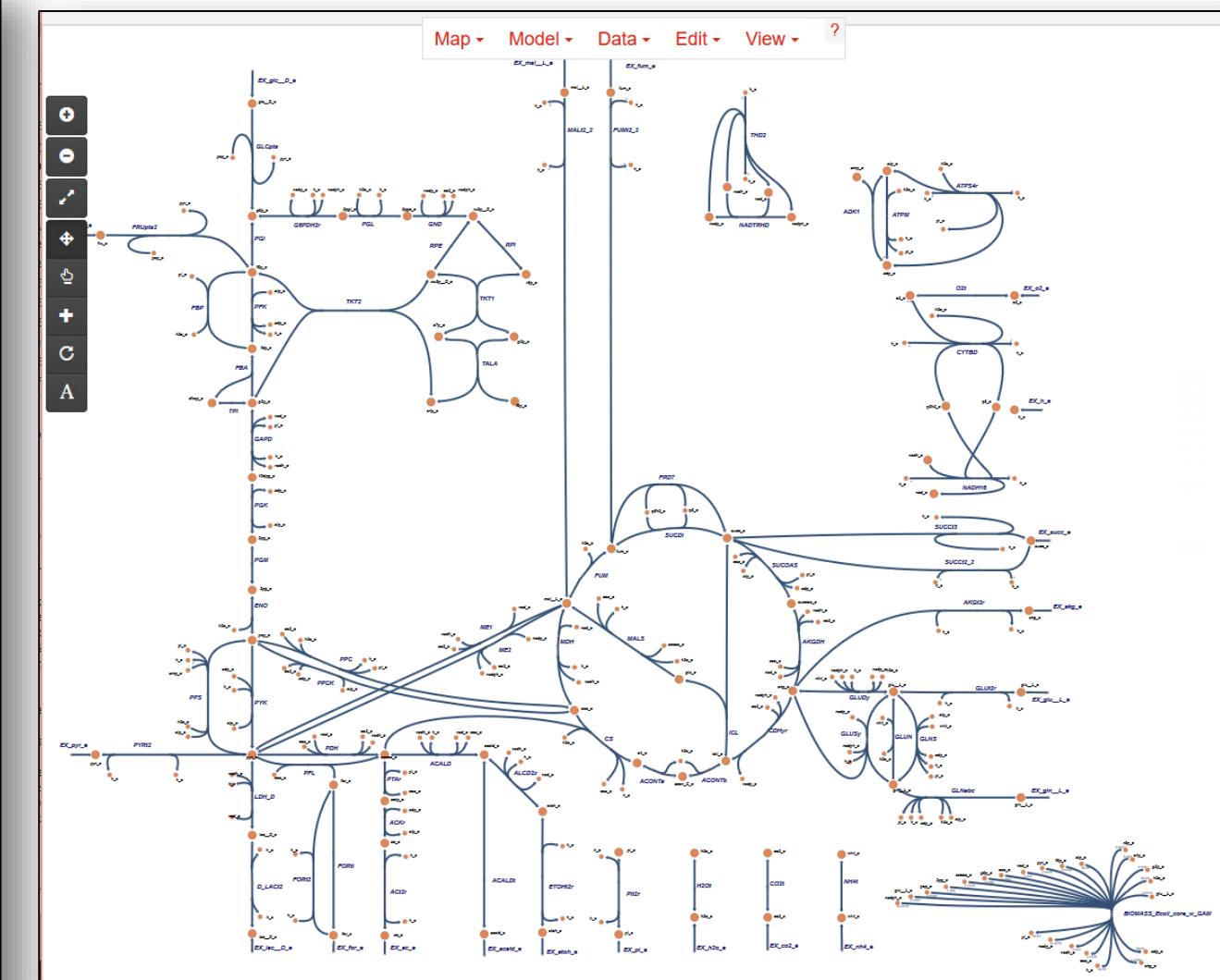
Search the database by model, reaction, metabolite, or gene ?

Search

  
View Models

  
View Metabolites

  
View Reactions



[http://bigg.ucsd.edu/models/e\\_coli\\_core](http://bigg.ucsd.edu/models/e_coli_core)

Schellenberger, J., J. O. Park, et al. (2010). "BiGG: a Biochemical Genetic and Genomic knowledgebase of large scale metabolic reconstructions." BMC Bioinformatics 11: 213.



# BIGG Models at <http://bigg.ucsd.edu/models>

## Models

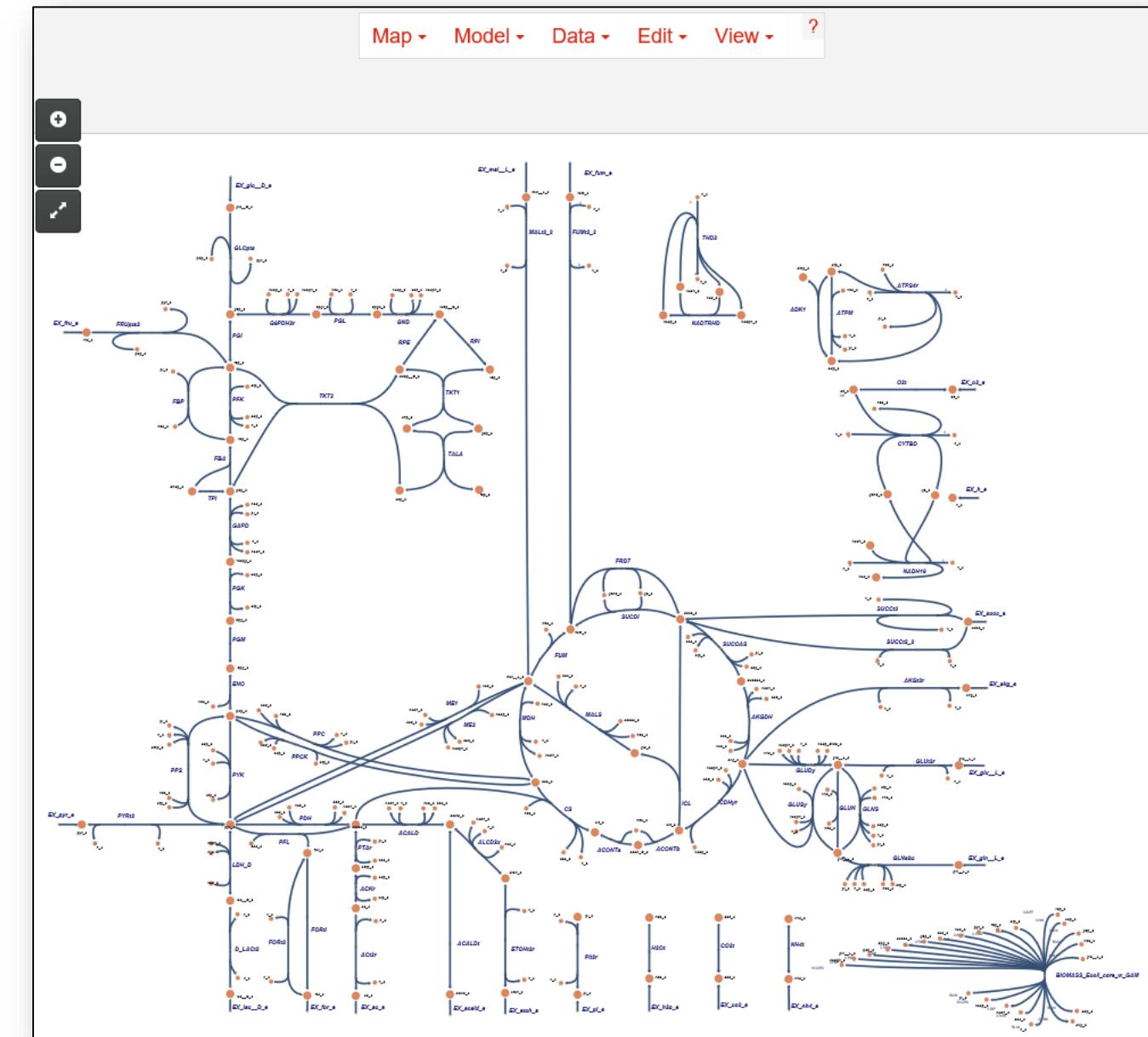
1 to 80 (80)					
BiGG ID	Organism	Metabolites	Reactions	Genes	
e_coli_core	Escherichia coli str. K-12 substr. MG1655	72	95	137	
iAB_RBC_283	Homo sapiens	342	469	346	
iAF1260	Escherichia coli str. K-12 substr. MG1655	1668	2382	1261	
iAF1260b	Escherichia coli str. K-12 substr. MG1655	1668	2388	1261	
iAF692	Methanosaerica barkeri str. Fusaro	628	690	692	
iAF987	Geobacter metallireducens GS-15	1109	1285	987	
iAPEC01_1312	Escherichia coli APEC O1	1942	2736	1313	
iAT_PLT_636	Homo sapiens	738	1008	636	
iB21_1397	Escherichia coli BL21(DE3)	1943	2742	1337	
iBWG_1329	Escherichia coli BW2952	1949	2742	1329	
ic_1306	Escherichia coli CFT073	1936	2727	1307	
iCHOv1	Cricetulus griseus	4456	6663	1766	
iE2348C_1286	Escherichia coli O127:H6 str. E2348/69	1919	2704	1287	
iEC042_1314	Escherichia coli 042	1926	2715	1314	
iEC55989_1330	Escherichia coli 55989	1953	2757	1330	
iECABU_c1320	Escherichia coli ABU 83972	1942	2732	1320	
iECB_1328	Escherichia coli B str. REL606	1951	2749	1329	
iECBD_1354	Escherichia coli 'BL21-Gold(DE3)pLysS AG'	1952	2749	1354	
iECD_1391	Escherichia coli BL21(DE3)	1943	2742	1333	



