



# The COBRA Toolbox v.3.0



# Learning Objectives

Each student should be able to:

- Explain the purpose of the COBRA Toolbox v.3.0,
- Understand the organization of the COBRA Toolbox v.3.0 website,
- Demonstrate basic operation of the COBRA Toolbox v.3.0.



# COBRA Toolbox v.3.0

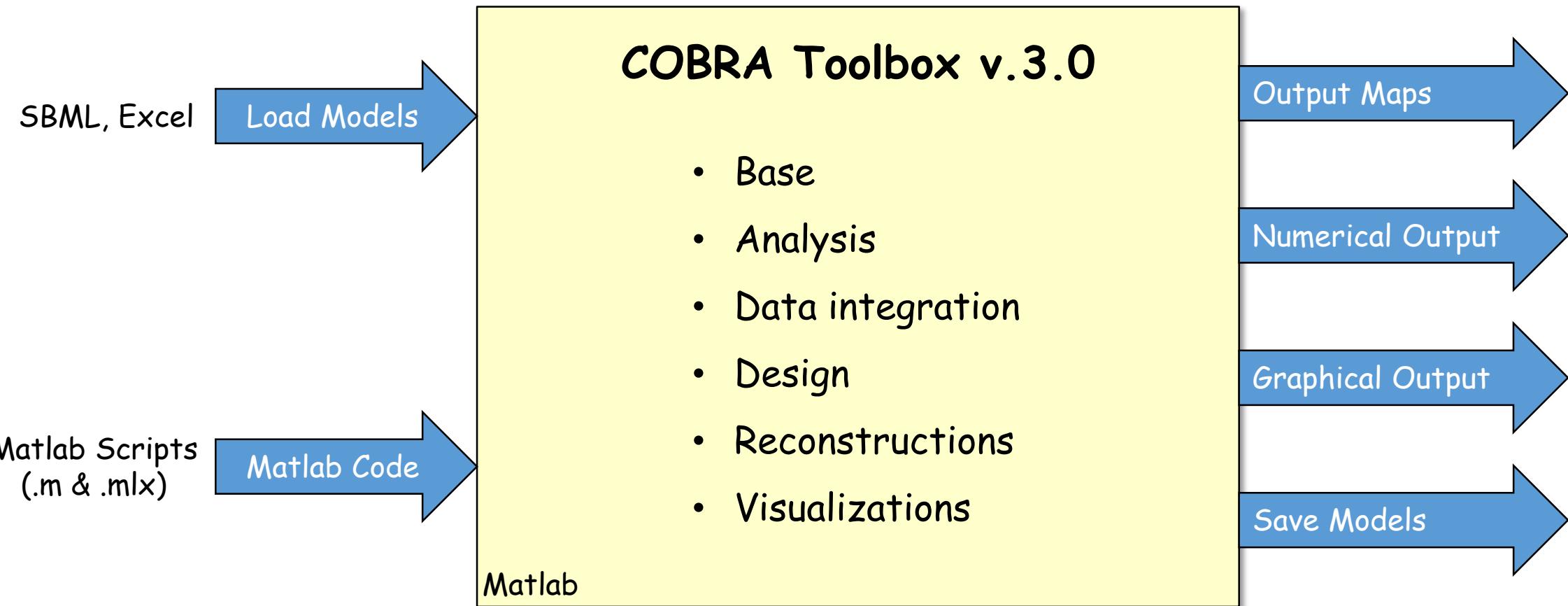
- COBRA Toolbox v.3.0 Overview
- Base Functions
- Analysis Functions
- Data Integration Functions
- Design Functions
- Reconstruction Functions
- Visualization Functions
- COBRA Toolbox Examples



<https://opencobra.github.io/cobratoolbox/stable/index.html>



# COBRA Toolbox v.3.0 Overview



Laurent Heirendt et al, Creation and analysis of biochemical constraint-based models: the COBRA Toolbox v3.0, Nature Protocols, volume 14, pages 639-702, 2019



# COBRA Toolbox v.3.0 Paper

Laurent Heirendt *et al*, Creation and analysis of biochemical constraint-based models: the COBRA Toolbox v3.0, Nature Protocols, volume 14, pages 639-702, 2019

<https://www.nature.com/articles/s41596-018-0098-2>

nature  
protocols

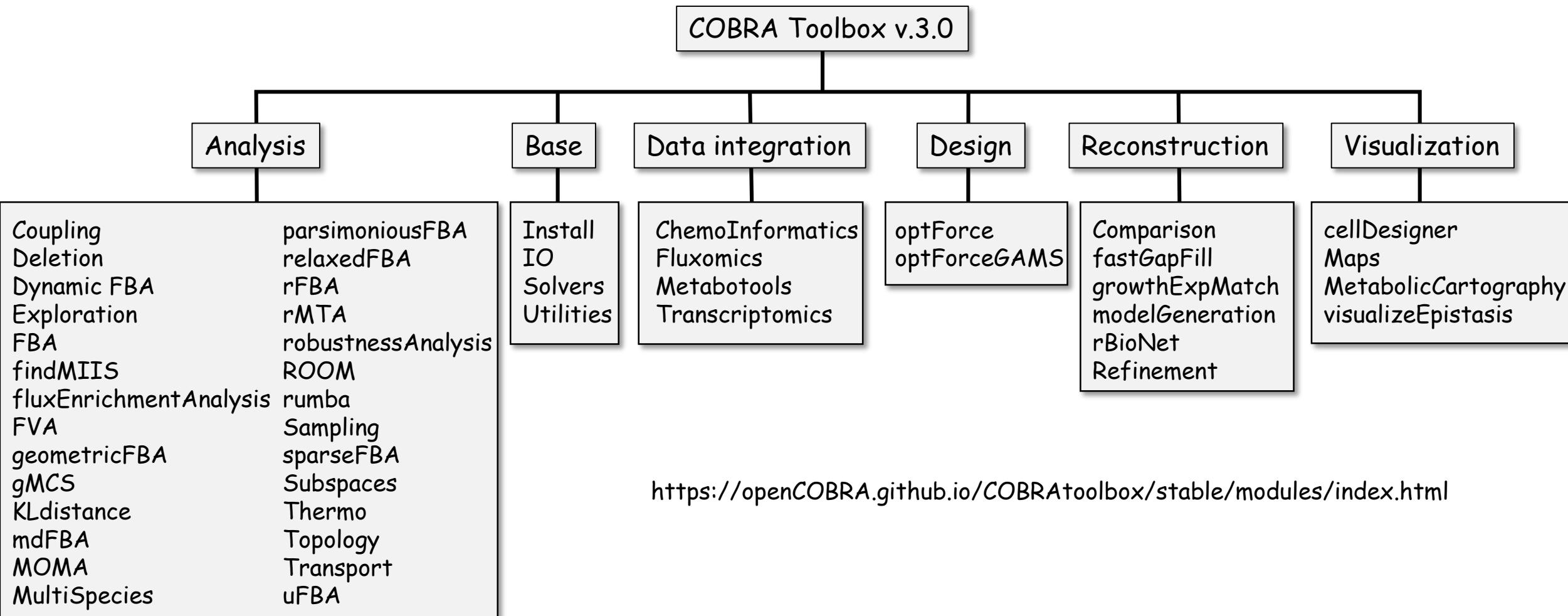
PROTOCOL UPDATE

<https://doi.org/10.1038/s41596-018-0098-2>

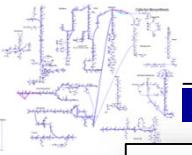
## Creation and analysis of biochemical constraint-based models using the COBRA Toolbox v.3.0

Laurent Heirendt<sup>1,24</sup>, Sylvain Arreckx<sup>1,24</sup>, Thomas Pfau<sup>1,2</sup>, Sebastián N. Mendoza<sup>1,3,4</sup>, Anne Richelle<sup>5</sup>, Almut Heinken<sup>1</sup>, Hulda S. Haraldsdóttir<sup>1</sup>, Jacek Wachowiak<sup>1</sup>, Sarah M. Keating<sup>6</sup>, Vanja Vlasov<sup>1</sup>, Stefania Magnusdóttir<sup>1</sup>, Chiam Yu Ng<sup>7</sup>, German Preciat<sup>1</sup>, Alise Žagare<sup>1</sup>, Siu H. J. Chan<sup>7</sup>, Maike K. Aurich<sup>1</sup>, Catherine M. Clancy<sup>1</sup>, Jennifer Modamio<sup>1</sup>, John T. Sauls<sup>8</sup>, Alberto Noronha<sup>1</sup>, Aarash Bordbar<sup>9</sup>, Benjamin Cousins<sup>10</sup>, Diana C. El Assal<sup>1</sup>, Luis V. Valcarcel<sup>1,11</sup>, Iñigo Apaolaza<sup>1,11</sup>, Susan Ghaderi<sup>1</sup>, Masoud Ahookhosh<sup>1</sup>, Marouen Ben Guebila<sup>1</sup>, Andrejs Kostromins<sup>12</sup>, Nicolas Sompairac<sup>13</sup>, Hoai M. Le<sup>1</sup>, Ding Ma<sup>14</sup>, Yuekai Sun<sup>15</sup>, Lin Wang<sup>7</sup>, James T. Yurkovich<sup>1,16</sup>, Miguel A. P. Oliveira<sup>1</sup>, Phan T. Vuong<sup>1</sup>, Lemmer P. El Assal<sup>1</sup>, Inna Kuperstein<sup>1,13</sup>, Andrei Zinovyev<sup>13</sup>, H. Scott Hinton<sup>17</sup>, William A. Bryant<sup>18</sup>, Francisco J. Aragón Artacho<sup>19</sup>, Francisco J. Planes<sup>11</sup>, Egils Stalidzans<sup>12</sup>, Alejandro Maass<sup>3,4</sup>, Santosh Vempala<sup>10</sup>, Michael Hucka<sup>20</sup>, Michael A. Saunders<sup>14</sup>, Costas D. Maranas<sup>7</sup>, Nathan E. Lewis<sup>1,21</sup>, Thomas Sauter<sup>2</sup>, Bernhard Ø. Palsson<sup>16,22</sup>, Ines Thiele<sup>1</sup> and Ronan M. T. Fleming<sup>1,23\*</sup>

Constraint-based reconstruction and analysis (COBRA) provides a molecular mechanistic framework for integrative analysis of experimental molecular systems biology data and quantitative prediction of physicochemically and biochemically feasible phenotypic states. The COBRA Toolbox is a comprehensive desktop software suite of interoperable COBRA methods. It has found widespread application in biology, biomedicine, and biotechnology because its functions can be flexibly combined to implement tailored COBRA protocols for any biochemical network. This protocol is an update to the COBRA Toolbox v.1.0 and v.2.0. Version 3.0 includes new methods for quality-controlled reconstruction, modeling, topological analysis, strain and experimental design, and network visualization, as well as network integration of chemoinformatic, metabolomic, transcriptomic, proteomic, and thermochemical data. New multi-lingual code integration also enables an expansion in COBRA application scope via high-precision, high-performance, and nonlinear numerical optimization solvers for multi-scale, multi-cellular, and reaction kinetic modeling, respectively. This protocol provides an overview of all these new features and can be adapted to generate and analyze constraint-based models in a wide variety of scenarios. The COBRA Toolbox v.3.0 provides an unparalleled depth of COBRA methods.



Laurent Heirendt et al, Creation and analysis of biochemical constraint-based models: the COBRA Toolbox v3.0, Nature Protocols, volume 14, pages 639-702, 2019



The screenshot shows the COBRA Toolbox documentation page. At the top is a large logo consisting of a stylized 'S' with a small MATLAB-like icon inside it. Below the logo is the title "The COBRA Toolbox". A search bar labeled "Search docs" is present. A vertical sidebar on the left contains links: Home, Installation (highlighted with a red arrow), Functions, Tutorials, How to contribute, How to cite, Support (highlighted with a red arrow), FAQ, Contributors, and Funding.

- Search docs
- Home
- Installation
- Functions
- Tutorials
- How to contribute
- How to cite
- Support
- FAQ
- Contributors
- Funding

# COBRA Toolbox v.3.0 Website

<https://openCOBRA.github.io/COBRAtoolbox/>

The screenshot shows the GitHub repository landing page for "The COBRA Toolbox". The title "The COBRA Toolbox" is displayed prominently. Below the title is a button with the text "View The COBRA Toolbox source code on GitHub". At the bottom of the page are four circular icons with text labels: "Installation" (red), "Tutorials" (blue), "Contributing" (orange), and "Documentation" (green).

The COBRA Toolbox

View The COBRA Toolbox source code on GitHub .

- Installation
- Tutorials
- Contributing
- Documentation



# COBRA Toolbox v.3.0 Installation

The COBRA Toolbox website features a search bar and a navigation menu. The 'Installation' section is expanded, showing sub-options: System Requirements, Solver compatibility, Binaries and Compilers, Solver installation, and Test the installation.

- System Requirements
  - Download and installation
- Solver compatibility
- Binaries and Compilers
- Solver installation
  - Test the installation

<https://openCOBRA.github.io/COBRAtoolbox/stable/installation.html#>

## Windows

Please download the [git](#) tools for Windows from [here](#). During the installation process, please ensure that you select **Use Git Bash** and **optional Unix tools** from the Windows Command prompt. In addition, please make sure that you select **Checkout as-is, commit Unix-style line endings**.

The screenshots show two configuration steps for Git 2.13.1.2 on Windows.

**Adjusting your PATH environment:** This window asks how to use Git from the command line. Three options are shown:

- Use Git from Git Bash only: This is the safest choice as your PATH will not be modified at all. You will only be able to use the Git command line tools from Git Bash.
- Use Git from the Windows Command Prompt: This option is considered safe as it only adds some minimal Git wrappers to your PATH to avoid cluttering your environment with optional Unix tools. You will be able to use Git from both Git Bash and the Windows Command Prompt.
- Use Git and optional Unix tools from the Windows Command Prompt: Both Git and the optional Unix tools will be added to your PATH. A warning message states: "Warning: This will override Windows tools like 'find' and 'sort'. Only use this option if you understand the implications."

**Configuring the line ending conversions:** This window asks how Git should treat line endings in text files. Two options are shown:

- Checkout Windows-style, commit Unix-style line endings: Git will convert LF to CRLF when checking out text files. When committing text files, CRLF will be converted to LF. For cross-platform projects, this is the recommended setting on Windows ("core.autocrlf" is set to "true").
- Checkout as-is, commit Unix-style line endings: Git will not perform any conversion when checking out text files. When



# COBRA

## Toolbox v.3.0

### Functions

#### FBA

findMIs

fluxEnrichmentAnalysis

#### FVA

geometricFVA

gMCS

KLdistance

mdFBA

MOMA

MultiSpecies

parsimoniousFBA

relaxedFBA

rFBA

rMTA

robustnessAnalysis

ROOM

rumba

Sampling

[Docs](#) » [Functions](#) » [Analysis](#) » FBA

<https://openCOBRA.github.io/COBRAtoolbox/stable/index.html>

## FBA

**changeObjective**(*model*, *rxnNameList*, *objectiveCoeff*) [\[source\]](#)

Changes the objective function of a constraint-based model

#### Usage

`model = changeObjective(model, rxnNameList, objectiveCoeff)`

#### Inputs

- **model** – COBRA structure
- **rxnNameList** – List of reactions (cell array or string)

#### Optional input

- **objectiveCoeff** – Value of objective coefficient for each reaction (Default = 1)

#### Output

- **model** – COBRA model structure with new objective



# COBRA Toolbox v.3.0 Tutorials



## The COBRA Toolbox

Search docs

- Home
- Installation
- Functions

⊖ Tutorials

- Analysis
- Base
- Data integration
- Design
- Reconstruction
- Visualization

How to contribute

How to cite

Support

Docs » Tutorials

<https://opencobra.github.io/cobratoolbox/stable/tutorials/index.html>

## Tutorials

### Analysis

- A step-by-step guide to parsimonious enzyme usage Flux Balance Analysis - pFBA
- Analyze Steady-State Community COBRA Models
- Atomically resolve a metabolic reconstruction
- Browse Networks in the Matlab Command Window Using surfNet
- Computation and analysis of microbe-microbe metabolic interactions
- Computation and analysis of rescued lethal gene deletions in a host-microbe model
- Creation and simulation of personalized microbiota models through metagenomic data integration
- Determining MinSpan vectors of COBRA model
- Flux Balance Analysis (FBA)
- Flux Variability analysis (FVA)
- genetic Minimal Cut Sets - gMCS
- Proton shuttle testing with sparse flux balance analysis
- Reaction essentiality across multiple models
- Relaxed Flux Balance Analysis: Recon 3
- Relaxed Flux Balance Analysis: Toy model
- Simulation of growth of human gut microbes on different diets
- Sparse Flux Balance Analysis



# COBRA Toolbox v.3.0 Support

The screenshot shows the Google Groups interface for the 'COBRA Toolbox' group. The left sidebar includes links for 'My groups', 'Home', 'Starred', 'Favorites' (with a note to click a star icon), 'Recently viewed' (listing 'cobra pie', 'Gurobi Optimization', 'MASS Toolbox', and 'COBRA Toolbox'), 'Recent searches' (listing 'ismember (in cobr...', 'ecolicoremodel (i...', 'minerva (in cobra...', 'reconmap (in cobr...', 'calculateQuantitat...'), and 'Recently posted to COBRA Toolbox'. The main content area shows a summary of the group: 'Shared publicly', 30 of 1213 topics, and 99+ unread messages. A prominent message encourages users to complete a survey at <https://goo.gl/forms/muwqARkQWytOcC9m1>. Below this, instructions for posting are provided, followed by a list of recent posts:

- Integrated model linking two microbes models (4) - By Y. Fan - 4 posts - 10 views - Jul 24
- Error in initialization of Cobra Toolbox (1) - By wentong yu - 1 post - 8 views - Jul 16
- problem running RHS (1) - By Catarina Ribeiro - 1 post - 4 views - Jul 16
- VerifyModel - Issues with model (10) - By Devika N .T - 10 posts - 13 views - Jul 14
- FastFVA and ibm\_CPLEX solver error (1) - By Bushra Dohai - 1 post - 2 views - Jul 9



# COBRA Toolbox v.3.0

- COBRA Toolbox v.3.0 Overview
- Base Functions
- Analysis Functions
- Data Integration Functions
- Design Functions
- Reconstruction Functions
- Visualization Functions
- COBRA Toolbox Examples