# Escher Visualization

<http://escher.github.io/>

The screenshot shows the Escher visualization interface. At the top right is the URL <http://escher.github.io/>. Below it is the large word "ESCHER". Underneath "ESCHER" is the tagline "Build, share, and embed visualizations of biological pathways." On the left, there is a "Filter by organism" dropdown set to "All". Below that are three sections: "Map" (set to "Core metabolism (e\_coli\_core)"), "Model (Optional)" (set to "e\_coli\_core"), and "Tool" (set to "Builder"). At the bottom left are three checkboxes: "Scroll to zoom (instead of scroll to pan)", "Never ask before reloading", and "Responsive pan and zoom". A large "Load map" button is at the bottom right.



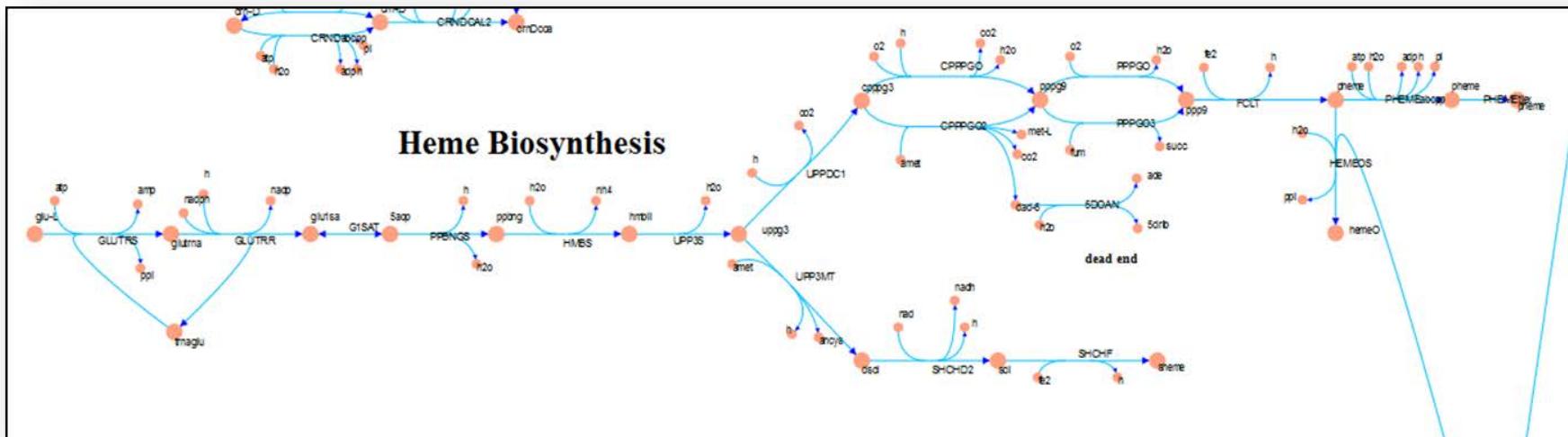


# Cobra Toolbox

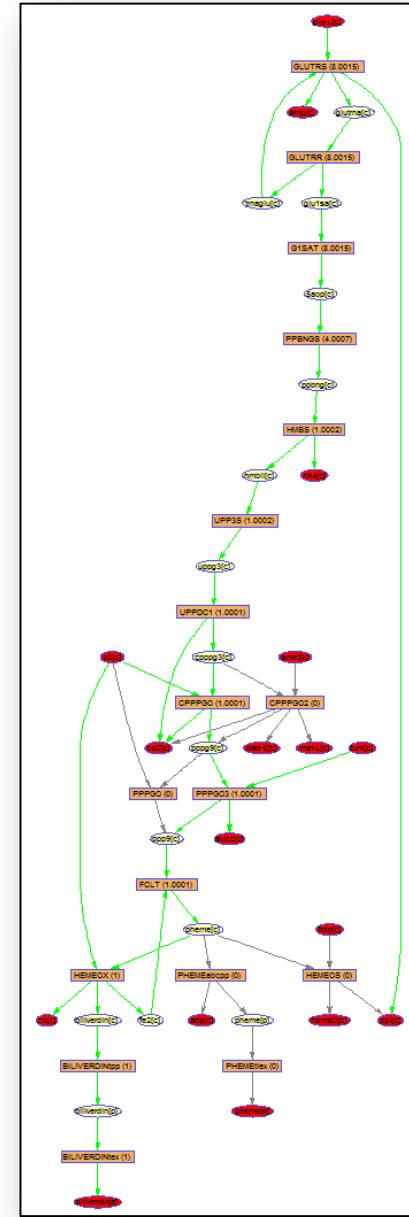
- Cobra Toolbox Overview
- Cobra Toolbox Fundamentals
- Cobra Examples
- BIGG Database
- • Paint4Net



# Visualization Tools



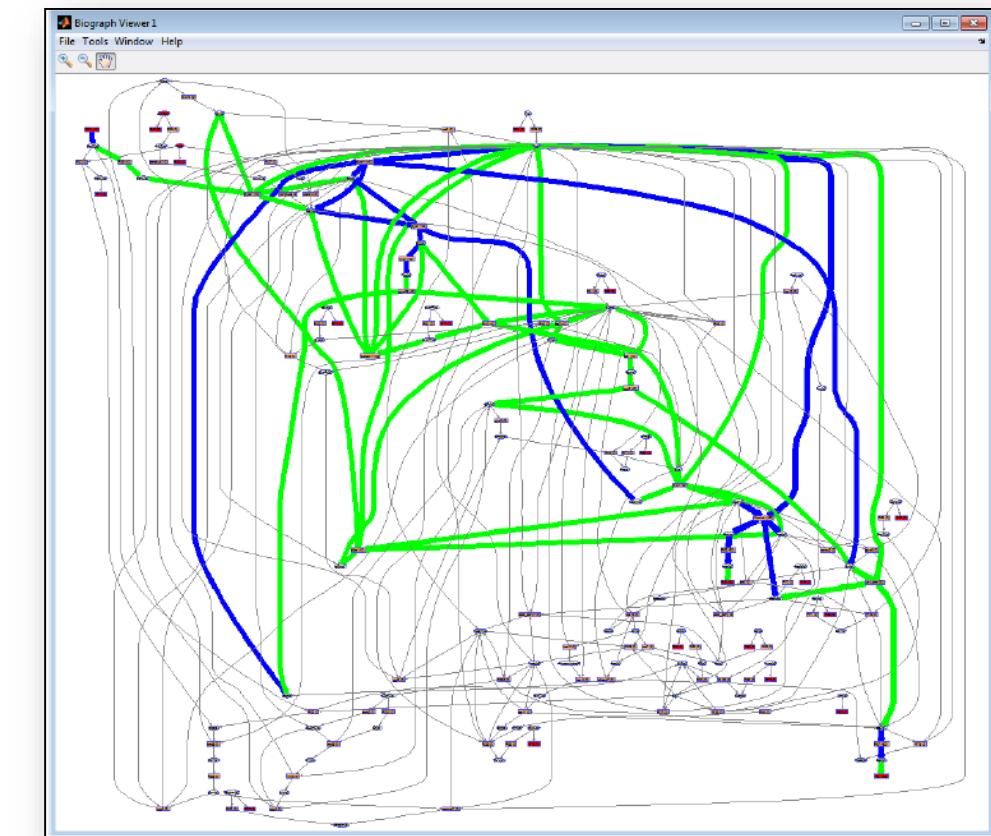
## Cobra Maps





# Paint4Net

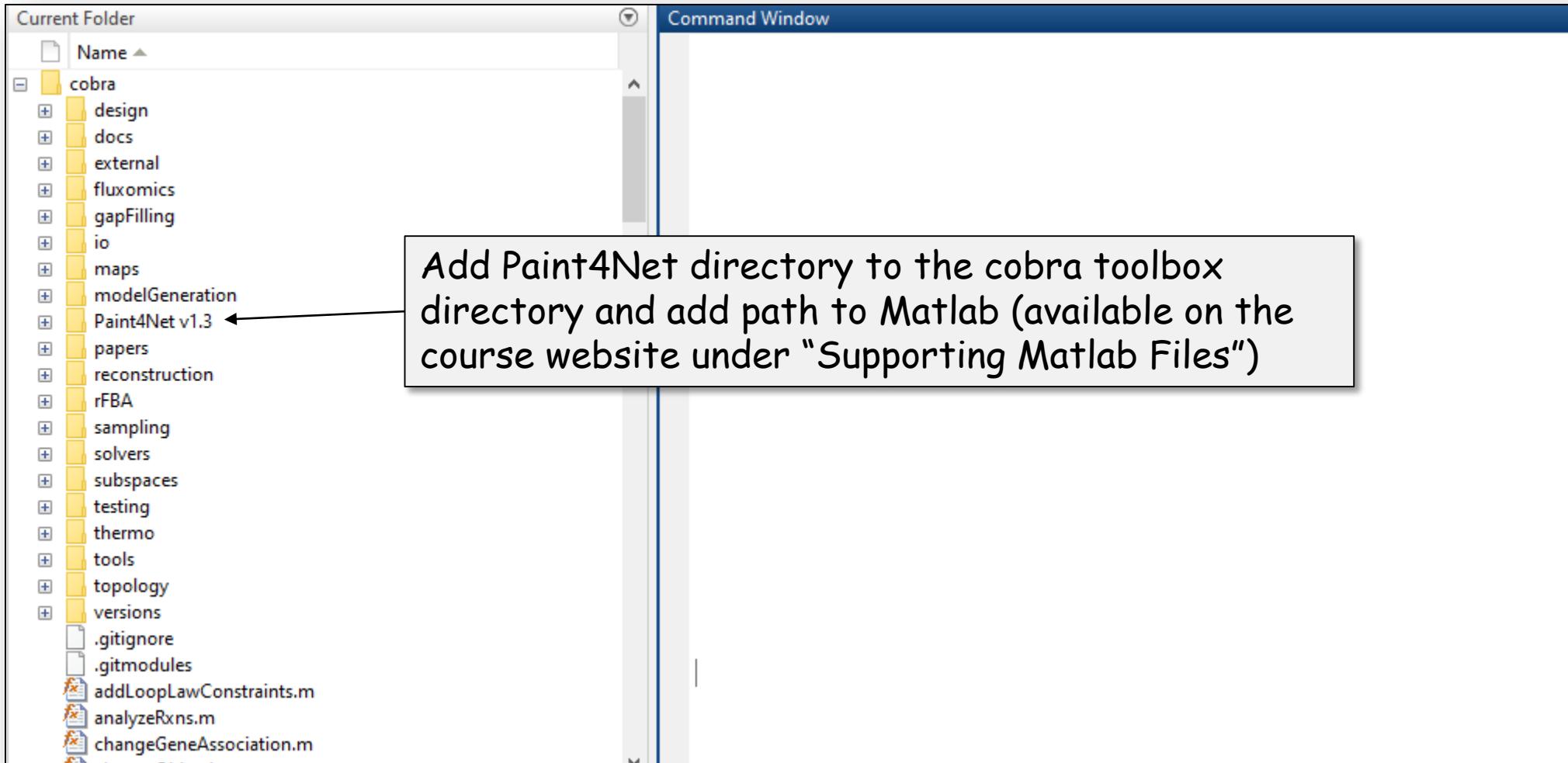
- Developed by Andrejs Kostromins
- Paint4Net is the COBRA Toolbox extension for visualization of constraints-based reconstruction and analysis (COBRA) models and reconstructions in the MATLAB environment.
- Uses the Bioinformatics toolbox to visualize COBRA models and reconstructions as a hypergraph.
- Paint4Net contains two main commands:
  - **draw\_by\_rxn**
    - For visualization of all or a part of a COBRA model by specified list of reactions.
  - **draw\_by\_met**
    - For visualization of the connectivity of a particular metabolite with other metabolites through reactions of a COBRA model



Kostromins, A. and E. Stalidzans (2012). "Paint4Net: COBRA Toolbox extension for visualization of stoichiometric models of metabolism." *Bio Systems*.



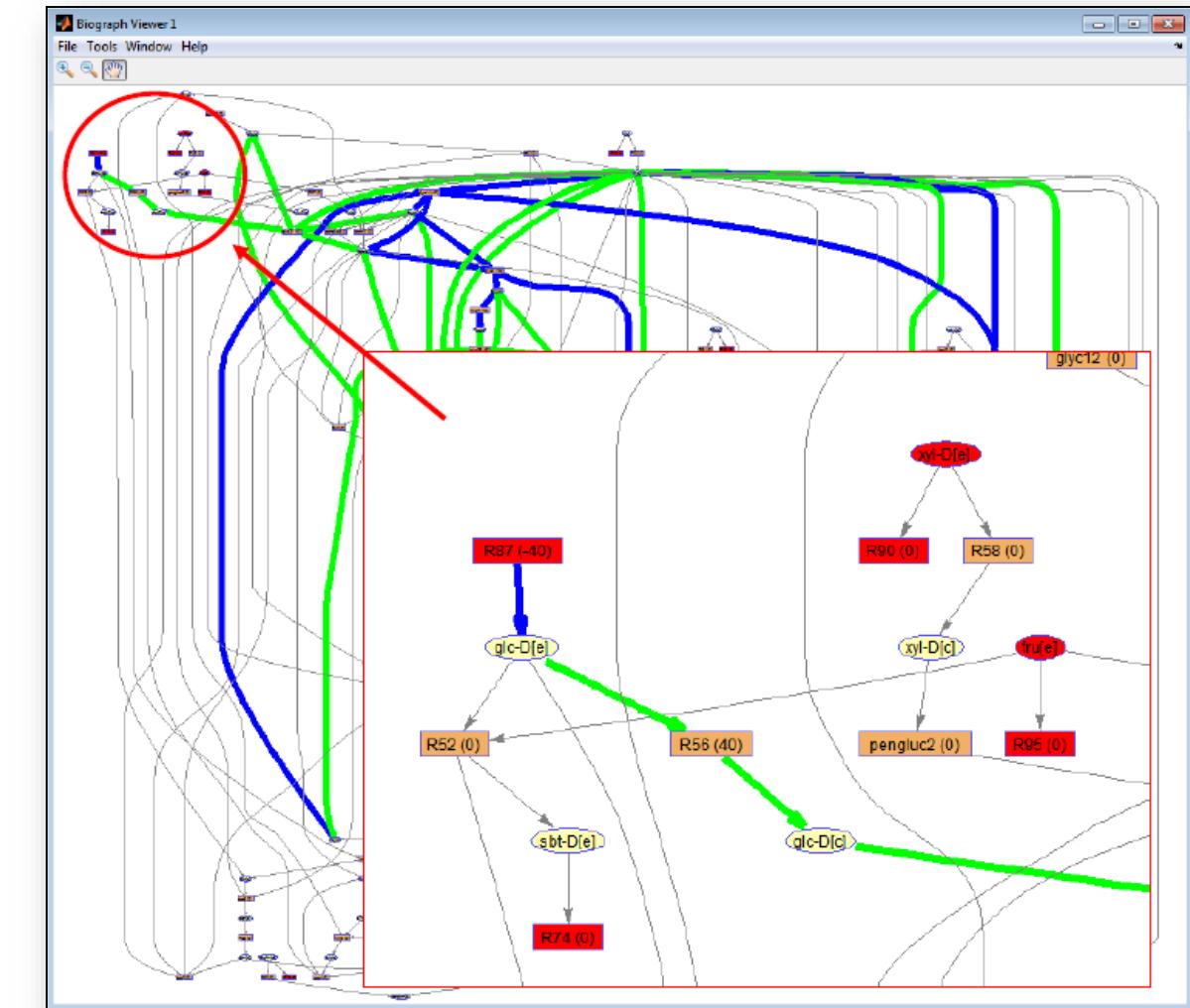
# Integrating Paint4Net with the Cobra Toolbox





# “draw\_by\_rxn” Cobra Toolbox Function

- Rectangles represent reactions;
- Numbers in rectangles represent flux rate through reaction.
- Red rectangles represent reactions with only one input or output flux (signaling a potential dead reaction);
- Ellipses represent metabolites;
- Red ellipses represent dead end metabolites;
- Grey edges represent zero-rate fluxes;
- Green edges represent positive-rate (forward) fluxes;
- Blue edges represent negative-rate (backward) fluxes.
- The thickness of the edges is calculated as percentage assuming the maximum rate of flux in the model corresponds to 100%.