- **Install**
  - ✓ Functions required for the installation process
- **IO**
  - ✓ Input/Output Functions
  - ✓ Connections to the BIGG and KEGG databases
  - ✓ Utilities for working with COBRA models
- **Solvers**
  - ✓ Interfaces to optimization solvers
- **Utilities**
  - ✓ Functions for working with COBRA models



# Constraint-based Metabolic Reconstructions & Analysis

2021 H. Scott Hinton

13

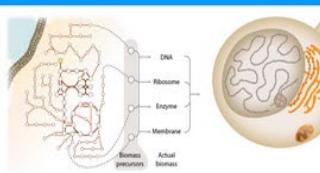
Systems Biology Research Group (<http://bigg.ucsd.edu>) About Advanced Search Web API

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

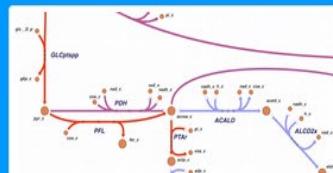
## BiGG Models

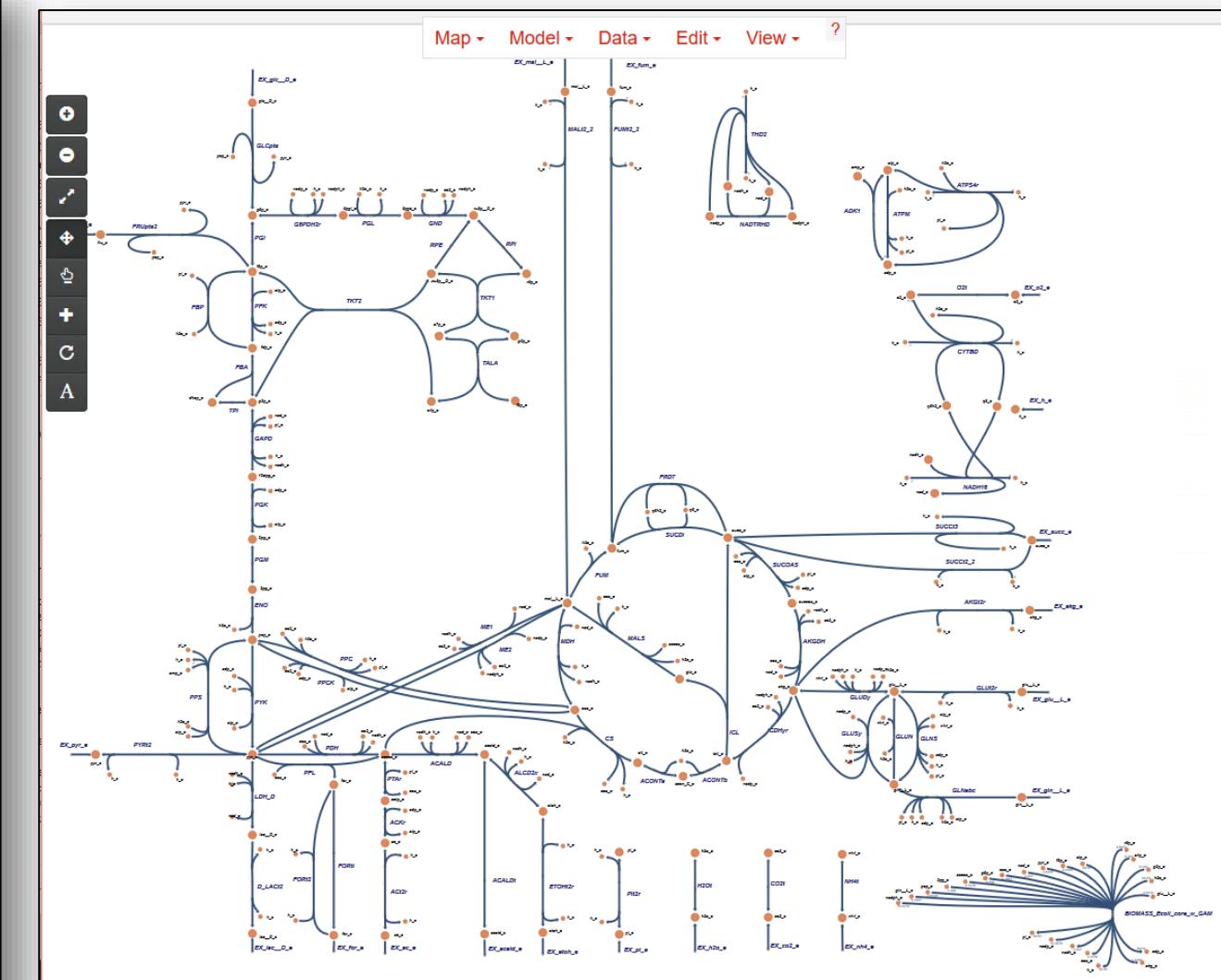
Search the database by model, reaction, metabolite, or gene ?

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



# “Base” Tutorials

- Engaging with The COBRA Toolbox community
- Initialise and verify The COBRA Toolbox
- Input and output of reconstructions and models

<https://opencobra.github.io/cobratoolbox/stable/tutorials/index.html>



## Initialise and verify The COBRA Toolbox

Authors: Sylvain Arreckx, Luxembourg Centre for Systems Biomedicine

Reviewers:

### MATERIALS - EQUIPMENT SETUP

Please ensure that all the required dependencies of The COBRA Toolbox have been properly installed by following the installation guide [here](#). In particular, git and curl must be installed.

### PROCEDURE

At the start of each MATLAB session, The COBRA Toolbox must be initialised. Navigate to the directory where you installed The COBRA Toolbox and initialise

```
initCobraToolbox(false) % false, as we don't want to update
```

The user who primarily uses the official openCOBRA repository may automatically initialise The COBRA Toolbox. To do so, edit the MATLAB startup.m file and add a line with initCobraToolbox so that The COBRA Toolbox is initialised each time that MATLAB is started.

```
if usejava('desktop') % This line of code is to avoid execution in non gui-environments
edit startup.m
end
```

### ANTICIPATED RESULTS

The initialisation step automatically checks the configuration of all of the required and some of the optional software dependencies. During initialisation, all git submodules are updated. The solver paths are set when available and compatible. A system-dependent table with the solver status is returned, together with solver suggestions. The user is also presented with options to update The COBRA Toolbox when necessary.



# COBRA Toolbox 3.0

- COBRA Toolbox v.3.0 Overview
- Base Functions
- **Analysis Functions**
- Data Integration Functions
- Design Functions
- Reconstruction Functions
- Visualization Functions
- COBRA Toolbox Examples

The analysis folder contains all of the methods for interrogation of the properties of a reconstruction or model, and combinations thereof, as well as the prediction of biochemical network states using constraint-based models. Below are some of the analysis types available in the COBRA Toolbox v.3.0.

Coupling	parsimoniousFBA
Deletion	relaxedFBA
Dynamic FBA	rFBA
Exploration	rMTA
FBA	robustnessAnalysis
findMIs	ROOM
fluxEnrichmentAnalysis	rumba
FVA	Sampling
geometricFBA	sparseFBA
gMCS	Subspaces
KLdistance	Thermo
mdFBA	Topology
MOMA	Transport
MultiSpecies	uFBA

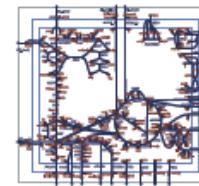
Laurent Heirendt et al, Creation and analysis of biochemical constraint-based models: the COBRA Toolbox v3.0, Nature Protocols, volume 14, pages 639-702, 2019



# Constraint-based Metabolic Reconstructions & Analysis

2021 H. Scott Hinton

a Genome-scale metabolic reconstruction



$A \leftrightarrow B + C$  Reaction 1  
 $B + 2C \rightarrow D$  Reaction 2  
 ...  
 Reaction  $n$

b Mathematically represent metabolic reactions and constraints

Reactions

1	2	...	$n$	Biomass	Glucose	Oxygen
A	-1					
B	1	-1				
C	1	-2				
D			1			
...						
m				-1	-1	

Metabolites

Stoichiometric matrix,  $S$

$\star$

Fluxes,  $v$

$$S \cdot v = 0$$

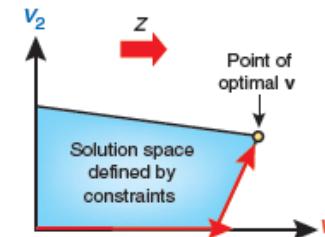
c Mass balance defines a system of linear equations

$$\begin{aligned} -v_1 + \dots &= 0 \\ v_1 - v_2 + \dots &= 0 \\ v_1 - 2v_2 + \dots &= 0 \\ v_2 + \dots &= 0 \\ \text{etc.} & \end{aligned}$$

d Define objective function ( $Z = c_1^* v_1 + c_2^* v_2 \dots$ )

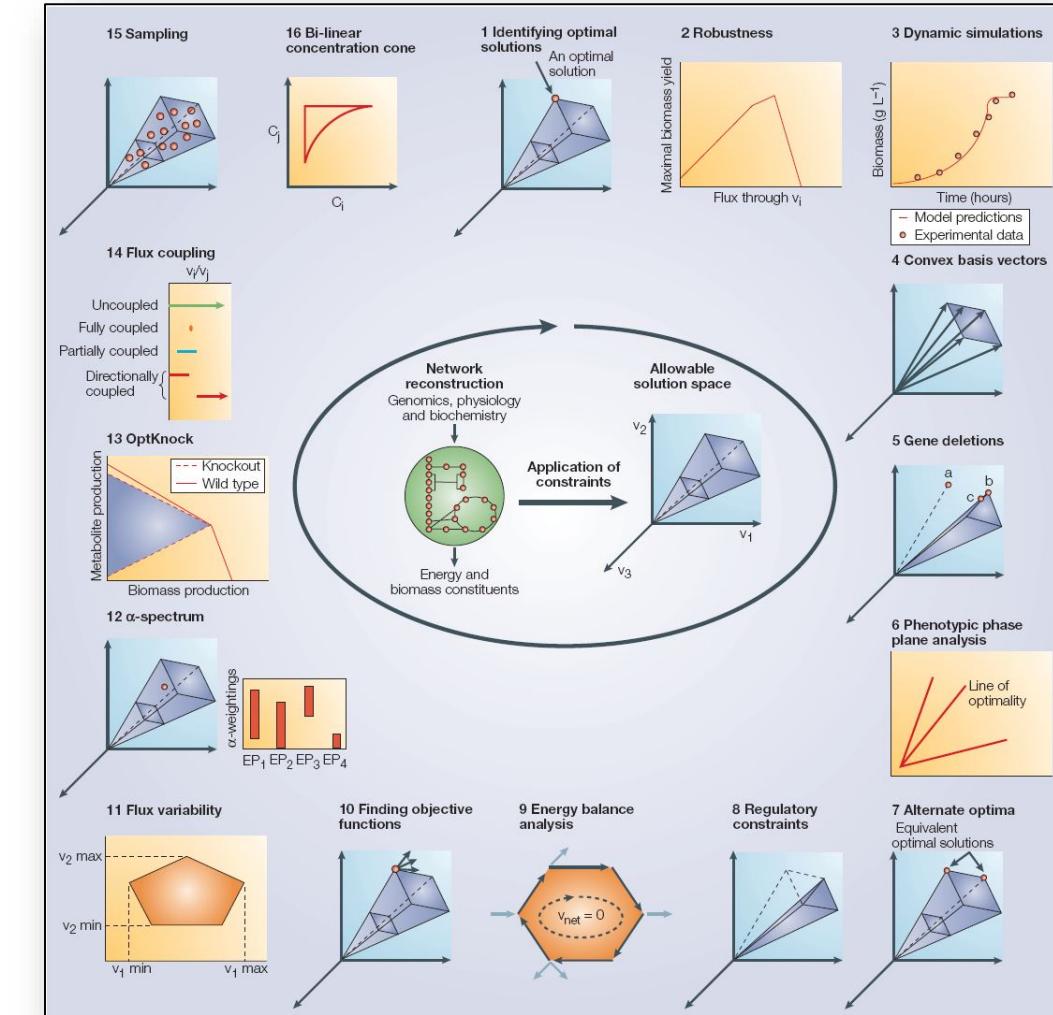
To predict growth,  $Z = v_{\text{biomass}}$

e Calculate fluxes that maximize  $Z$



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

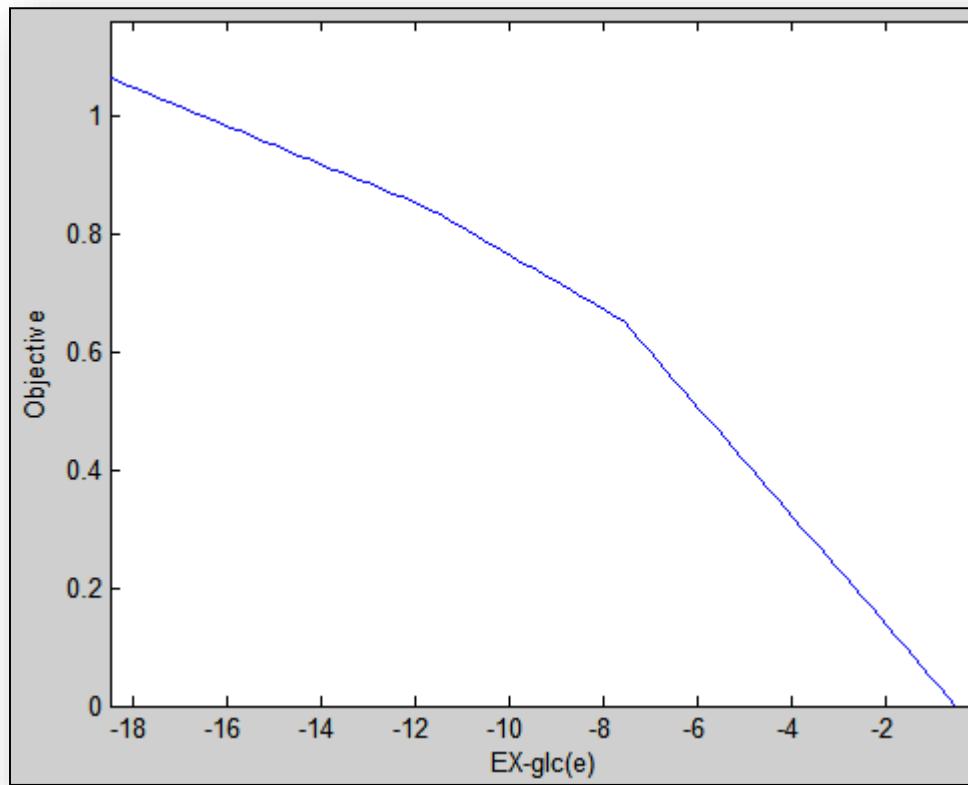
# Flux Balance Analysis



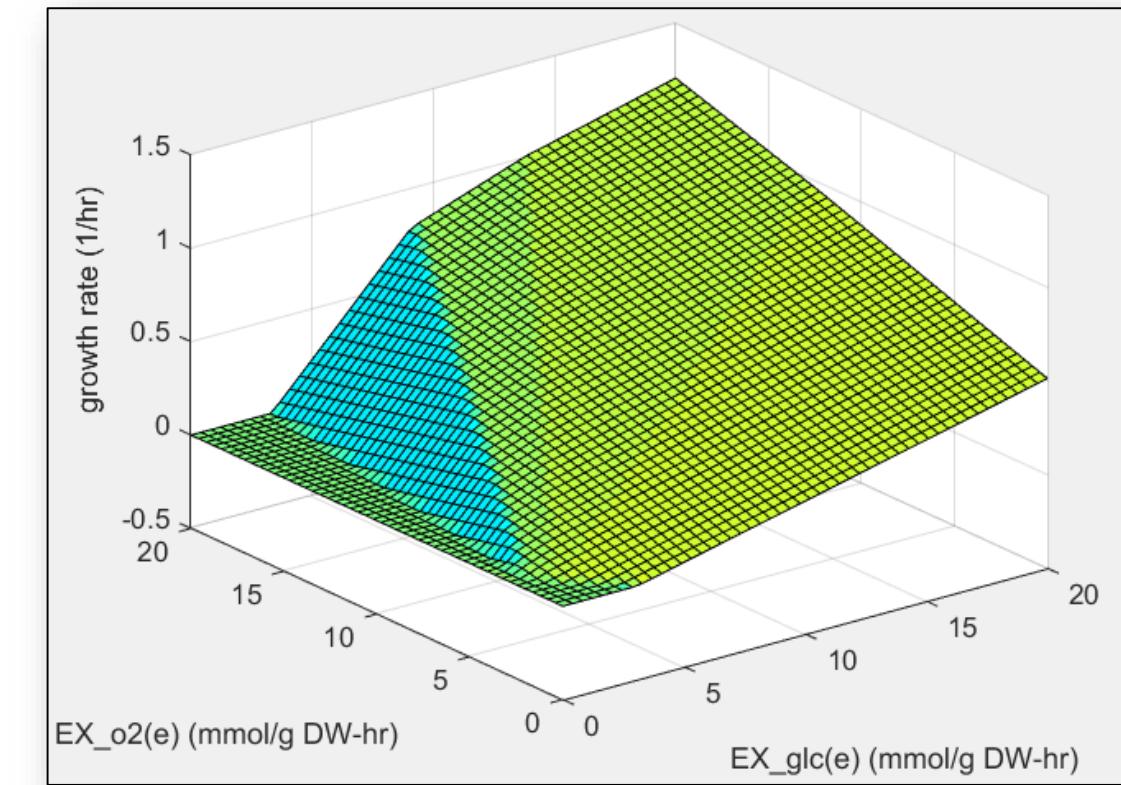
Price, N. D., J. L. Reed, et al. (2004). "Genome-scale models of microbial cells: evaluating the consequences of constraints." *Nature reviews. Microbiology* 2(11): 886-897.