# Plotting Textbook Model

VisualizeFlux\_Textbook.m

```
clear;

load('ecoli_textbook.mat');

model = changeRxnBounds(model,'EX_glc(e)',-5,'l');
model = changeRxnBounds(model,'EX_o2(e)',-20,'l');

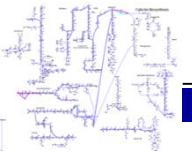
model = changeObjective(model,'Biomass_Ecoli_core_N(w/GAM)_Nmet2');

FBAsolution = optimizeCbModel(model,'max',0,0);

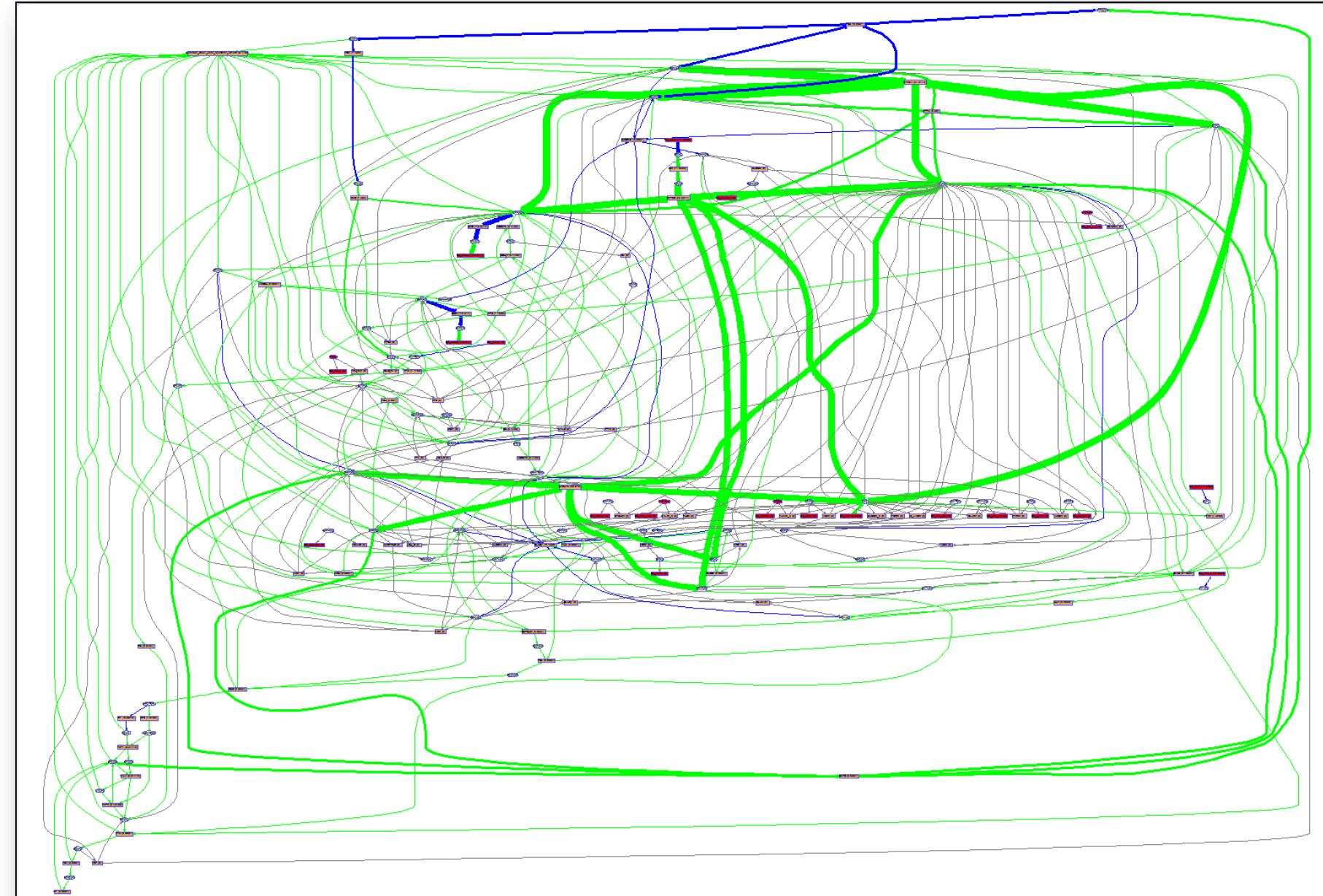
cofactors =
{'amp[c]','atp[c]','adp[c]','pi[c]','nad[c]','nadh[c]','nadph[c]','nadp[c]','h[c]','h2o[c]','co2[c]'

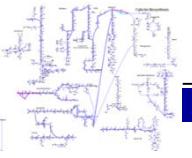
% Plot includes cofactors
[involvedMets,deadEnds]= draw_by_rxn (model,model.rxns,'true','struc',{''},{''},FBAsolution.x);

% Plot removes cofactors
[involvedMets,deadEnds]= draw_by_rxn (model,model.rxns,'true','struc',{''},cofactors,FBAsolution.x);
```