# Robustness Analysis & Phenotype Phase Plane Analysis



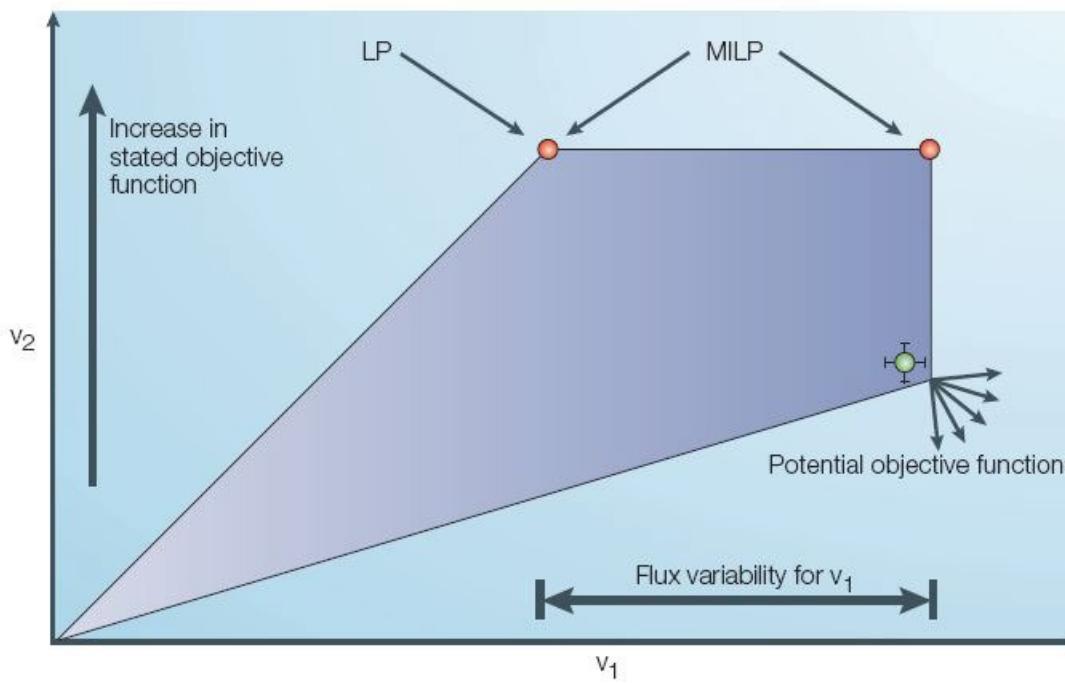
Robustness Analysis



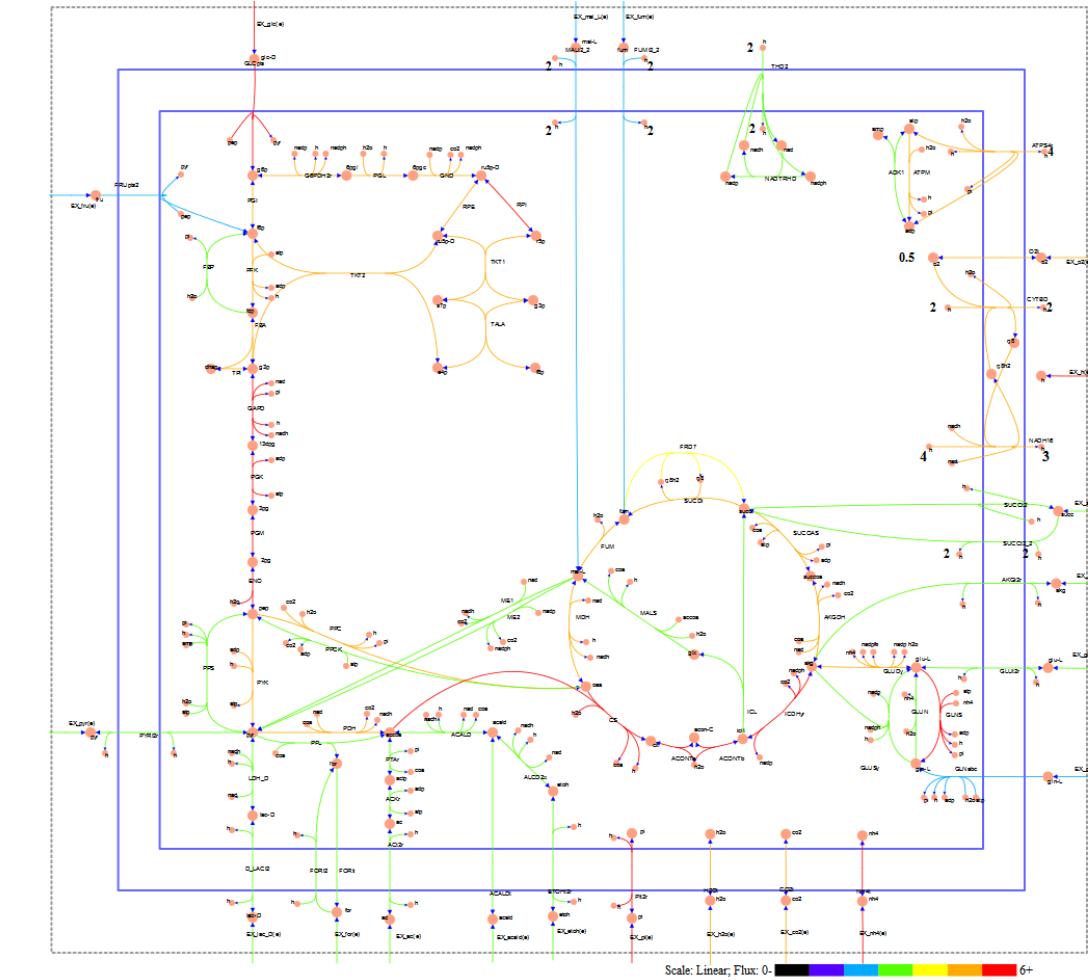
Phenotype Phase Plane Analysis



# Flux Variability Analysis & Parsimonious Flux Balance Analysis



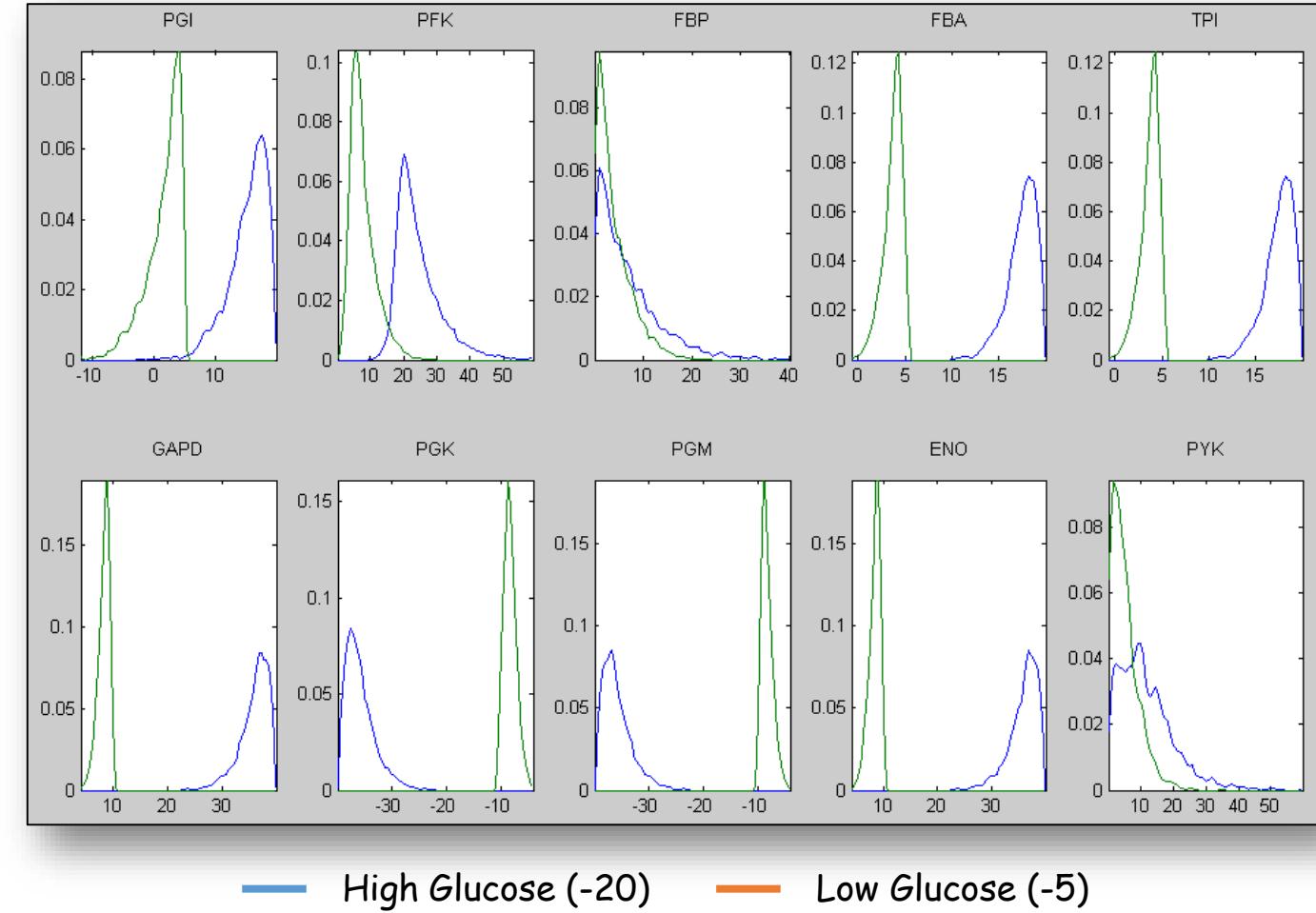
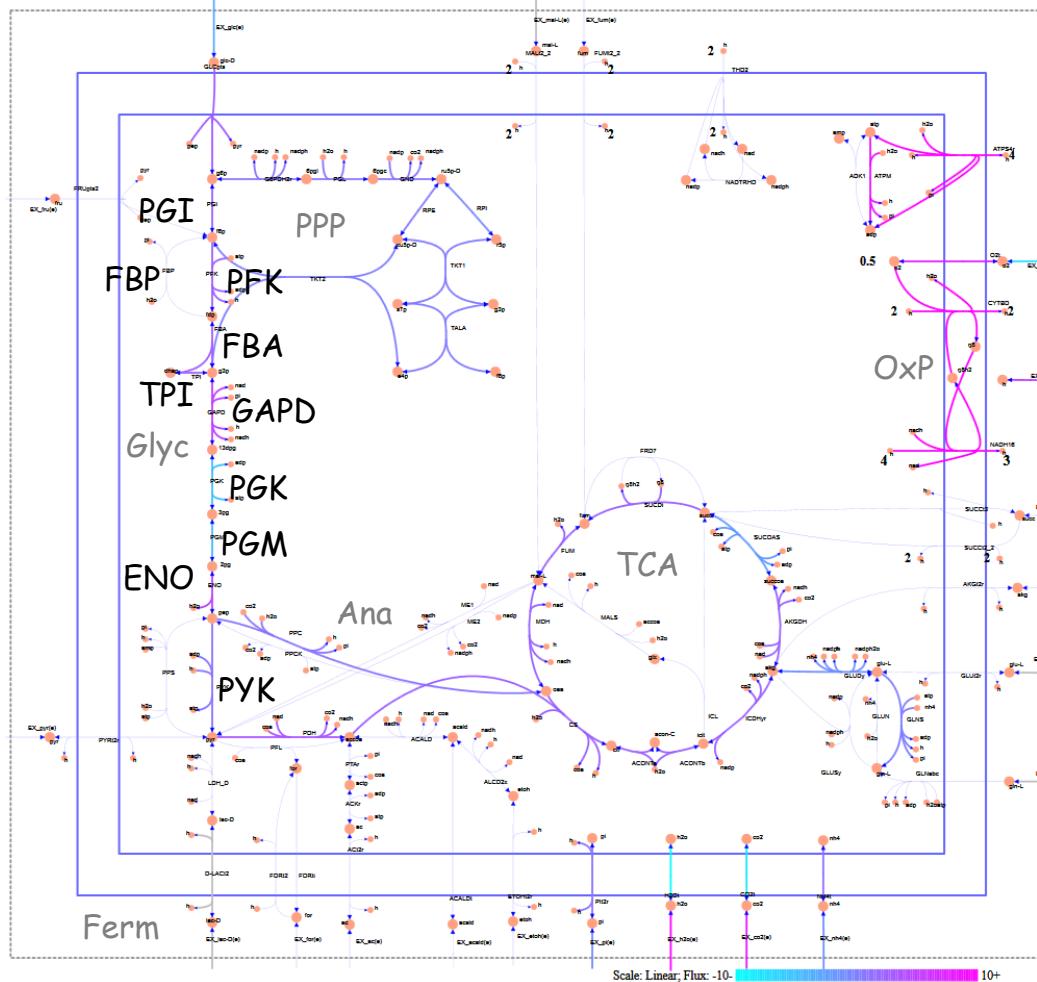
Flux Variability Analysis



Parsimonious Flux Balance Analysis



# Randomized Sampling



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.



# "Analysis" Tutorials

- A step-by-step guide to parsimonious enzyme usage Flux Balance Analysis - pFBA
- Analyze Steady-State Community COBRA Models
- Atomically resolve a metabolic reconstruction
- Browse Networks in the Matlab Command Window Using surfNet
- Computation and analysis of microbe-microbe metabolic interactions
- Computation and analysis of rescued lethal gene deletions in a host-microbe model
- Creation and simulation of personalized microbiota models through metagenomic data integration
- Determining MinSpan vectors of COBRA model
- Flux Balance Analysis (FBA)
- Flux Variability analysis (FVA)
- genetic Minimal Cut Sets - gMCS
- Proton shuttle testing with sparse flux balance analysis
- Reaction essentiality across multiple models
- Relaxed Flux Balance Analysis: Recon 3
- Relaxed Flux Balance Analysis: Toy model
- Simulation of growth of human gut microbes on different diets
- Sparse Flux Balance Analysis
- Sparse flux balance analysis test for a minimal stoichiometrically balanced cycle involving ATP hydrolysis
- Sparse Linear Optimisation
- Thermodynamically constrain a metabolic model
- Uniform sampling
- Variational Kinetics
- Varying Parameters analysis

<https://opencobra.github.io/cobratoolbox/stable/tutorials/index.html>



## Flux Balance Analysis (FBA)

Author(s): Vanja Vlasov, Marouen Ben Guebila, Systems Biochemistry Group, LCSB, University of Luxembourg,

Thomas Pfau, Systems Biology Group, LSRU, University of Luxembourg

Reviewer(s): Ines Thiele, Catherine Clancy, Systems Biochemistry Group, LCSB, University of Luxembourg

Thomas Pfau, Systems Biology Group, LSRU, University of Luxembourg

### INTRODUCTION

Flux balance analysis (FBA) evaluates the metabolic flux distribution<sup>1</sup>, and is one of the most used modelling approaches for metabolic systems.

The applications of FBA for molecular systems biology include prediction of the growth rates, uptake rates, knockout lethality and product secretion. In FBA, the solution space is constrained by the assumption of a steady-state, under which each internal metabolite is consumed at the same rate as it is produced.

For the quantitative estimation of the metabolic fluxes, linear programming (LP) can be used to solve the stoichiometric matrix for a given objective function under different constraints. The constraints of the problem depict the space of all eligible possibilities from which an optimal solution can be selected;

$$\begin{aligned} \min_v \quad & c^T v \\ \text{s.t.} \quad & S v = b, \\ & l \leq v \leq u, \end{aligned}$$

Equation 1: Formula of standard FBA.



# “Analysis” Function Categories

The analysis folder contains all of the methods for interrogation of the properties of a reconstruction or model, and combinations thereof, as well as the prediction of biochemical network states using constraint-based models.

Coupling - Reaction coupling

Deletion - Deleting reactions & genes

Dynamic FBA - Dynamic flux balance analysis

Exploration - Exploring model characteristics

FBA - Flux balance analysis

findMIIS - Finds the Minimal Irreducible Infeasible Subset

fluxEnrichmentAnalysis - Flux enrichment analysis

FVA - Flux variability analysis

geometricFBA - Geometric flux balance analysis

gMCS - genetic Minimal Cut Sets

Kldistance - Kullback-Leibler Distance

mdFBA - Metabolic dilution flux balance analysis

MOMA - Minimization of metabolic adjustment

MultiSpecies - Multispecies modeling tools

parsimoniousFBA - Parsimonious flux balance analysis

relaxedFBA - Relaxed flux balance analysis

rFBA - Regulatory flux balance analysis

rMTA - Metabolic Transformation Analysis

robustnessAnalysis - Robustness analysis

ROOM - Regulatory on/off minimization

Rumba - Flux change at metabolic branch points

Sampling - Sampling analysis

sparseFBA - Sparse flux balance analysis

Subspaces - Working with fundamental subspaces of S

Thermo - Thermodynamic analysis tools

Topology - Topology tools

Transport - Analyzing model transport systems

uFBA - Micro flux balance analysis



# COBRA Toolbox v.3.0

- COBRA Toolbox v.3.0 Overview
- Base Functions
- Analysis Functions
- Data Integration Functions 
- Design Functions
- Reconstruction Functions
- Visualization Functions
- COBRA Toolbox Examples

The “dataIntegration” folder contains the methods for integration of metabolomic, transcriptomic, proteomic, and thermodynamic data with a reconstruction or model. This includes functions to support the following

- ChemoInformatics
- Fluxomics
- Metabotools
- Transcriptomics

Laurent Heirendt et al, Creation and analysis of biochemical constraint-based models: the COBRA Toolbox v3.0, *Nature Protocols*, volume 14, pages 639-702, 2019



# Data Integration Functionality

- **Cheminformatics**

- Cheminformatics
  - ✓ "Cheminformatics is the mixing of those information resources to transform data into information and information into knowledge for the intended purpose of making better decisions faster in the area of drug lead identification and optimization." (Brown, Frank (2005). "*Editorial Opinion: Chemoinformatics - a ten year update*". *Current Opinion in Drug Discovery & Development.* 8 (3): 296-302.)

- **Fluxomics**

- Fluxomics
  - ✓ Fluxomics describes the various approaches that seek to determine the rates of metabolic reactions within a biological entity. The significance of fluxomics is that metabolic fluxes determine the cellular phenotype. It has the added advantage of being based on the metabolome which has fewer components than the genome or proteome. (Cascante, Marta et al (eds.). *Fluxomics*. Springer International Publishing. pp. 237-250.)