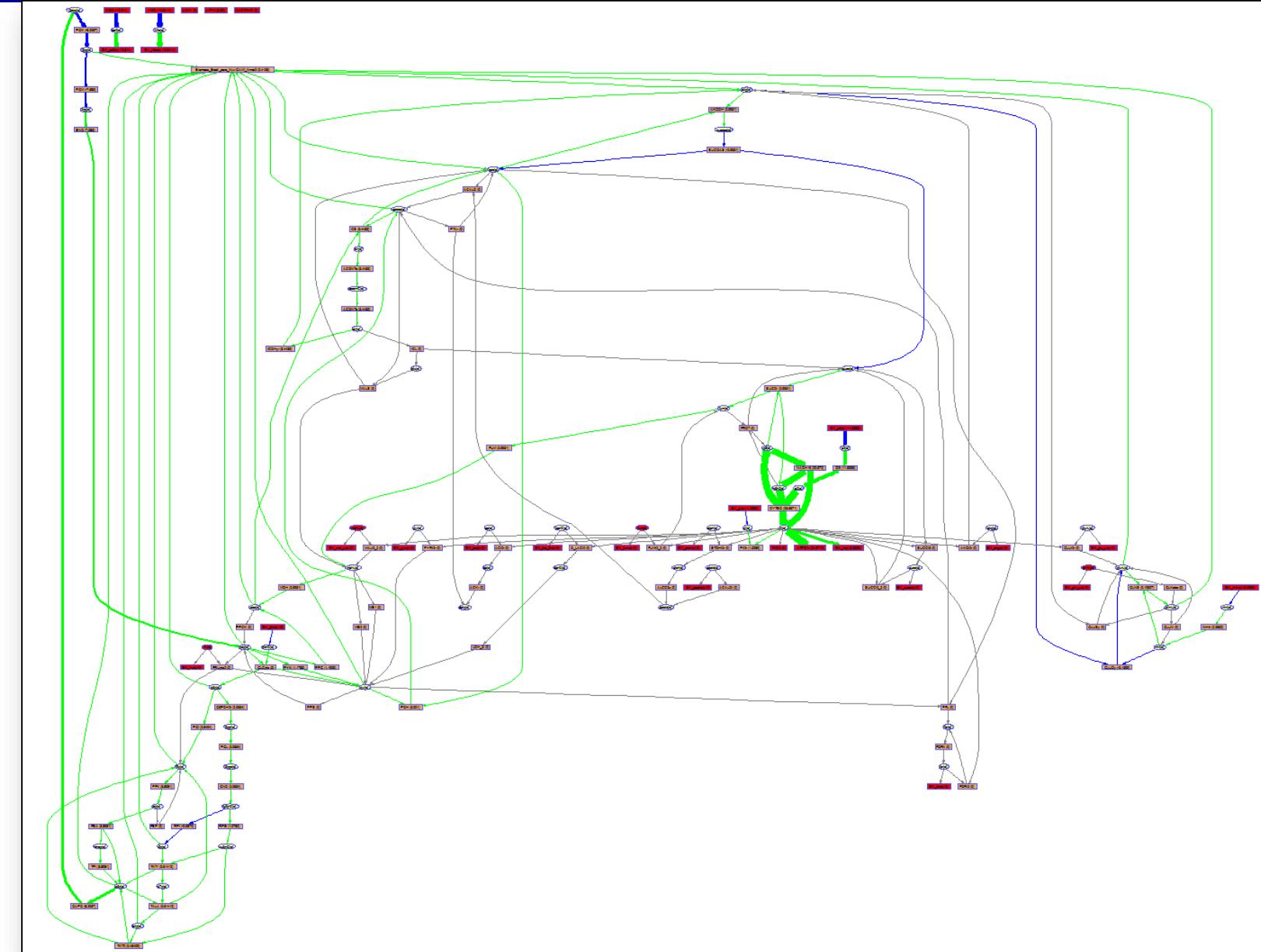


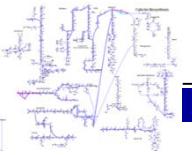
## Textbook Model With Cofactors





## Textbook Model Without Cofactors





# Plotting Active Reactions in Core Model

VisualizeFlux\_core.m

```
clear;

load('ecoli_textbook.mat');

model = changeRxnBounds(model,'EX_glc(e)',-5,'l');
model = changeRxnBounds(model,'EX_o2(e)',-20,'l');
model = changeObjective(model,'Biomass_Ecoli_core_N(w/GAM)_Nmet2');

FBAsolution = optimizeCbModel(model,'max',0,0); % Add two zeros to prevent loops

rxnID = findRxnIDs(model,'Biomass_Ecoli_core_N(w/GAM)_Nmet2'); % Find reaction ID for Biomass Reaction

m = 0;

[n,nLab] = size(model.rxns); % Find the number of reactions in the model

for i=1:n

    if(i~=rxnID) % Remove biomass reaction

        if(FBAsolution.x(i) ~= 0) % Find reactions that are nonzero

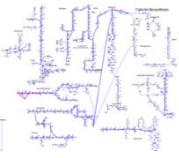
            m = m+1;

            fluxReactions(m) = model.rxns(i); % Put nonzero reaction IDs in fluxReaction vector

        end
    end
end

cofactors = {'amp[c]', 'atp[c]', 'adp[c]', 'pi[c]', 'nad[c]', 'nadh[c]', 'nadph[c]', 'nadp[c]', 'h[c]', 'h2o[c]', 'co2[c]'}

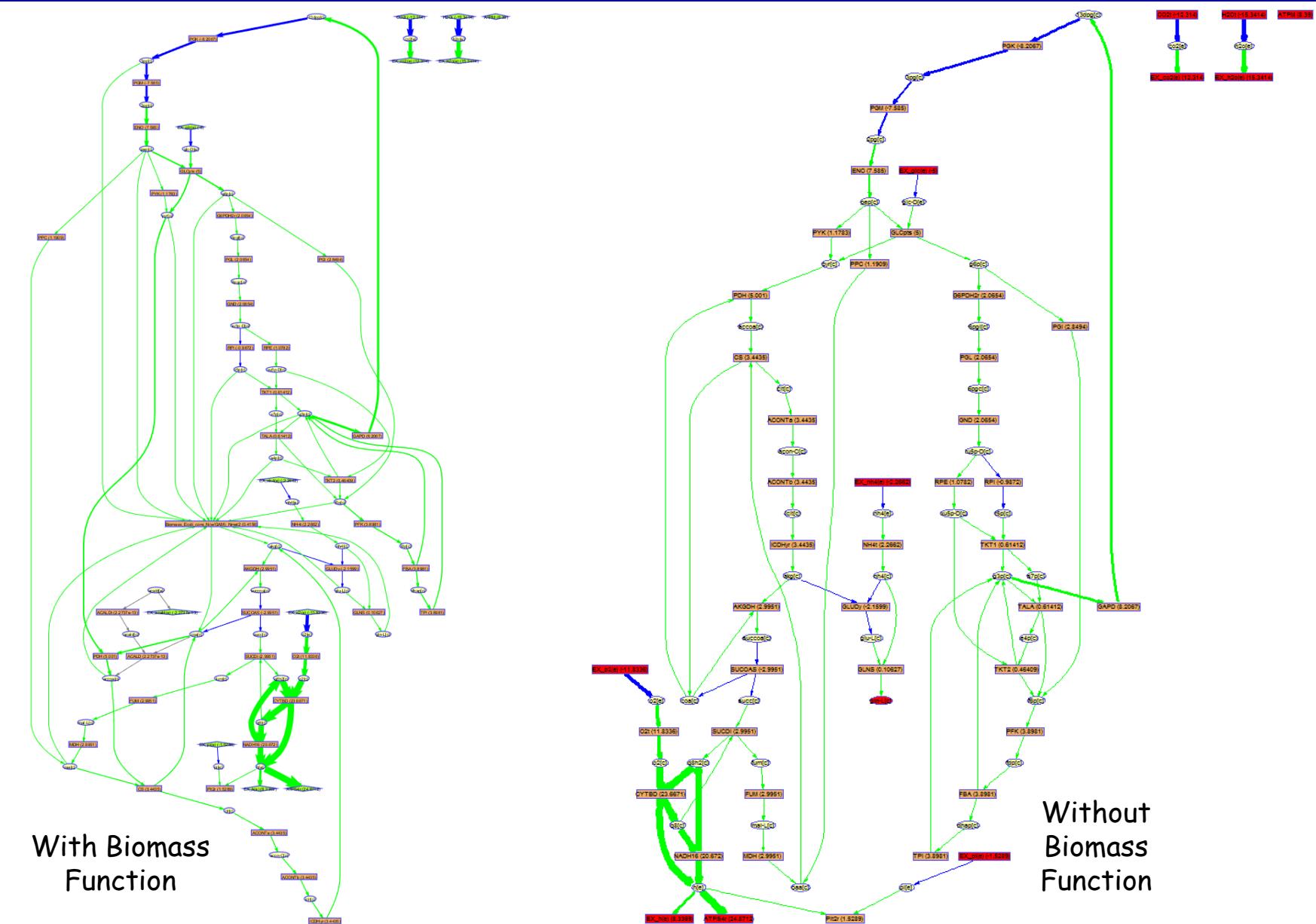
[involvedMets,deadEnds]= draw_by_rxn (model,fluxReactions,'true','struc',{''},cofactors,FBAsolution.x);
```



# Plot of the Active Reactions in Core Model

`VisualizeFlux_core.m`

`VisualizeFlux_Active_core.m`





# Model Subsystems (Reactions)

A	B		C	
1	abbreviation	officialName	equation	subSystem
2	ACALD	acetaldehyde dehydrogenase (acetylating)	[c] : acald + coa + nad <==> accoa + h + nadh	Pyruvate Metabolism
3	ACALDt	acetaldehyde reversible transport	acald[e] <==> acald[c]	Transport, Extracellular
4	ACKr	acetate kinase	[c] : ac + atp <==> actp + adp	Pyruvate Metabolism
5	ACONTa	aconitase (half-reaction A, Citrate hydro-lyase)	[c] : cit <==> acon-C + h2o	Citric Acid Cycle
6	ACONTb	aconitase (half-reaction B, Isocitrate hydro-lyase)	[c] : acon-C + h2o <==> icit	Citric Acid Cycle
7	Act2r	acetate reversible transport via proton symport	ac[e] + h[e] <==> ac[c] + h[c]	Transport, Extracellular
8	ADK1	adenylate kinase	[c] : amp + atp <==> (2) adp	Oxidative Phosphorylation
9	AKGDH	2-Oxoglutarate dehydrogenase	[c] : akg + coa + nad --> co2 + nadh + succoa	Citric Acid Cycle
10	AKGt2r	2-oxoglutarate reversible transport via symport	akg[e] + h[e] <==> akg[c] + h[c]	Transport, Extracellular
11	ALCD2x	alcohol dehydrogenase (ethanol)	[c] : etoh + nad <==> acald + h + nadh	Pyruvate Metabolism
12	ATPM	ATP maintenance requirement	[c] : atp + h2o --> adp + h + pi	Oxidative Phosphorylation
13	ATPS4r	ATP synthase (four protons for one ATP)	adp[c] + (4) h[e] + pi[c] <==> atp[c] + (3) h[c] + h2o[c]	Oxidative Phosphorylation
14	Biomass_Ecoli	Biomass Objective Function with GAM	[c] : (1.496) 3pg + (3.7478) accoa + (59.8100) atp + (0.3610) e4p + (0.0709) f6p + (0.1290) g:	
15	CO2t	CO2 transporter via diffusion	co2[e] <==> co2[c]	Transport, Extracellular
16	CS	citrate synthase	[c] : accoa + h2o + oaa --> cit + coa + h	Citric Acid Cycle
17	CYTBD	cytochrome oxidase bd (ubiquinol-8: 2 protons)	(2) h[c] + (0.5) o2[c] + q8h2[c] --> (2) h[e] + h2o[c] + q8	Oxidative Phosphorylation
18	D_LACT2	D-lactate transport via proton symport	h[e] + lac-D[e] <==> h[c] + lac-D[c]	Transport, Extracellular
19	ENO	enolase	[c] : 2pg <==> h2o + pep	Glycolysis/Gluconeogenesis
20	ETOHt2r	ethanol reversible transport via proton symport	etoh[e] + h[e] <==> etoh[c] + h[c]	Transport, Extracellular
21	EX_ac(e)	Acetate exchange	[e] : ac <==>	Exchange
22	EX_acald(e)	Acetaldehyde exchange	[e] : acald <==>	Exchange
23	EX_akg(e)	2-Oxoglutarate exchange	[e] : akg <==>	Exchange
24	EX_co2(e)	CO2 exchange	[e] : co2 <==>	Exchange
25	EX_etoh(e)	Ethanol exchange	[e] : etoh <==>	Exchange
26	EX_for(e)	Formate exchange	[e] : for <==>	Exchange
27	EX_fru(e)	D-Fructose exchange	[e] : fru <==>	Exchange



# Plotting Model Subsystems

Paint4Net\_core\_subSystem.m

```
clear;

load('ecoli_textbook.mat');

model = changeRxnBounds(model,'EX_glc(e)',-5,'l');
model = changeRxnBounds(model,'EX_o2(e)',-20,'l');
model = changeObjective(model,'Biomass_Ecoli_core_N(w/GAM)_Nmet2');

FBAsolution = optimizeCbModel(model,'max');

% Identify cofactors

cofactors = {'amp[c]','atp[c]','adp[c]','pi[c]','nad[c]','nadh[c]','nadph[c]','nadp[c]','h[c]','h2o[c]','co2[c]'

%Extract & plot single subsystem reactions (Citric Acid Cycle)

fluxReactions = model.rxn(ismember(model.subSystems,'Citric Acid Cycle'));

[involvedMets,deadEnds]= draw_by_rxn (model,fluxReactions,'true','struc',{''},{''},FBAsolution.x);

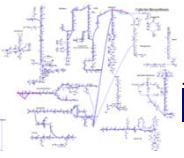
% Extract multiple subsystem reactions

includedSubSystems = {'Citric Acid Cycle','Pyruvate Metabolism','Oxidative Phosphorylation',...
    'Glycolysis/Gluconeogenesis','Pentose Phosphate Pathway','Glutamate Metabolism'};

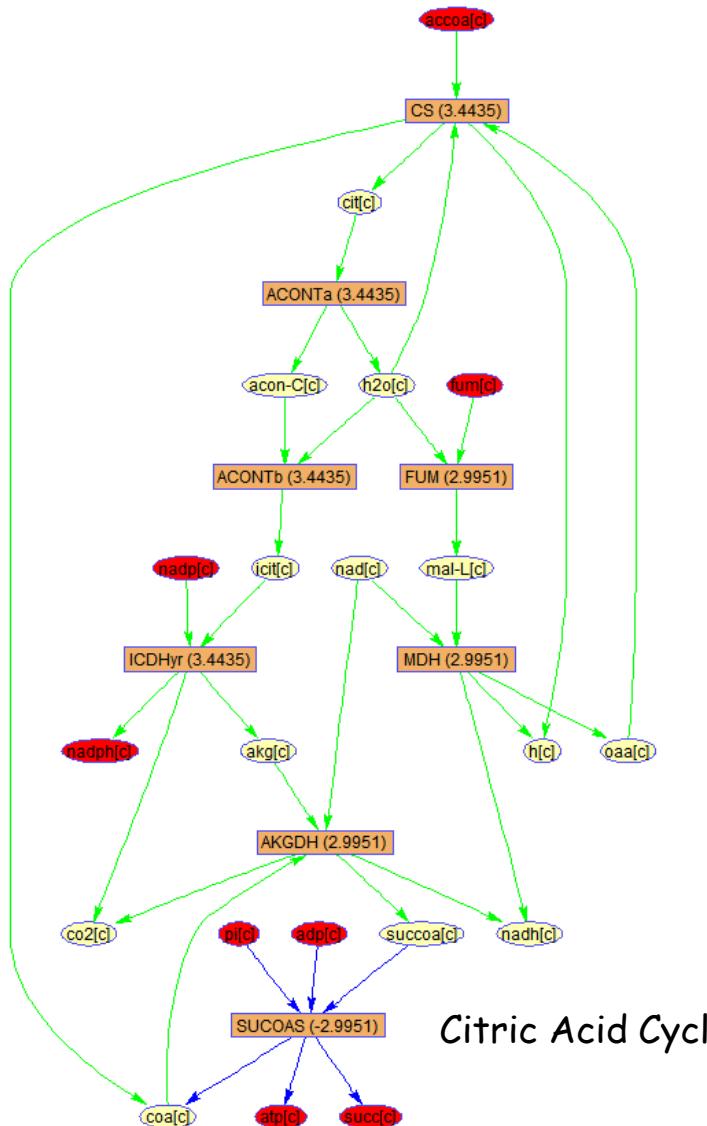
fluxReactions = model.rxn(ismember(model.subSystems,includedSubSystems));

% Plot multiple subsystem

[involvedMets,deadEnds]= draw_by_rxn (model,fluxReactions,'true','struc',{''},cofactors,FBAsolution.x);
```