- **Metabolomics**

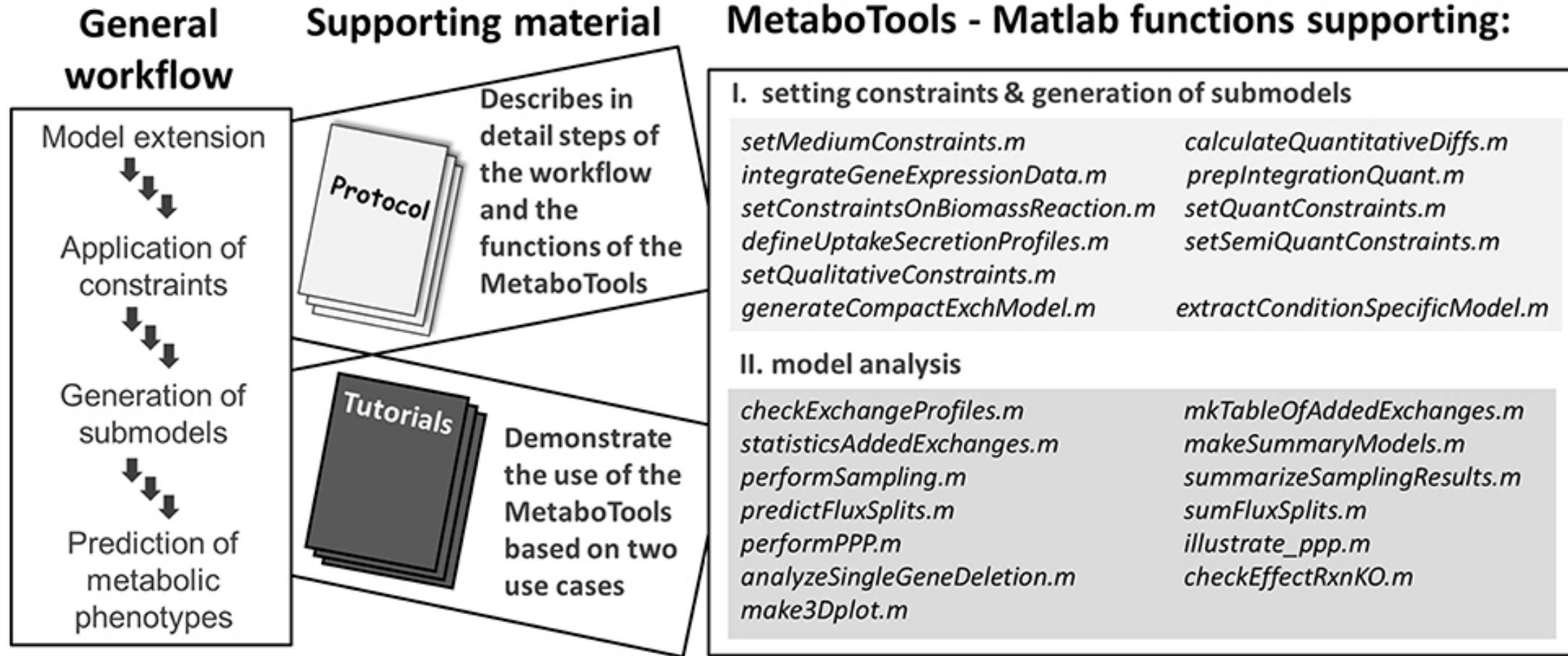
- Metabolomics
  - ✓ The scientific study of chemical processes involving metabolites, the small molecule intermediates and products of metabolism. Specifically, metabolomics is the "systematic study of the unique chemical fingerprints that specific cellular processes leave behind", the study of their small-molecule metabolite profiles. ( Daviss B (April 2005). "Growing pains for metabolomics". *The Scientist.* 19 (8): 25-28.)

- **Transcriptomics**

- Transcriptomics
  - ✓ Transcriptomics technologies are the techniques used to study an organism's transcriptome, the sum of all of its RNA transcripts. (McGettigan PA (February 2013). "Transcriptomics in the RNA-seq era". *Current Opinion in Chemical Biology.* 17 (1): 4-11.)



# MetaboTools Workflow, Protocol and Tutorials



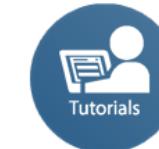
Aurich, Maike K., Ronan MT Fleming, and Ines Thiele. "Metabotools: A comprehensive toolbox for analysis of genome-scale metabolic models." *Frontiers in physiology* 7 (2016): 327.



# “Data Integration” Tutorials

- Extraction of context-specific models
- Metabotools tutorial I
- Metabotools tutorial II - Integration of quantitative metabolomic data
- unsteady-state Flux Balance Analysis (uFBA)

<https://opencobra.github.io/cobratoolbox/stable/tutorials/index.html>



## Metabotools tutorial I

Authors: Maike K. Aurich, Sylvain Arreckx, Systems Biochemistry Group, LCSB, University of Luxembourg.

Reviewer(s): Anne Richelle, Lewis Lab at University of California, San Diego.

### INTRODUCTION

In this tutorial, we generate contextualized models of two lymphoblastic leukemia cell lines, CCRF-CEM and Molt- 4 cells. They will be generated by integrating semi-quantitative metabolomic data, transcriptomic data, and growth rates. We will afterwards analyze the solution space of these models by using a sampling analysis.

Before running a section in the tutorial, read the corresponding sections in the MetaboTools protocol and supplemental tutorial (Data sheet 2, <http://journal.frontiersin.org/article/10.3389/fphys.2016.00327/full>).

### PROCEDURE

Clear workspace and initialize the COBRA Toolbox

```
clear  
initCobraToolbox(false) % false, as we don't want to update
```

#### Step 0 - Define the output location and set the LP solver

Define the output path and set the solver for LP problem

```
global CBTDIR % set path to cobratoolbox (pathToCOBRA)  
outputPath = pwd;% ouputPath = 'ADD YOUR PATH TO YOUR OUTPUT FOLDER'  
solver = 'glpk'; % solver = 'ADD YOUR SOLVER'; %, e.g., 'cplex_direct' for ILOG  
solverOK = changeCobraSolver(solver, 'LP');
```



# COBRA Toolbox v.3.0

- COBRA Toolbox v.3.0 Overview
- Base Functions
- Analysis Functions
- Data Integration Functions
- Design Functions
- Reconstruction Functions
- Visualization Functions
- COBRA Toolbox Examples

The “design” folder contains new strain design methods and a new modeling language interface to GAMS (general algebraic modeling system), a high-level modeling system for mathematical optimization. They include the functions necessary to use the “optForce” design tools.

- optForce
- optForceGAMS

Laurent Heirendt et al, Creation and analysis of biochemical constraint-based models: the COBRA Toolbox v3.0, Nature Protocols, volume 14, pages 639-702, 2019



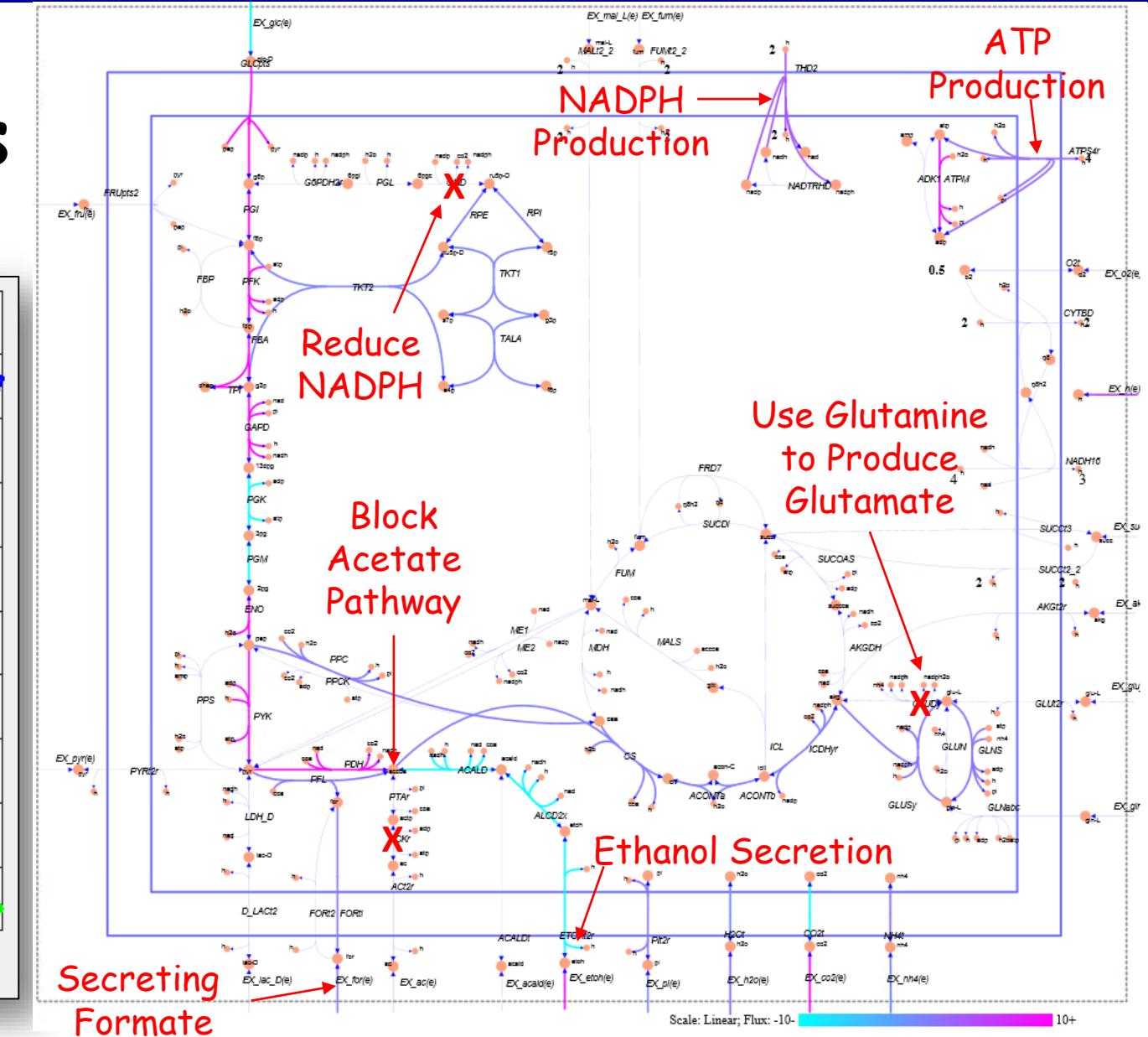
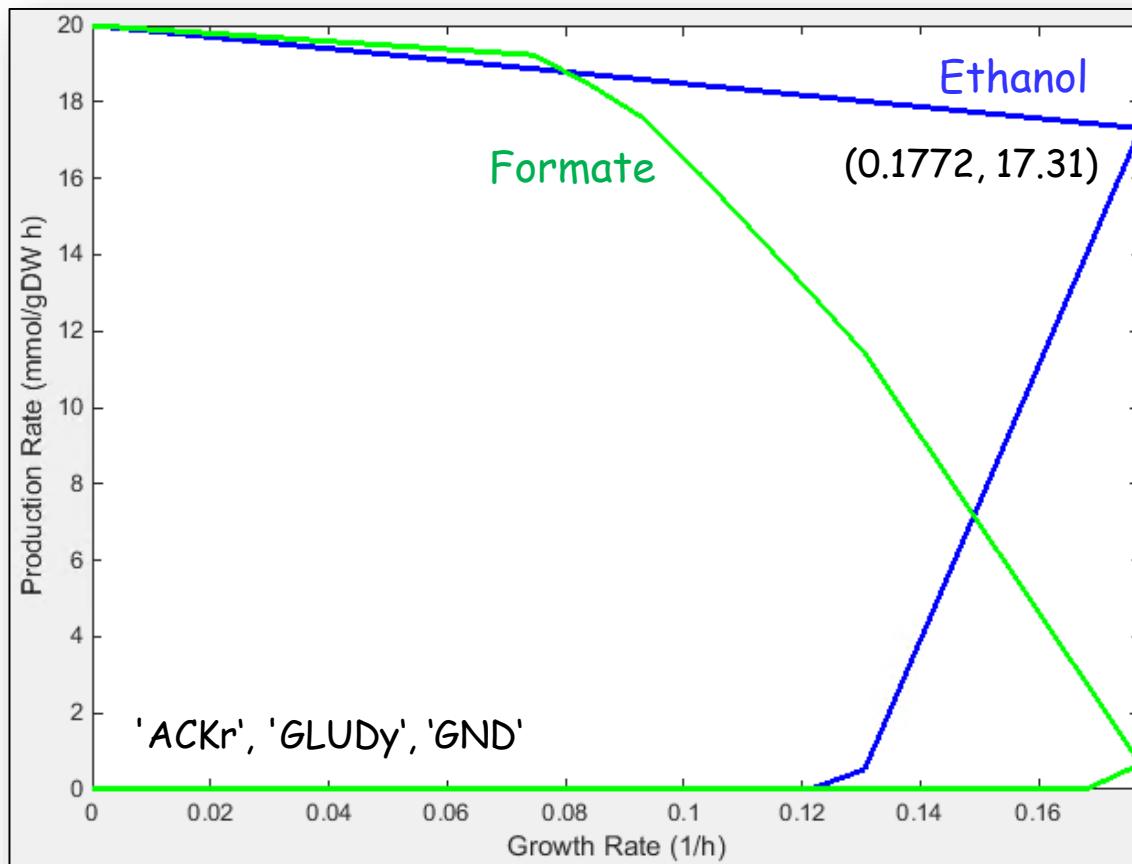
# Strain Design Algorithms

A variety of strain design algorithms are implemented within the COBRA Toolbox v.3.0, including

- **OptKnock** - OptKnock is an algorithm suggesting the genetic manipulation that lead to the overproduction of a specified metabolite [1]. OptKnock pinpoints which set of reactions to remove (i.e. deletion of the genes associated to these reactions) from a metabolic network to obtain a mutant that will produce a particular target of interest at a higher rate than the wild-type strain. (Burgard, A. P., Pharkya, P. & Maranas, C. D. (2003). OptKnock: A Bilevel Programming Framework for Identifying Gene Knockout Strategies for Microbial Strain Optimization. *Biotechnology and Bioengineering*, 84(6), 647-657.)
- **OptGene** - The optGene algorithm will find sets of reactions that should increase the production of your target when they are deleted from the network. Since optGene is based on a genetic algorithm, the solutions found could vary between different runnings, even though the algorithm has been executed with the same input parameters. (Patil, K. R., Rocha, I., Förster, J., & Nielsen, J. (2005). Evolutionary programming as a platform for *in silico* metabolic engineering. *BMC bioinformatics*, 6(1), 308.)
- **GDLS** - GDLS (Genetic Design Local Search) attempts to find genetic designs with greater *in silico* production of desired metabolites. (Lun, D. S. et al. Large-scale identification of genetic design strategies using local search. *Mol. Syst. Biol.* 5, 296 (2009).)
- **OptForce** - the OptForce method can identify not only gene deletion but also up- and downregulation strategies. (Ranganathan, S., Suthers, P. F. & Maranas, C. D. OptForce: an optimization procedure for identifying all genetic manipulations leading to targeted overproductions. *PLoS Comput. Biol.* 6, e1000744 (2010).)



# Gene/Reaction Knockouts





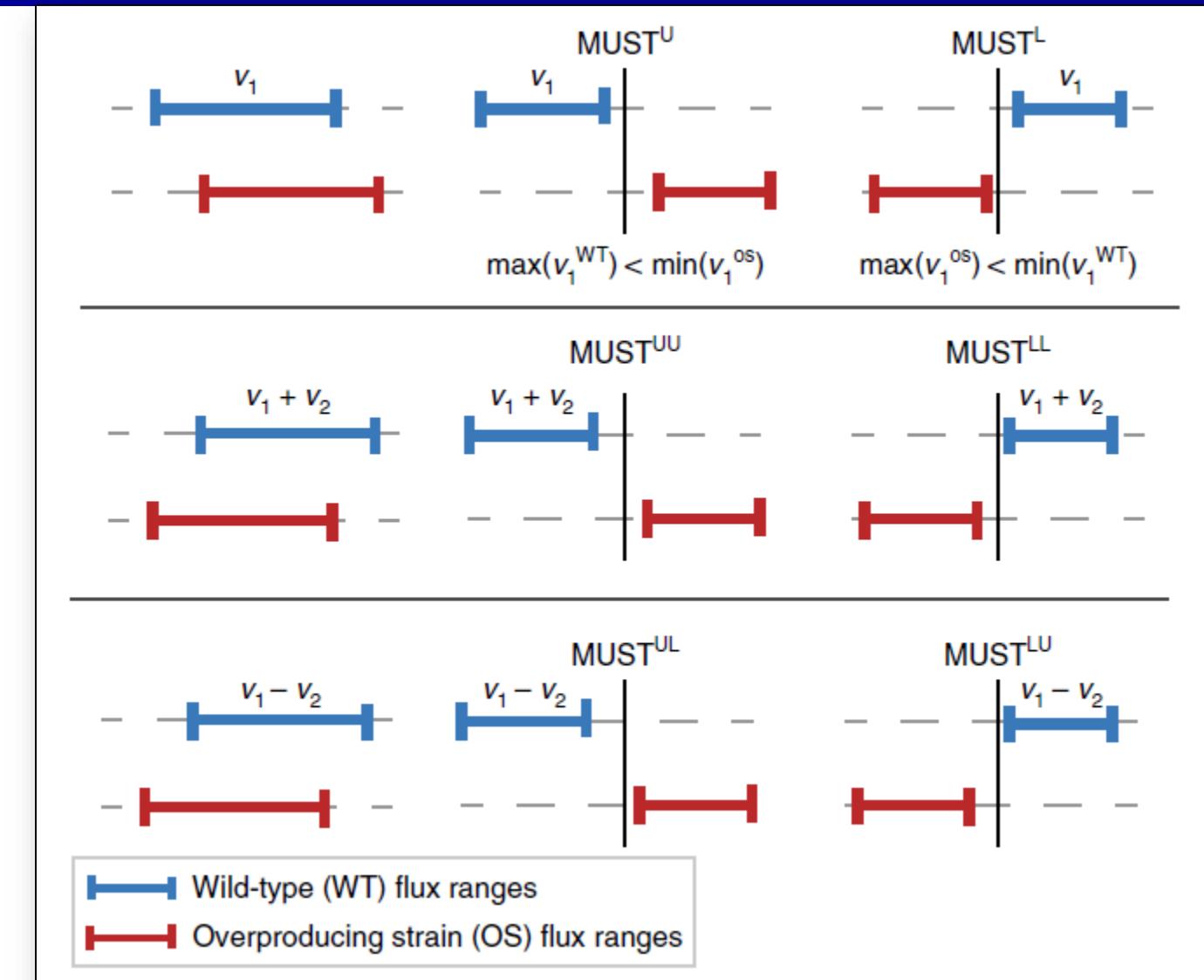
# OptForce

In the OptForce procedure, the MUST sets are determined by contrasting the flux ranges obtained using flux variability analysis (FVA) of a wild-type (blue bars) and an overproducing strain (red bars).

The first order MUST sets (top panel) are denoted MUSTL and MUSTU. For instance, a reaction belongs to the MUSTU set if the upper bound of the flux range in the wild-type is less than the lower bound of the flux range of the overproducing strain.

The center and bottom panels show all possible second-order MUST sets.

Ranganathan, S., Suthers, P. F. & Maranas, C. D. OptForce: an optimization procedure for identifying all genetic manipulations leading to targeted overproductions. *PLoS Comput. Biol.* 6, e1000744 (2010).



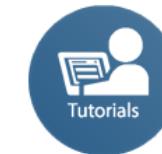
Laurent Heirendt et al, Creation and analysis of biochemical constraint-based models: the COBRA Toolbox v3.0, *Nature Protocols*, volume 14, pages 639-702, 2019



# “Design” Tutorials

- OptForce
- OptGene Tutorial
- OptKnock Tutorial
- tutorialNoName

<https://opencobra.github.io/cobratoolbox/stable/tutorials/index.html>



## OptForce

**Author:** Sebastián N. Mendoza, Center for Mathematical Modeling, University of Chile.  
snmendoza@uc.cl

**Reviewer(s):** Chiam Yu Ng (Costas D. Maranas group), Lin Wang (Costas D. Maranas group), John Sauls

### INTRODUCTION:

In this tutorial we will run optForce. For a detailed description of the procedure, please see [1]. Briefly, the problem is to find a set of interventions of size "K" such that when these interventions are applied to a wild-type strain, the mutant created will produce a particular target of interest in a higher rate than the wild-type strain. The interventions could be knockouts (lead to zero the flux for a particular reaction), upregulations (increase the flux for a particular reaction) and downregulations (decrease the flux for a particular reaction).

For example, imagine that we would like to increase the production of succinate in Escherichia coli. Which are the interventions needed to increase the production of succinate? We will approach this problem in this tutorial and we will see how each of the steps of OptForce are solved.

### MATERIALS

### EQUIPMENT

1. MATLAB
2. A solver for Mixed Integer Linear Programming (MILP) problems. For example, Gurobi.

### EQUIPMENT SETUP

Use changeCobraSolver to choose the solver for MILP problems.



# COBRA Toolbox v.3.0

- COBRA Toolbox v.3.0 Overview
- Base Functions
- Analysis Functions
- Data Integration Functions
- Design Functions
- • Reconstruction Functions
- Visualization Functions
- COBRA Toolbox Examples

The “reconstruction” folder contains all of the methods associated with the reconstruction and refinement of a biochemical network to match experimental data, as well as the conversion of a reconstruction into various forms of constraint-based models. They include functions to support the following capabilities.

- Comparison
- fastGapFill
- growthExpMatch
- modelGeneration
- rBioNet
- Refinement

Laurent Heirendt et al, Creation and analysis of biochemical constraint-based models: the COBRA Toolbox v3.0, *Nature Protocols*, volume 14, pages 639–702, 2019



# Main Fields of A Standard Model Structure

Field name	Size	Data type	Field description	Field name	Size	Data type	Field description
.b	$m \times 1$	Double	The coefficients of the constraints of the metabolites ( $Sv = b$ )	.subSystems	$n \times 1$	Cell of cell of char	Subsystem assignments for each reaction (one reaction may correspond to multiple subsystems)
.csense	$m \times 1$	Char	The sense of the constraints represented by $b$ ; each row is either 'E' (equality), 'L' (less than), or 'G' (greater than)	.ub	$n \times 1$	Double	Upper bounds for fluxes through the reactions
.metCharges	$m \times 1$	Numeric	The charge of the respective metabolite (NaN if unknown)	.S	$m \times n$	Numeric	The stoichiometric matrix containing the model structure (for large models, a sparse format is suggested)
.metFormulas	$m \times 1$	Cell of char	Elemental formula for each metabolite	.geneNames	$g \times 1$	Cell of char	Full name of each corresponding gene
.metInChiString	$m \times 1$	Cell of char	Formula for each metabolite in the InCHI strings format	.genes	$g \times 1$	Cell of char	Identifiers of the genes in the model
.metNames	$m \times 1$	Cell of char	Full name of each corresponding metabolite	.proteinNames	$g \times 1$	Cell of char	Full name for each protein
.mets	$m \times 1$	Cell of char	Identifiers of the metabolites	.proteins	$g \times 1$	Cell of char	Proteins associated with each gene (one protein per gene)
.metSmiles	$m \times 1$	Cell of char	Formula for each metabolite in SMILES format	.rxnGeneMat	$n \times g$	Numeric or logical	Matrix with rows corresponding to reactions and columns corresponding to genes
.c	$n \times 1$	Double	The objective coefficient of the reactions	.compNames	$c \times 1$	Cell of char	Descriptions of the compartments (compNames ( $m$ ) is associated with comps ( $m$ ))
.grRules	$n \times 1$	Cell of char	A string representation of the gene-protein-reaction rules defined in a readable format	.comps	$c \times 1$	Cell of char	Symbols for compartments
.lb	$n \times 1$	Double	Lower bounds for fluxes through the reactions	.osenseStr	$1 \times 3$	Char	The objective sense: either 'max' (maximization) or 'min' (minimization)
.rxnConfidenceScores	$n \times 1$	Numeric	Confidence scores for reaction presence (0-5, with 5 being the highest confidence)				
.rxnECNumbers	$n \times 1$	Cell of char	Enzyme Commission (EC) number for each reaction				
.rxnNames	$n \times 1$	Cell of char	Full name of each corresponding reaction				
.rxnNotes	$n \times 1$	Cell of char	Description of each corresponding reaction				
.rxnReferences	$n \times 1$	Cell of char	Description of references for each corresponding reaction				
.rxns	$n \times 1$	Cell	Identifiers of the reactions				

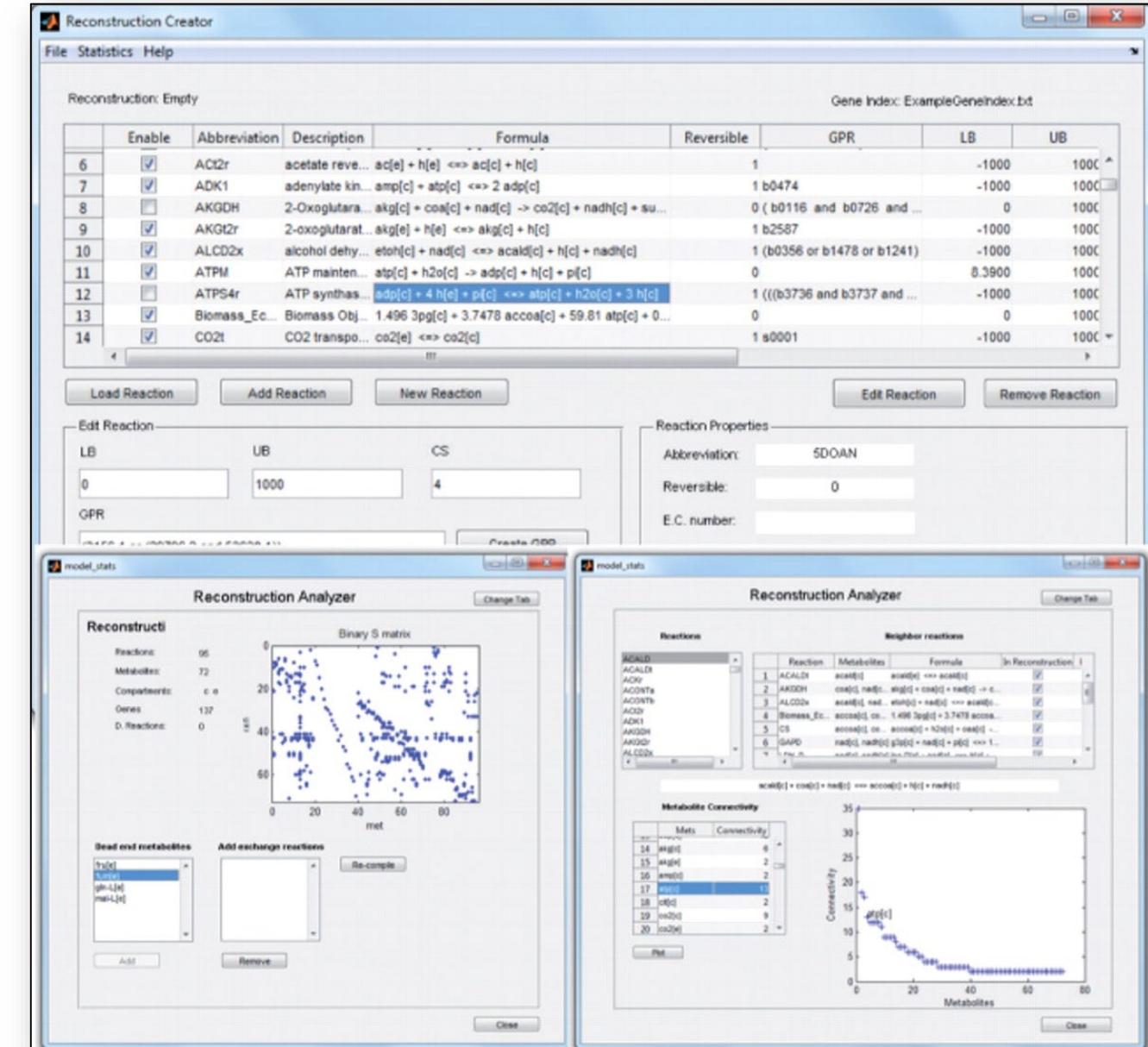
Laurent Heirendt et al, Creation and analysis of biochemical constraint-based models: the COBRA Toolbox v3.0, Nature Protocols, volume 14, pages 639-702, 2019



## rBioNet

- To manipulate an existing reconstruction in the COBRA Toolbox, one can use rBioNet, use a spreadsheet, or generate scripts with reconstruction functions. Each approach has its advantages and disadvantages. When adding a new reaction or gene-protein-reaction association, rBioNet ensures that reconstruction standards are satisfied, but it may make the changes less tractable when many reactions are added.
- A spreadsheet-based approach is tractable, but allows only for the addition, and not the removal, of reactions.
- By contrast, using reconstruction functions provides an exact specification for all the refinements made to a reconstruction.
- One can also combine these approaches by first formulating the reactions and gene-protein-reaction associations with rBioNet and then adding sets of reactions using reconstruction functions.

Stefan Gretar Thorleifsson, Ines Thiele, rBioNet: A COBRA toolbox extension for reconstructing high-quality biochemical networks, Bioinformatics, Volume 27, Issue 14, 15 July 2011, Pages 2009-2010,





# "Reconstruction" Tutorials

- Adding biological constraints to a flux balance model
- Adding more complex constraints on a model
- Constraint-based modelling concepts
- Convert a reconstruction into a flux balance analysis model
- Create a generic subnetwork from Recon 3D
- Create an overview table with model properties
- Creating a Model
- E.coli Core Model for Beginners (PART 1)
- E.coli Core Model for Beginners (PART 2)
- E.coli Core Model for Beginners (PART 3)
- Example use of functions listed in the Standard operating procedure for metabolic reconstruction.
- FastGapFill tutorial
- Find leakage and siphon modes in a reconstruction
- Generation and manipulation of reconstructions with rBioNet
- How to use modelBorgifier
- Model manipulation
- Numerical properties of a reconstruction
- Test physiologically relevant ATP yields from different carbon sources for a metabolic model
- Testing basic properties of a metabolic model (aka sanity checks)
- Testing chemical and biochemical fidelity

<https://opencobra.github.io/cobratoolbox/stable/tutorials/index.html>

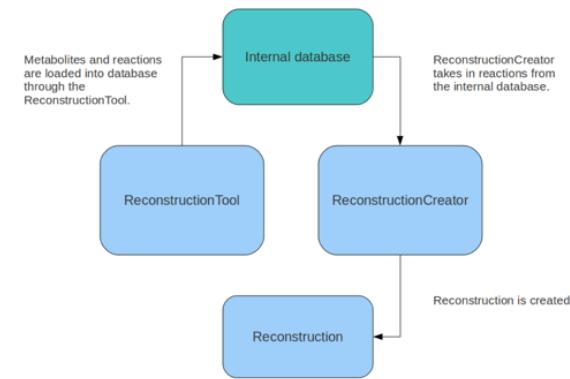
[!\[\]\(c3873e051643137b5c55fb35a066ddca\_img.jpg\) Document \[.pdf\]](#)[!\[\]\(737e63e3461d429e55c00ad15e76c608\_img.jpg\) Live Script \[.mlx\]](#)[!\[\]\(71340ad9bf49b335e043e689b30f99df\_img.jpg\) Source File \[.m\]](#)[!\[\]\(f6ff18f7862261b3e797943592941285\_img.jpg\) View on GitHub](#)[!\[\]\(d1da8e3794e4ffea29f08d5b6988efd8\_img.jpg\) Tutorials](#)

**Generation and manipulation of reconstructions with rBioNet**

**Author(s):** Ines Thiele, Ronan M. T. Fleming, Systems Biochemistry Group, LCSB, University of Luxembourg.

**Reviewer(s):** Catherine Clancy, Stefania Magnusdottir, LCSB, University of Luxembourg.

rBioNet is a reconstruction tool that lets you assemble reconstruction in a user friendly environment. In this tutorial you shall learn how you can use this tool to either start a new reconstruction or load in an existing one, followed by, its analysis. The tool consists of 3 main parts, i.e., metabolite creator, reaction creator and reconstruction creator. The metabolite creator is used to add in metabolites and its associated information, i.e., its elemental formula, charge, identifiers (for e.g., KEGG ID, PubChem ID etc.) and other associated attributes. Alternatively, a text file containing all the necessary information in the same order as in the metabolite database can be loaded directly. The reaction creator is used to formulate reactions and as stated before a text file containing all the necessary information about the reaction abbreviation, description, formula, reversibility, confidence score, notes, references. Alternatively, a text file containing all the necessary information in the same order as in the reaction database can be loaded on to the reaction creator directly. The reconstruction creator is used to load in reactions from the reactions database and then assign GPRs (gene-protein-reaction association), subsystem, add in more information in the notes and reference section. Once you have completed your reconstruction you can look at the S-matrix, identify dead ends, look for neighboring reaction to a particular reaction and plot metabolite connectivity in the reconstruction creator with its statistics function.



```
graph TD; RT[ReconstructionTool] -- "Metabolites and reactions are loaded into database through the ReconstructionTool." --> ID[Internal database]; ID -- "ReconstructionCreator takes in reactions from the internal database." --> RC[ReconstructionCreator]; RC -- "Reconstruction is created." --> R[Reconstruction]
```



# COBRA Toolbox v.3.0

- COBRA Toolbox v.3.0 Overview
- Base Functions
- Analysis Functions
- Data Integration Functions
- Design Functions
- Reconstruction Functions
- Visualization Functions
- COBRA Toolbox Examples

The “visualization” folder contains all of the methods for the visualization of predictions within a biochemical network context, using various biochemical cartography tools that interoperate with the COBRA Toolbox v.3.0. They include functions to support the following capabilities.