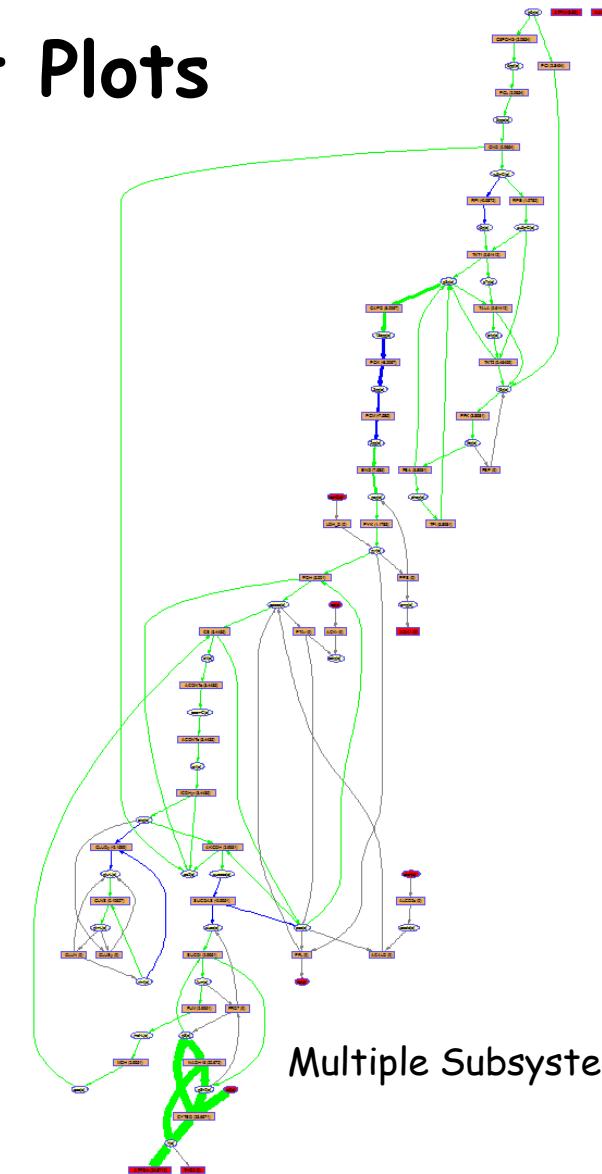


# Paint4Net Plots

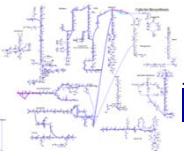


Citric Acid Cycle

Paint4Net\_core\_subSystem.m

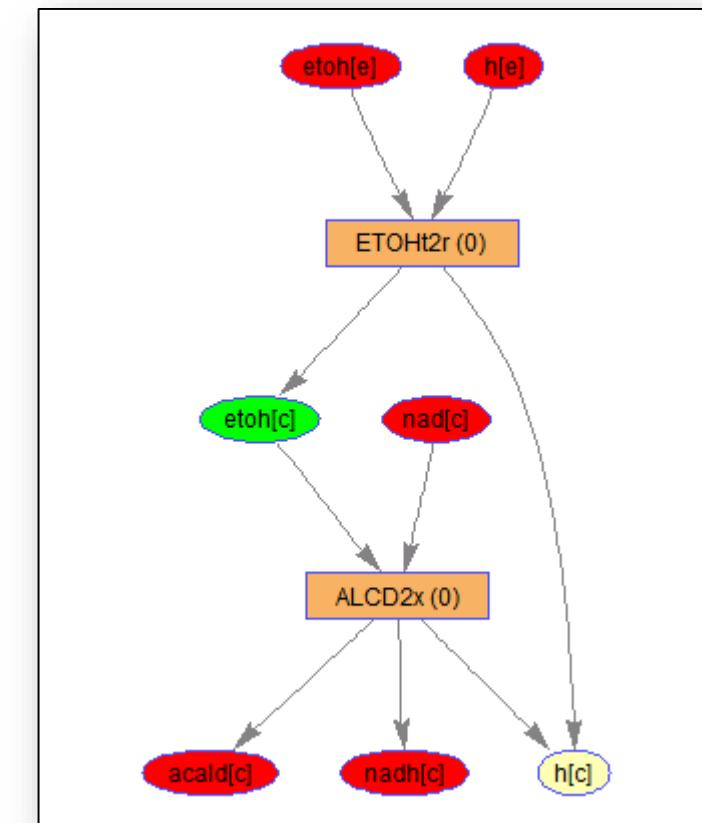


Multiple Subsystems



# draw\_by\_met

- Rectangles represent reactions;
- Numbers in rectangles represent flux rate through reaction.
- Red rectangles represent reactions with only one input or output flux (signaling a potential dead reaction);
- Ellipses represent metabolites;
- Red ellipses represent dead end metabolites;
- Grey edges represent zero-rate fluxes;
- Green edges represent positive-rate (forward) fluxes;
- Blue edges represent negative-rate (backward) fluxes.
- The thickness of the edges is calculated as percentage assuming the maximum rate of flux in the model corresponds to 100%.





# Plotting 'etoh[c]' Connectivity

VisualizeMets\_Textbook.m

```
clear;

load('ecoli_textbook.mat');

model = changeRxnBounds(model,'EX_glc(e)',-5,'l');

model = changeRxnBounds(model,'EX_o2(e)',-20,'l');

model = changeObjective(model,'Biomass_Ecoli_core_N(w/GAM)_Nmet2');

FBAsolution = optimizeCbModel(model,'max',0,0);

cofactors = {'amp[c]','atp[c]','adp[c]','pi[c]','nad[c]','nadh[c]','nadph[c]','nadp[c]','h[c]','h2o[c]','co2[c]'

% Plot connectivity to 'etoh[c]', include cofactors. Radius = 1

[invovledRxns,involvedMets,deadEnds]= draw_by_met (model,{['etoh[c]']},'true',1,'struc',{''},FBAsolution.x);

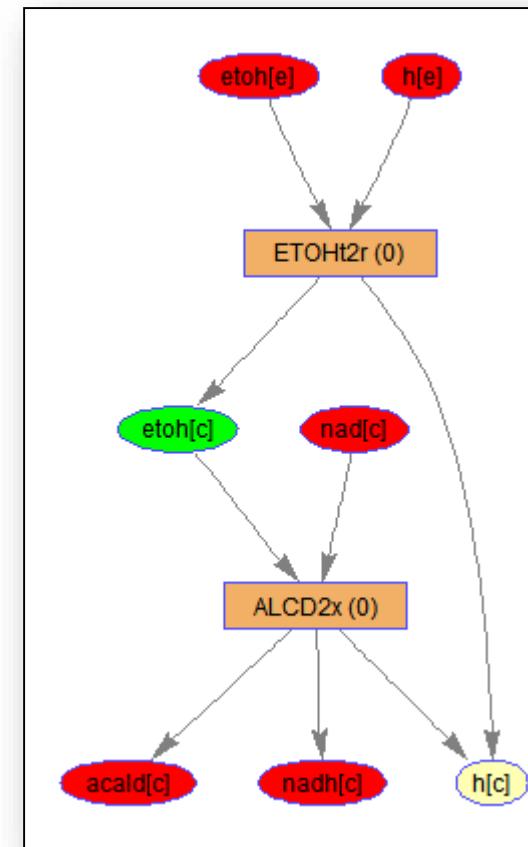
% Plot connectivity to 'etoh[c]', remove cofactors. Radius = 1

[invovledRxns,involvedMets,deadEnds]= draw_by_met (model,{['etoh[c]']},'true',1,'struc',cofactors,FBAsolution.x);
```