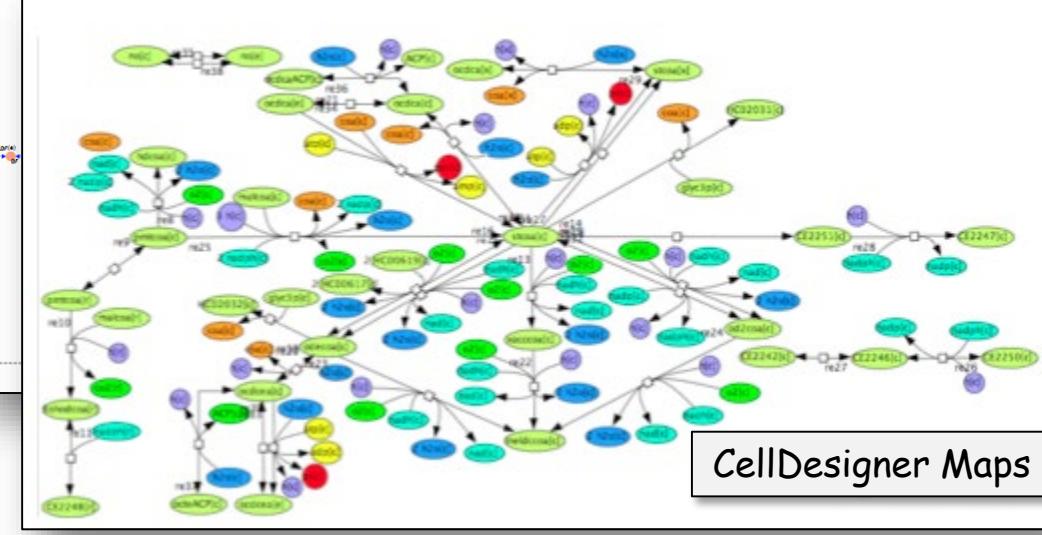
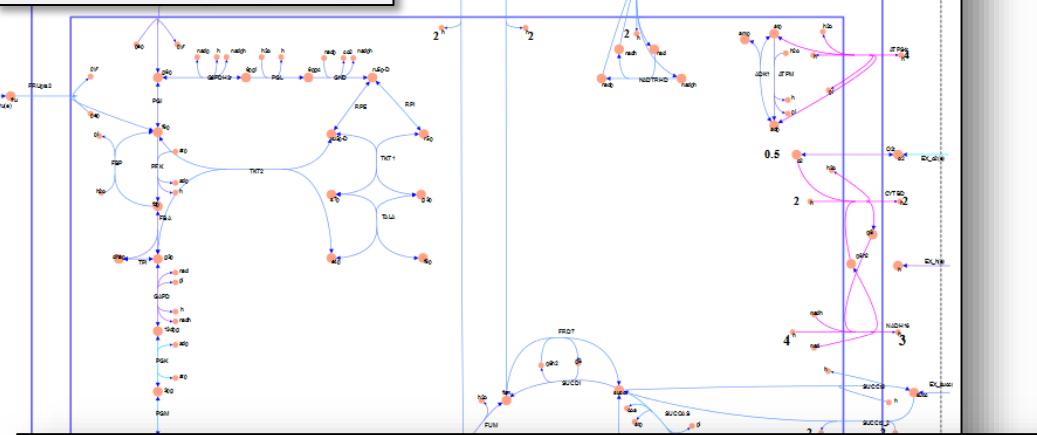
- cellDesigner
- Maps
- MetabolicCartography
- visualizeEpistasis

Laurent Heirendt et al, Creation and analysis of biochemical constraint-based models: the COBRA Toolbox v3.0, Nature Protocols, volume 14, pages 639-702, 2019



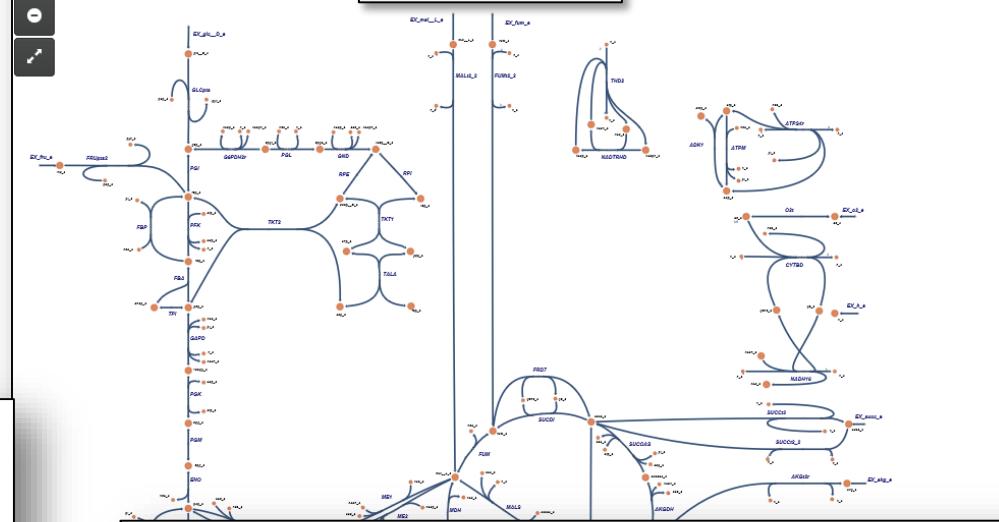
# Visualization Tools

COBRA Legacy Maps

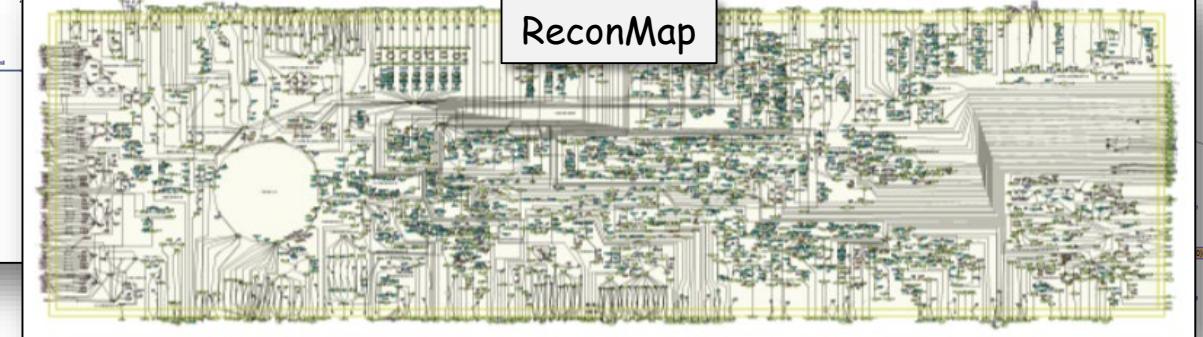


Map • Model • Data • Edit • View • ?

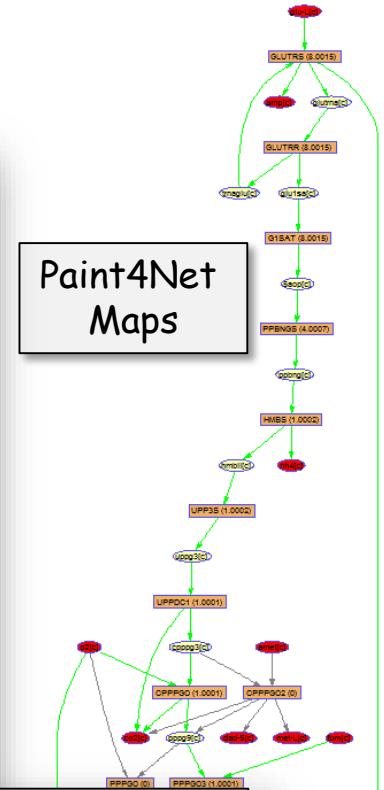
Escher Maps



ReconMap



Paint4Net Maps



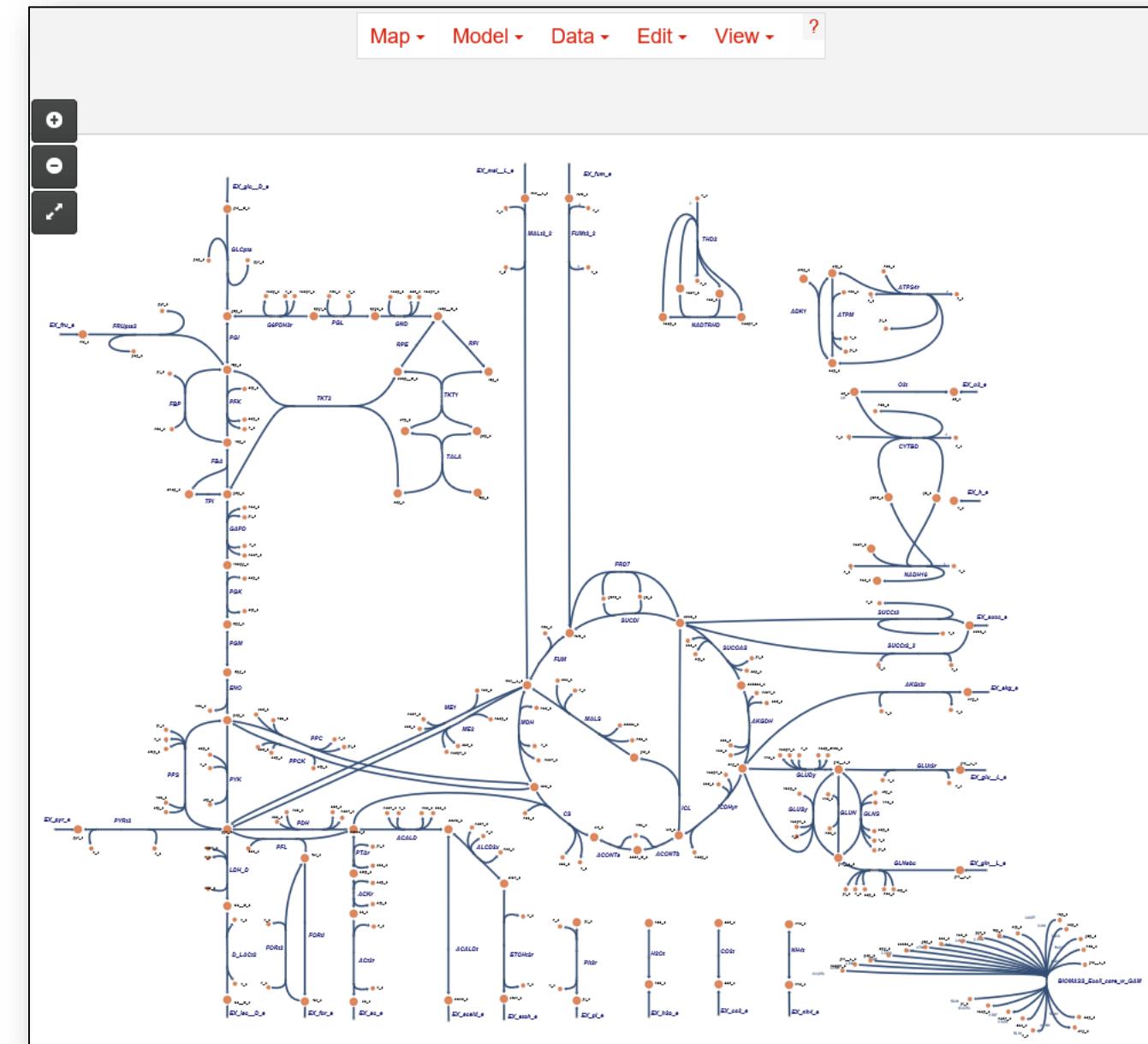
Laurent Heirendt et al, Creation and analysis of biochemical constraint-based models:  
the COBRA Toolbox v3.0, Nature Protocols, volume 14, pages 639-702, 2019



# 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" (selected), "Model (Optional)", and "Tool". The "Map" section has a dropdown for "Core metabolism (e\_coli\_core)" and a "Load map" button. The "Model (Optional)" section has a dropdown for "e\_coli\_core" and a "Builder" dropdown. The "Tool" section has a dropdown for "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".





# “Visualization” Tutorials

- Introduction to Cell Designer
- Metabolic visualisation in ReconMap (Minerva)
- Paint4Net: visualisation toolbox for COBRA
- Visualisation and map manipulation in Cell Designer  
(PART 1)
- Visualisation and map manipulation in Cell Designer  
(PART 2)

<https://opencobra.github.io/cobratoolbox/stable/tutorials/index.html>



## Paint4Net: visualisation toolbox for COBRA

### Author(s):

Andrejs Kostromins, Biosystems Group, Department of Computer Systems, Latvia University of Agriculture, Liela iela 2, LV-3001 Jelgava, Latvia.

Egils Stalidzans, Institute of Microbiology and Biotechnology, University of Latvia, Jelgavas iela 1, LV-1004, Latvia.

### Reviewer(s):

Agris Pentjuss, Institute of Microbiology and Biotechnology, University of Latvia, Jelgavas iela 1, LV-1004, Latvia.

Almut Heinken, Luxembourg Centre for Systems Biomedicine, Universiy of Luxembourg, 6 avenue du Swing, Belvaux, L-4367, Luxembourg.

### IMPORTANT NOTE

Paint4Net uses Bioinformatics Toolbox to generate visualisation layout, however it is not supported in .mlx causing an error during function execution. Thus the functions involving visualisation were run in regular MATLAB command window and each visualisation layout was saved as a static figure and placed in the .mlx tutorial, while the corresponding functions were run in .mlx with visualisation feature turned off (input argument `drawMap` was set to 'false') to get outputs (without visualisation) in .mlx without crashing. Be aware of this issue when you are running the functions in .mlx. All `drawMap` input arguments are set to 'true' in .mlx. Change it to 'false' to avoid an error.



# COBRA Toolbox v.3.0

- COBRA Toolbox v.3.0 Overview
- Base Functions
- Analysis Functions
- Data Integration Functions
- Design Functions
- Reconstruction Functions
- Visualization Functions
- • COBRA Toolbox Examples



# Installing COBRA Toolbox 3.0

The screenshot shows a web page for the COBRA Toolbox documentation. On the left, there's a sidebar with a logo, a search bar, and navigation links: Home, Installation (which is expanded), System Requirements, Download and installation, Solver compatibility, Binaries and Compilers, Solver installation, and Test the installation.

The main content area has a breadcrumb trail "Docs » Installation" and a URL "https://openCOBRA.github.io/COBRAtoolbox/stable/installation.html#". The main title is "Installation". Below it is a section titled "System Requirements" with the sub-question "Can I check if everything is properly set up before I start?". Under "MATLAB", it says: "Please ensure that you have a compatible and working MATLAB installation. The list of compatible solvers is available [here](#)". A note below states: "No support is provided for versions older than R2014b. MATLAB is released on a twice-yearly schedule. After the latest release (version b), it may be a couple of months before certain methods with dependencies on other software become compatible. For example, the latest releases of MATLAB may not be compatible with the existing solver interfaces, necessitating an update of the MATLAB interface provided by the solver developers, or an update of the COBRA Toolbox, or both."

## Docs » Installation

<https://openCOBRA.github.io/COBRAtoolbox/stable/installation.html#>

## Installation

### System Requirements

Can I check if everything is properly set up before I start?

#### MATLAB

Please ensure that you have a compatible and working MATLAB installation. The list of compatible solvers is available [here](#).

No support is provided for versions older than R2014b. MATLAB is released on a twice-yearly schedule. After the latest release (version b), it may be a couple of months before certain methods with dependencies on other software become compatible. For example, the latest releases of MATLAB may not be compatible with the existing solver interfaces, necessitating an update of the MATLAB interface provided by the solver developers, or an update of the COBRA Toolbox, or both.



Starting Matlab

The screenshot shows the MATLAB R2019a interface. The Command Window displays the path: C:\Users\hinton\cobratoolbox. A red arrow points from the text "COBRA Toolbox Directory" to this path. The Current Folder browser on the left shows the contents of the cobratoolbox directory, including subfolders like .artenolis, .github, .tmp, binary, docs, external, papers, src, test, and tutorials, along with files such as .artenolis.yml, .gitattributes, .gitignore, .gitmodules, codecov.yml, initCobraToolbox.m, LICENSE.md, and README.rst. The Workspace browser on the right shows the Command History window with the following code:

```
x = linspace(0,xmax...  
phi(1*10^-6)  
li  
phi(x) = exp(li*k*s...  
phi(x) = exp(li*k*s...  
phi(x) = exp(li*k*s...  
phi = exp(li*k*sqrt(...  
phi(x) = exp(li*k*s...  
x  
30  
0.15
```



# Initializing the COBRA Toolbox

The screenshot shows the MATLAB R2019a interface with the following details:

- Current Folder:** C:\Users\hinton\cobratoolbox
- Command Window:** Displays the command `>> initCobraToolbox`. Below it, a large decorative logo for "COBRA Toolbox" is displayed, consisting of a grid of small squares forming a stylized letter "C". To the right of the logo, the text "Constraint-Based Re", "The COBRA Toolbo", "Documentation:", and a link "<http://opencobra.github.io>" are visible.
- Workspace:** Shows variables like 0.15, 557, 0.5, 1, 30, 0.15, 557, 0.5, 0, and a timestamp "%-- 7/18/2019 11:47...".
- Command History:** Shows previous commands run in the session.
- Bottom Left:** A pink callout box contains the text "Run initCobraToolbox.m".

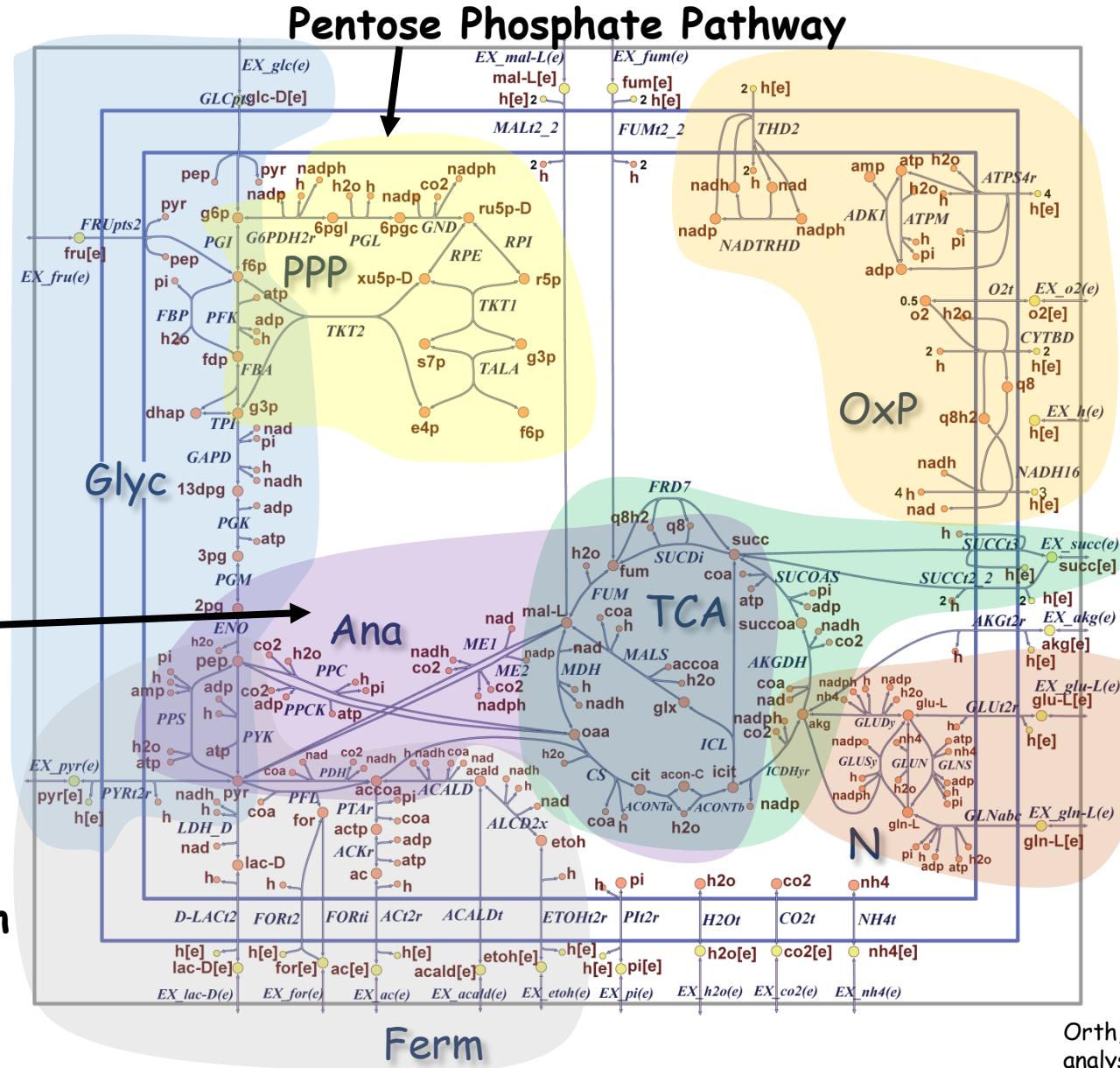


# *E. coli* Core Model

Glycolysis

Glyoxylate Cycle,  
Gluconeogenesis, and  
Anapleurotic Reactions

Fermentation



Oxidative  
Phosphorylation and  
Transfer of Reducing  
Equivalents

Tricarboxylic Acid  
Cycle (TCA)

Nitrogen  
Metabolism

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



# Loading a COBRA-based Model

The screenshot shows the MATLAB environment with several windows open:

- Current Folder:** Displays the directory structure, including files like `AerobicGlucoseBioMass.m`, `e_coli_core.json`, and `Introduction_lecture.mlx`.
- Command Window:** Shows the command `>> model = readCbModel('e coli_core_model.mat')` being run. A red box highlights the file path. Below it, the output shows the `model` variable is a `struct` with fields:
  - `S: [72x95 double]` (Stoichiometric Matrix)
  - `mets: {72x1 cell}` (Metabolites)
  - `b: [72x1 double]`
  - `csense: [72x1 char]`
  - `rxns: {95x1 cell}` (Reactions)
  - `lb: [95x1 double]` (Lower Bounds)
  - `ub: [95x1 double]` (Upper Bounds)
  - `c: [95x1 double]`
  - `osenseStr: 'max'` (Objective Function)
  - `genes: {137x1 cell}` (Genes)
  - `rules: {95x1 cell}`
  - `metCharges: [72x1 int32]`
  - `metFormulas: {72x1 cell}`
  - `metNames: {72x1 cell}`
  - `metInChiString: {72x1 cell}`
  - `metKEGGID: {72x1 cell}`
  - `metChEBIID: {72x1 cell}`
  - `metPubChemID: {72x1 cell}`
  - `grRules: {95x1 cell}`
  - `rxnGeneMat: [95x137 double]`
  - `rxnConfidenceScores: [95x1 double]`
  - `rxnNames: {95x1 cell}`
  - `rxnNotes: {95x1 cell}`
  - `rxnECNumbers: {95x1 cell}`
  - `rxnReferences: {95x1 cell}`
- Workspace:** Shows the variable `model` is a `1x1 struct`. A red arrow points to this entry.
- Command History:** Lists the commands run, including `initCobraToolbox`, `save path`, `gurobi_setup`, `savepath`, `initCobraToolbox`, `testAll`, and `Y`.



## COBRA Model Structure

The screenshot shows the MATLAB environment with several windows open:

- Current Folder:** Shows a list of files including 'Cobra Toolbox Matlab Files', 'Paint4Net v1.3', and various M-files related to metabolic modeling.
- Variables - model:** A central window displaying the contents of the 'model' variable. It is described as a "1x1 struct with 27 fields". The table lists fields such as 'modelVersion', 'rxns', 'mets', 'S', 'rev', 'c', 'metNames', 'metFormulas', 'lb', 'ub', 'metCharge', 'rules', 'genes', 'rxnGeneMat', 'grRules', 'subSystems', 'confidenceScores', and 'rxnReferences'. Each field is followed by its data type and size.
- Workspace:** A window on the right showing the variable 'model' as a '1x1 struct'.
- Command History:** A window at the bottom showing the command 'load('ecoli\_textbook.mat')'.

A red arrow points from the text "Double click on Model to open window with the model variables" to the 'model' entry in the Current Folder browser.

Double click on Model to open window with the model variables

Field	Value
modelVersion	1x1 struct
rxns	95x1 cell
mets	72x1 cell
S	72x95 sparse double
rev	95x1 double
c	95x1 double
metNames	72x1 cell
metFormulas	72x1 cell
lb	95x1 double
ub	95x1 double
metCharge	72x1 int32
rules	95x1 cell
genes	137x1 cell
rxnGeneMat	95x137 sparse double
grRules	95x1 cell
subSystems	95x1 cell
confidenceScores	95x1 cell
rxnReferences	95x1 cell
ECM	"Ecoli_K12_MG1655"



# 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

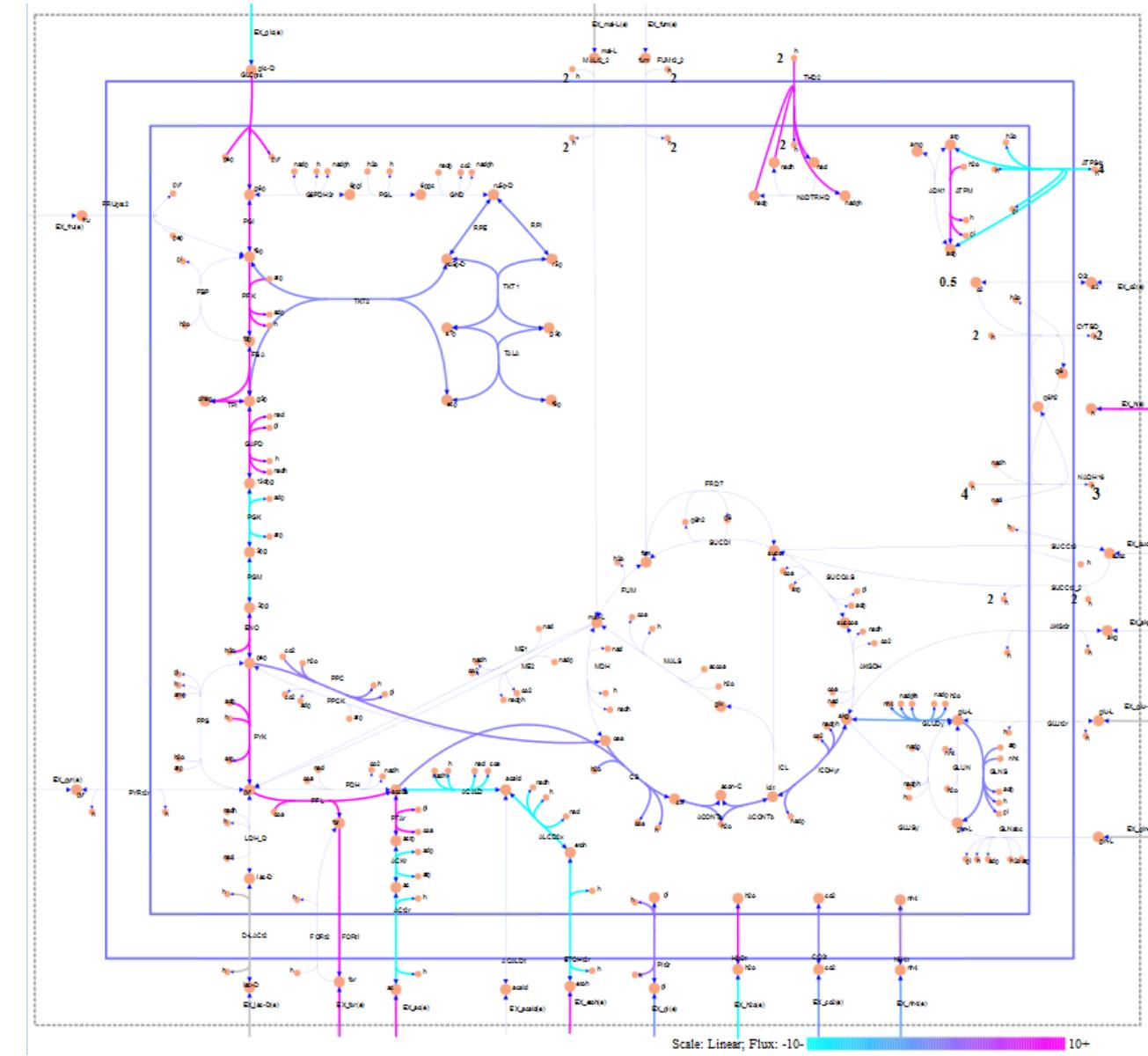


# Calculating the Flux Values

## Anaerobic Growth on Glucose

### Exchange Reactions

Biomass	0.470565
EX_ac(e)	15.1732
EX_co2(e)	-0.840759
EX_eto(h)	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





# Solving for Fluxes (Analysis/FBA)

The screenshot shows the MATLAB environment with the following components:

- Current Folder**: Displays the file tree with files like `.tmp`, `AerobicGlucoseBioMass.m`, `AerobicGlucoseBioMassEscher.m`, etc.
- Command Window**: Shows the command `>> solution = optimizeCbModel(model)` and its output. The output is a `struct` with fields:
  - `f: 0.8739` (labeled as "Growth-rate")
  - `v: [95x1 double]`
  - `y: [72x1 double]`
  - `w: [95x1 double]`
  - `s: [72x1 double]`
  - `solver: 'gurobi'`
  - `algorithm: 'default'`
  - `stat: 1`
  - `origStat: 'OPTIMAL'`
  - `time: 0.0160`
  - `basis: [1x1 struct]`
  - `vars_v: []`
  - `x: [95x1 double]` (labeled as "Flux Vector")
- Workspace**: Shows the variables `model` and `solution`.
- Command History**: Shows the commands run in the session, including `initCobraToolbox`, `AerobicGlucoseBioMass`, `load('ecoli_textbook.mat')`, `load('ecoli_core_model.mat')`, `solution = optimizeCbModel(model)`, and `printFluxVector(model,solution)`.

Annotations in red text and arrows point to specific parts of the output:

- An arrow points from the text "Structure used to store solution" to the `struct` keyword in the output.
- An arrow points from the text "Growth-rate" to the value `0.8739`.
- An arrow points from the text "Flux Vector" to the `x: [95x1 double]` entry.

**solution = optimizeCbModel(model);**



# Structure of Optimization Solution

The screenshot shows the MATLAB interface with the following windows:

- Current Folder:** Shows files like .tmp, AerobicGlucoseBioMass.m, and Introduction\_lecture.mlx.
- Variables - solution:** A table showing the fields and values of the 'solution' struct.

Field	Value
origStatText	[]
f	0.8739
v	95x1 double
y	72x1 double
w	95x1 double
s	72x1 double
ch solver	'gurobi'
ch algorithm	'default'
stat	1
ch origStat	'OPTIMAL'
time	0.0160
basis	1x1 struct
vars_v	[]
x	95x1 double
- Workspace:** Shows the 'model' and 'solution' variables as 1x1 structs.

Name	Value
model	1x1 struct
solution	1x1 struct

A red arrow points to the 'solution' entry with the text "Double Click".
- Command History:** Shows the commands initCobraToolbox, AerobicGlucoseBioMass, and initCobraToolbox again.

Red annotations highlight specific fields in the 'solution' struct:

- Objective Function Value: Points to the 'f' field (0.8739).
- Shadow Prices: Points to the 'v' field (95x1 double).
- Reduced Costs: Points to the 'w' field (95x1 double).
- Solver: Points to the 'ch solver' field ('gurobi').
- Flux Vector: Points to the 'x' field (95x1 double).



# Printing the Optimized Flux Solutions (Analysis/Exploration)

The screenshot shows the MATLAB environment with the following windows:

- Current Folder:** Displays a list of files including various metabolic models (e.g., Abiotrophia\_defectiva, Acidaminococcus, Acinetobacter, AntCore, cardiac\_mit\_glcup, ecoli\_core\_model, iAF1260, iJO1366, ME\_matrix\_GlcAer, modelReg, Recon1.0, Recon2.0, Recon2.v04) and command history.
- Command Window:** Shows the MATLAB command `>> printFluxVector(model, solution.x, true)` and its output. The output lists flux values for various reactions. A red box highlights the reaction `EX_co2(e)`, which has a flux value of 22.81. Red arrows point from this value to the text "Reaction Name" and "Flux Passing through the Reaction (mmols g<sub>DW</sub><sup>-1</sup>h<sup>-1</sup>)". Another red arrow points from the value 0.8739 to the text "Growth-rate".
- Workspace:** Shows variables `model` and `solution` in the workspace.
- Command History:** Shows the commands used to load the model and calculate the solution.

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

Reaction	Flux (mmols g <sub>DW</sub> <sup>-1</sup> h <sup>-1</sup> )
EX_co2(e)	22.81
EX_glc(e)	-10
EX_h2o(e)	29.18
EX_h(e)	17.53
EX_nh4(e)	-4.765
EX_o2(e)	-21.8
EX_pi(e)	-3.215
FBA	7.477
FUM	5.064
G6PDH2r	4.96
GAPD	
GLCpt	
GLNS	



# 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)



# Exploring Reaction & Metabolite Vectors in Matlab

The figure shows four Matlab windows illustrating the exploration of metabolic models:

- Variables - model**: Shows the structure of the variable `model`, which is a `1x1 struct` with 28 fields. Fields include `S` (72x95 sparse double), `mets` (72x1 cell), `b` (72x1 double), `ch` (72x1 char), `rxns` (95x1 cell), `lb` (95x1 double), `ub` (95x1 double), `c` (95x1 double), `ch` (osenseStr: 'max'), `genes` (137x1 cell), `rules` (95x1 cell), `metCharges` (72x1 int32), `metFormulas` (72x1 cell), `metNames` (72x1 cell), `metInchiString` (72x1 cell), `metKEGGID` (72x1 cell), and `metChEBIID` (72x1 cell). Annotations point to `mets` as the "Metabolite Vector" and `rxns` as the "Reaction Vector".
- Workspace**: Shows the workspace with variables `ans`, `FBAsolution`, `map`, `model`, and `options`. A red arrow points to the `model` variable with the text "Double Click".
- Reaction Vector**: Shows the variable `model.rxns` as a 1x1 struct containing a list of 15 reactions: ACALD, ACALDt, ACKr, ACONTa, ACONTb, ACt2r, ADK1, AKGDH, AKGt2r, ALCD2x, ATPM, ATPS4r, Biomass\_Ecoli\_core\_N(w/GAM)-Nmet2, CO2t, and CS.
- Metabolite Vector**: Shows the variable `model.mets` as a 1x1 struct containing a list of 15 metabolites: 13dpg[c], 2pg[c], 3pg[c], 6pgc[c], 6pgl[c], ac[c], ac[e], acald[c], acald[e], accoa[c], acon-C[c], actp[c], adp[c], akg[c], and akn[e].



# 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', 0, 0);
```

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**

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



# "AerobicGlucoseBioMass.m" Matlab Script

```
clear; % Clear Workspace  
  
load('ecoli_core_model.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 =  
  
  struct with fields:  
  