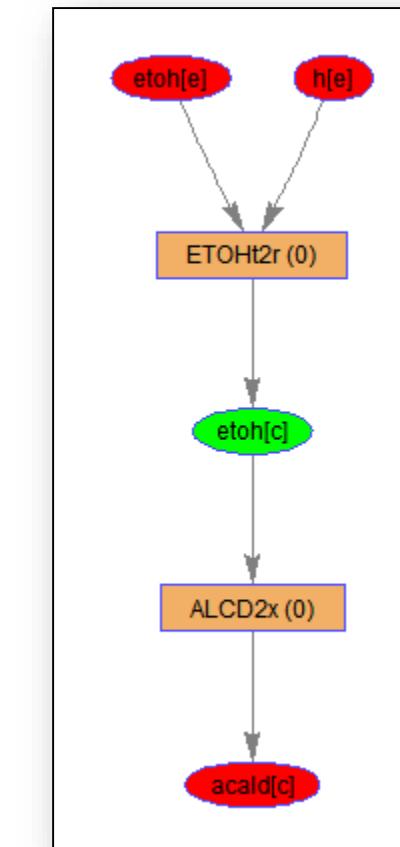


# 'etoh[c]' Connectivity (Radius = 1)

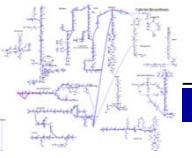
VisualizeMets\_Textbook.m



With Cofactors

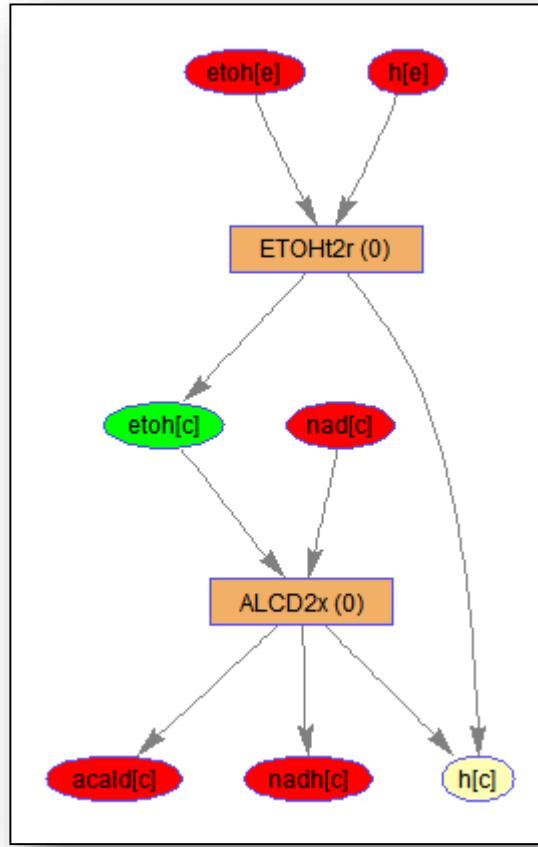


Without Cofactors

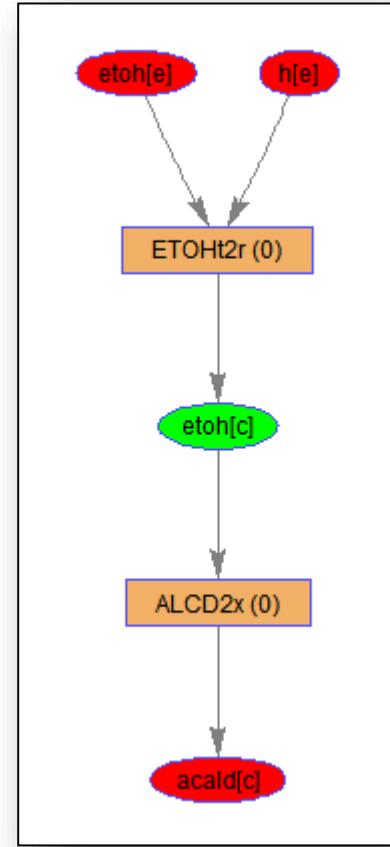


## 'etoh[c]' Connectivity (Radius = 1)

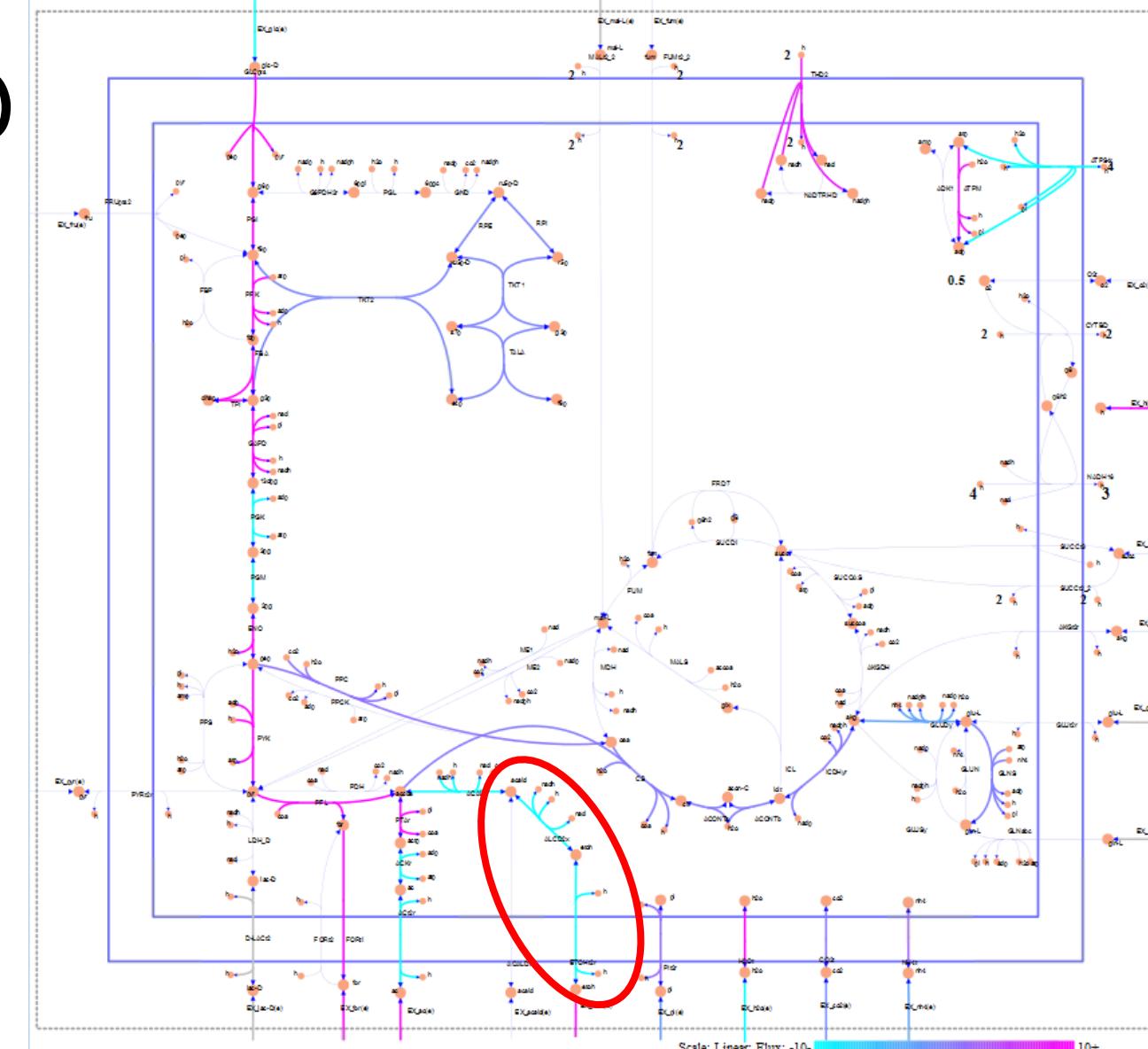
VisualizeMets\_Textbook.m

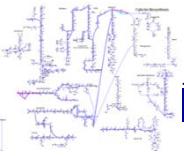


With Cofactors



Without Cofactors





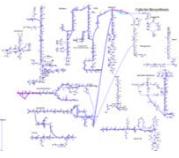
# Cobra Toolbox

- Cobra Toolbox Overview
- Cobra Toolbox Fundamentals
- Cobra Examples
- BIGG Database
- Paint4Net



# Cobra Toolbox & Matlab Functions Discussed in this Lecture

```
• load('ecoli_textbook.mat'); % Load textbook model  
• model = changeRxnBounds(model,'EX_glc(e)',-18.5,'l'); % Set lower bound of glucose  
• model = changeRxnBounds(model,'EX_o2(e)',-1000,'l'); % Set lower bound of oxygen  
• model = changeObjective(model,'Biomass_Ecoli_core_N(w/GAM)_Nmet2'); % Set objective function (Biomass)  
• FBAsolution = optimizeCbModel(model,'max',0,0); % Find optimized flux values  
• map=readCbMap('ecoli_Textbook_ExportMap'); % Input ecoli textbook map template  
• drawFlux(map, model, FBAsolution.x, options); % Draw Map  
• printFluxVector(model, FBAsolution.x, true); % Print flux values  
• [involvedMets,deadEnds]= draw_by_rxn (model,model.rxns,'true','struc',{''},cofactors,FBAsolution.x);  
• rxnID = findRxnIDs(model,'Biomass_Ecoli_core_N(w/GAM)_Nmet2'); % Find reaction ID for Biomass Reaction  
• [n,nLab] = size(model.rxns); % Find the number of reactions in the model  
• [invovledRxns,involvedMets,deadEnds]= draw_by_met (model,{['etoh[c]']},'true',1,'struc',cofactors,FBAsolution.x);  
• printFluxVector(model, solution.x, true) % Print reactions and their FBA solutions  
• fluxReactions = model.rxns(ismember(model.subSystems,'Citric Acid Cycle'))); % ismember find rxnsID's
```



# Reflective Questions

- What is the openCobra Project?
- What is the relationship between Matlab and the Cobra Toolbox?
- Where can you find the documentation for all the Cobra Toolbox functions?
- What is the start-up process that must be followed to use the Cobra Toolbox?
- What is the difference between the Cobra Toolbox and a Cobra model?
- How is a Cobra model stored in the Cobra Matlab interface?
- What Matlab function is used to load a Cobra model?
- What location in a Cobra model stores the reactions?
- What location in a Cobra model stores the metabolites?
- What location in a Cobra model stores the stoichiometric matrix?
- What location in a Cobra model stores the objective function?
- What Cobra Toolbox function is used to calculate the network fluxes?
- What location in a optimized solution struct stores the flux vector?
- What location in a optimized solution struct stores the objective function value?
- How can you calculate the cell doubling time?
- What Cobra Toolbox function is used to read an export map?
- What process must be followed to print a Cobra model map using the Cobra Toolbox?
- What Cobra Toolbox function is used to write fluxes onto an map?
- What is the default file name for maps created by the Cobra Toolbox?
- What Cobra Toolbox function is used to change the upper and lower bounds of a reaction?
- What Cobra Toolbox function is define the objective function?
- Explain the capabilities of the printFluxVector command?
- What is the difference between the BIGG Database and Paint4Net?
- What are the advantages of the "draw\_by\_rxn" function in Paint4Net?
- What are the advantages of the "draw\_by\_met" function in Paint4Net?
- What does connectivity mean in the Paint4Net function?