    cont: [169x1 double]  
    int: [74x1 double]  
    full: [243x1 double]  
    obj: 1.6531  
    solver: 'gurobi'  
    stat: 1  
origStat: 'OPTIMAL'  
time: 0.0550  
f: 1.6531  
x: [95x1 double]  
v: [95x1 double]
```

Document Written	
ACONTa	10.37
ACONTb	10.37
AKGDH	8.582
ATPM	8.39
ATPS4r	80.61
Biomass_Ecoli_core_N(w/GAM)-Nmet2	
CO2t	-40.65
CS	10.37
CYTBD	77.48

"optimizeCbModel"  
Output

Growth-rate

1.653

ENO	26.84	PIt2r	6.081
EX_co2(e)	40.65	PPC	4.737
EX_glc(e)	-18.5	PYK	2.744
EX_h2o(e)	52.69	RPE	5.399
EX_h(e)	33.16	RPI	-4.482
EX_nh4(e)	-9.014	SUCDi	8.582
EX_o2(e)	-38.74	SUCOAS	-8.582
EX_pi(e)	-6.081	TALA	2.998
FBA	13.56	TKT1	2.998
FUM	8.582	TKT2	2.401
G6PDH2r	9.881	TPI	13.56
GAPD	29.31		
GLCpts	18.5		
GLNS	0.4227		
GLUDy	-8.591		
GND	9.881		
H2Ot			
ICDHyr	-52.69		
MDH	10.37	Reactions	
NADH16	8.582		Flux
NH4t	68.9		
O2t	9.014		
PDH	38.74		
PFK	16.56		
PGI	13.56		
PGK	8.28		
PGL	-29.31		
PGM	9.881		
	-26.84		



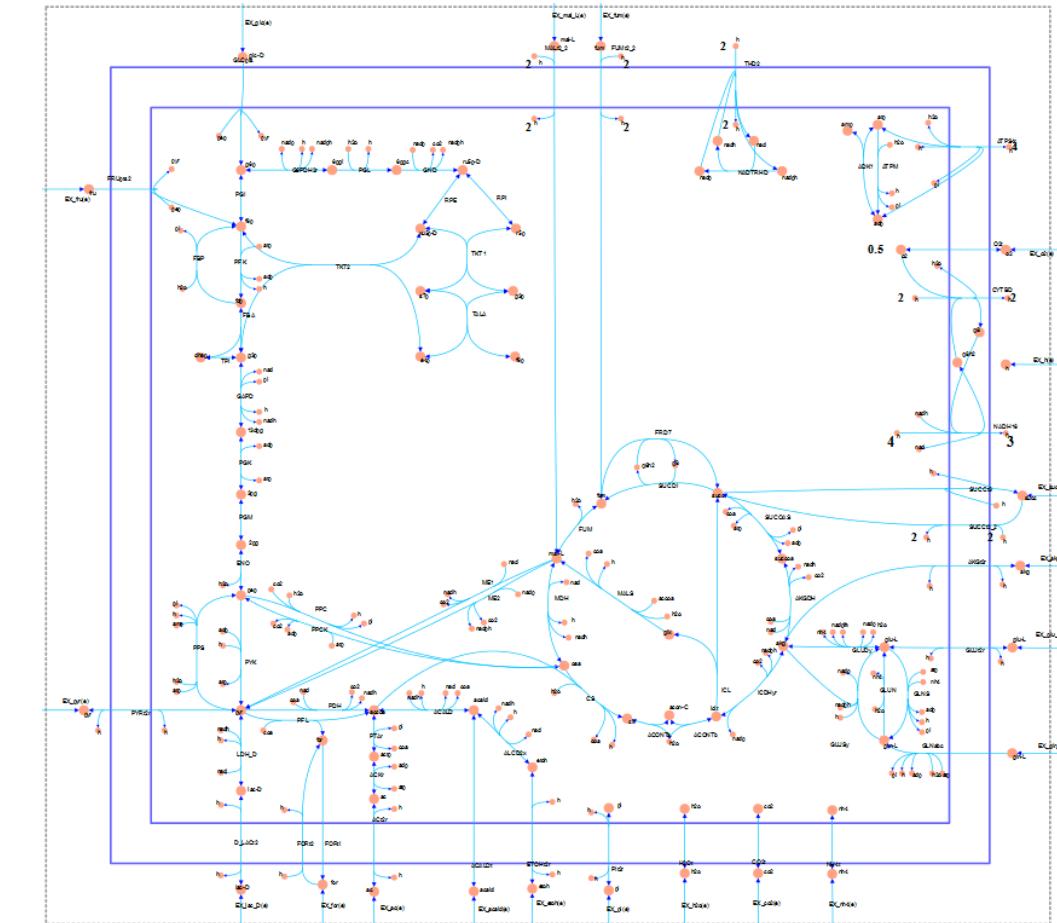
# Drawing Legacy COBRA Maps

- 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

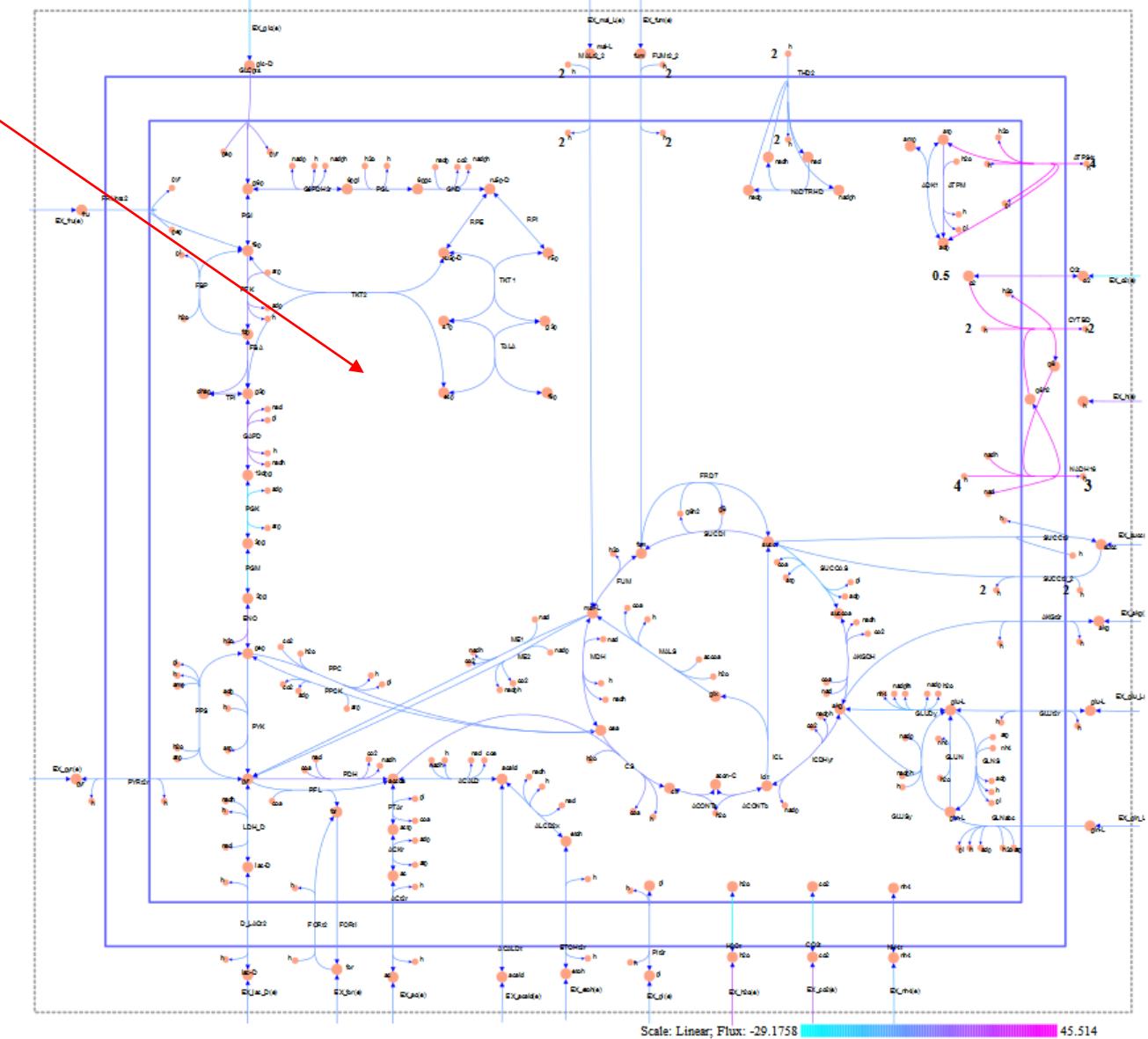


Flux values are stored on a map located in the file named "target.svg" that should be located in the current active directory

## Drawing Flux Values onto a Legacy COBRA Map

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

Exported Map      COBRA model      Flux Vector



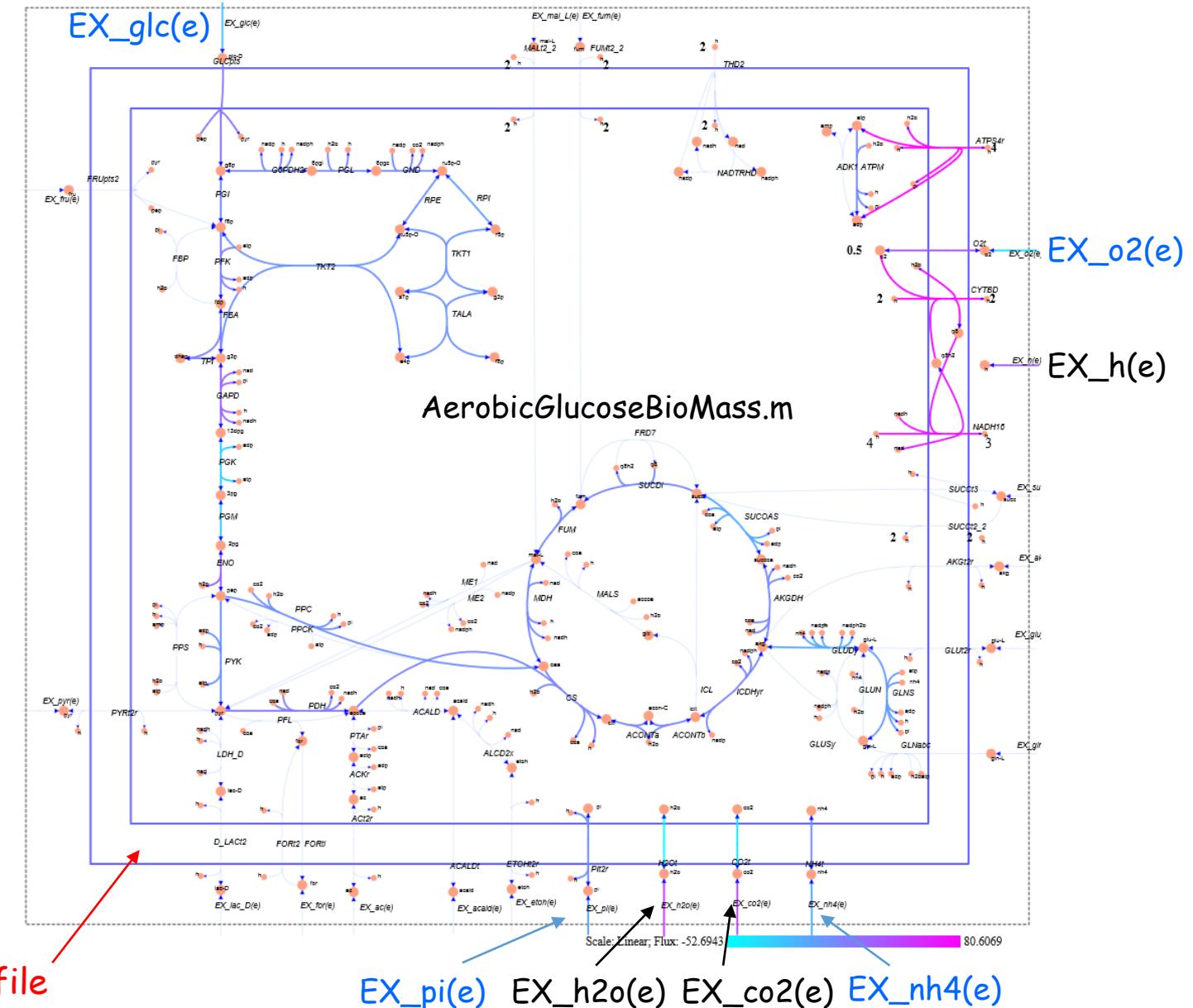


# Aerobic Growth on Glucose (Legacy Map)

## 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

Target.svg file





# 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 oxygen (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', 0, 0);
```

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*

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



# 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

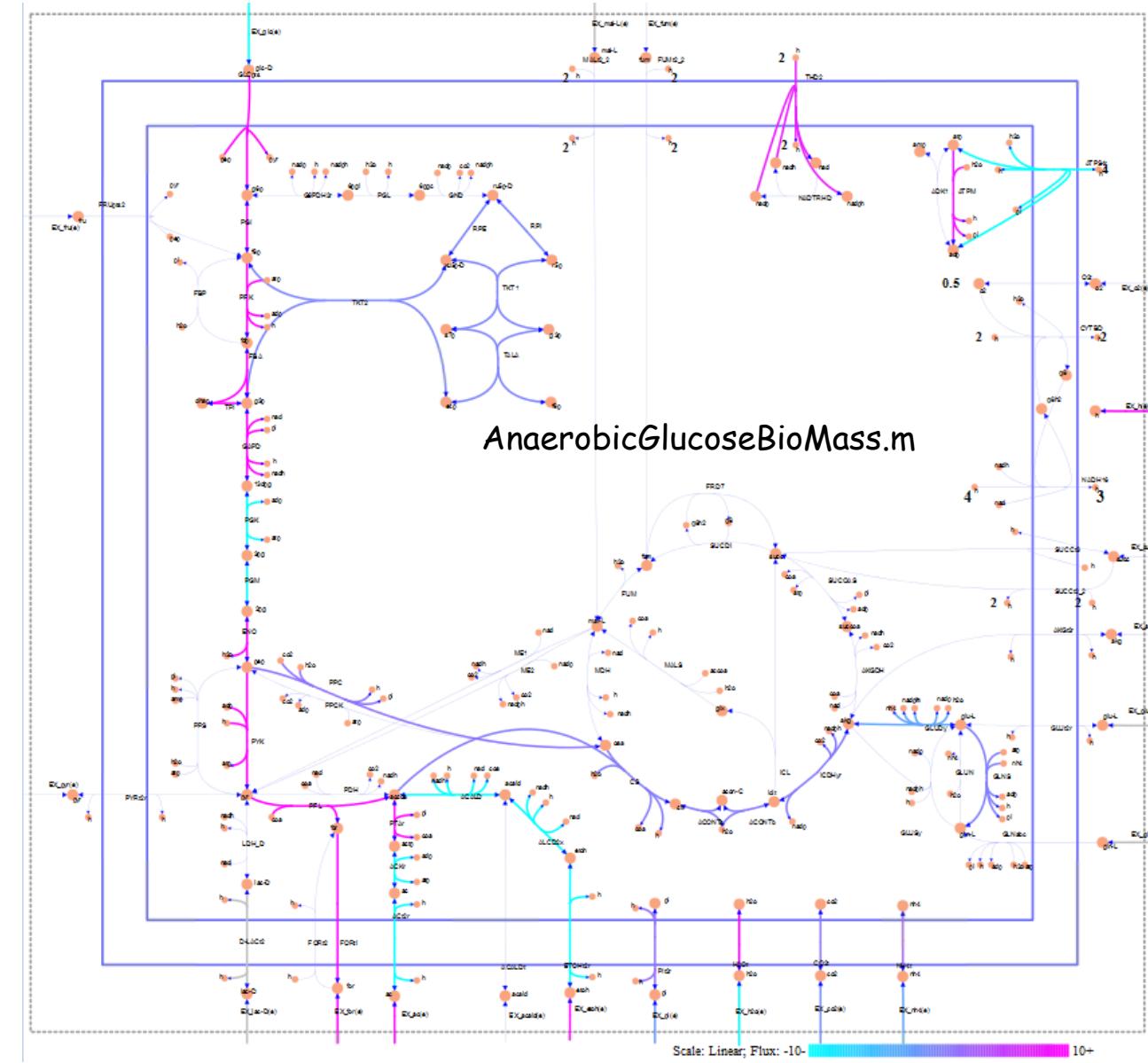
EX\_glc(e) -18.5

EX\_h2o(e) -12.0879

EX\_h(e) 56.7321

EX\_nh4(e) -2.5659

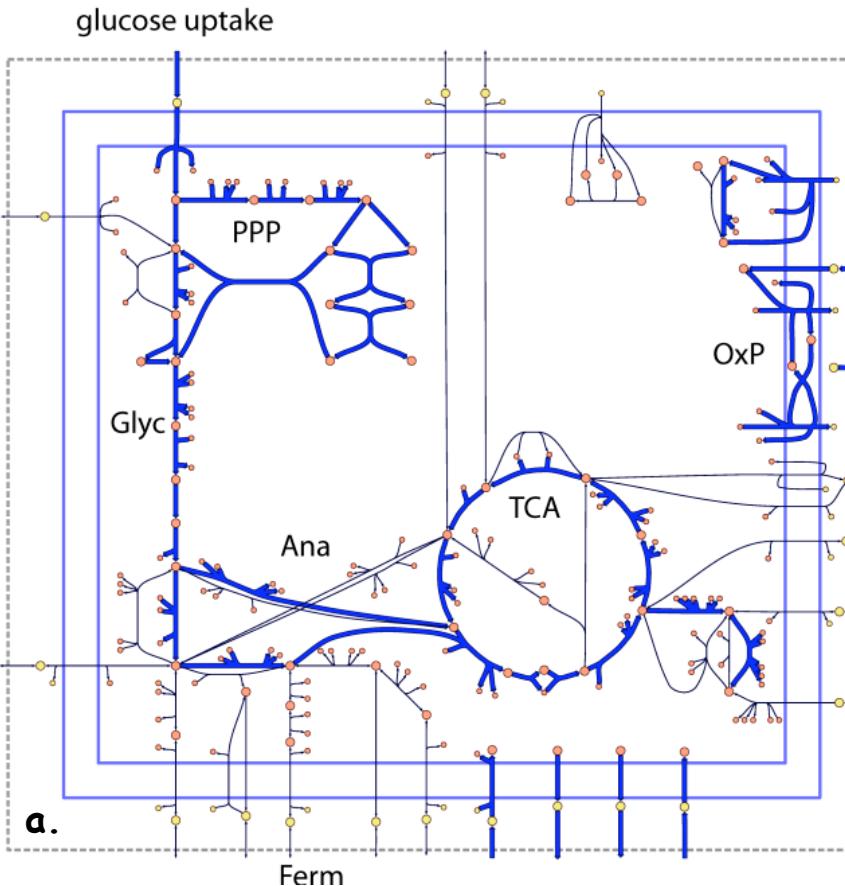
EX\_pi(e) -1.73107



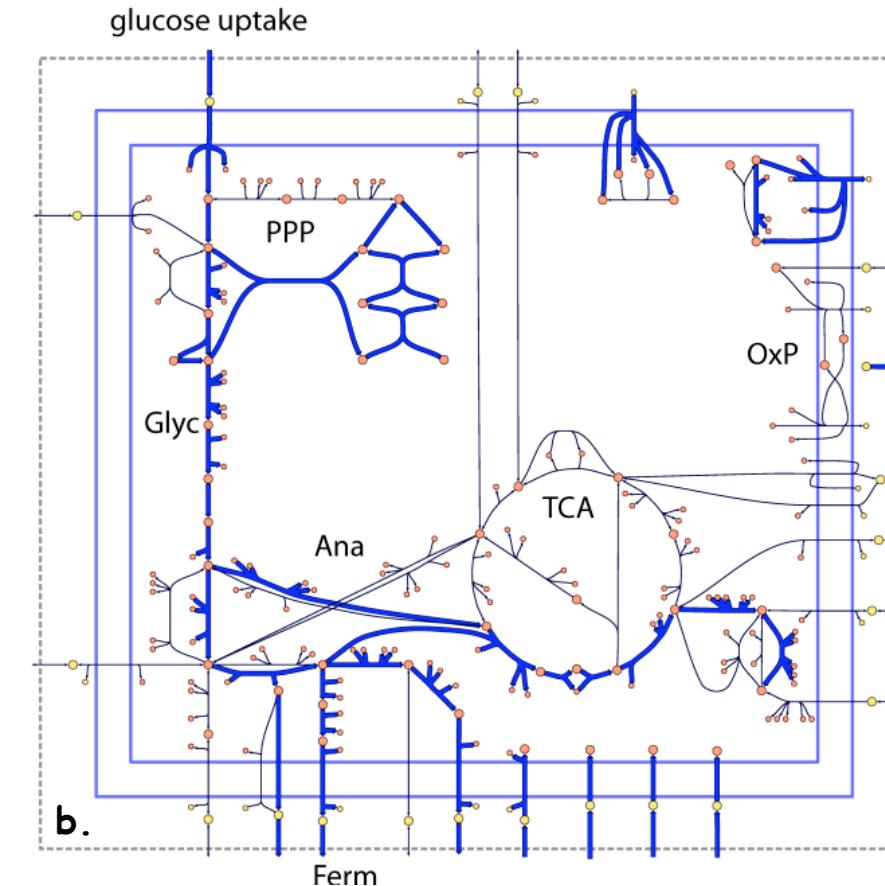


# Aerobic vs. Anaerobic Growth

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', 0, 0);
```

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

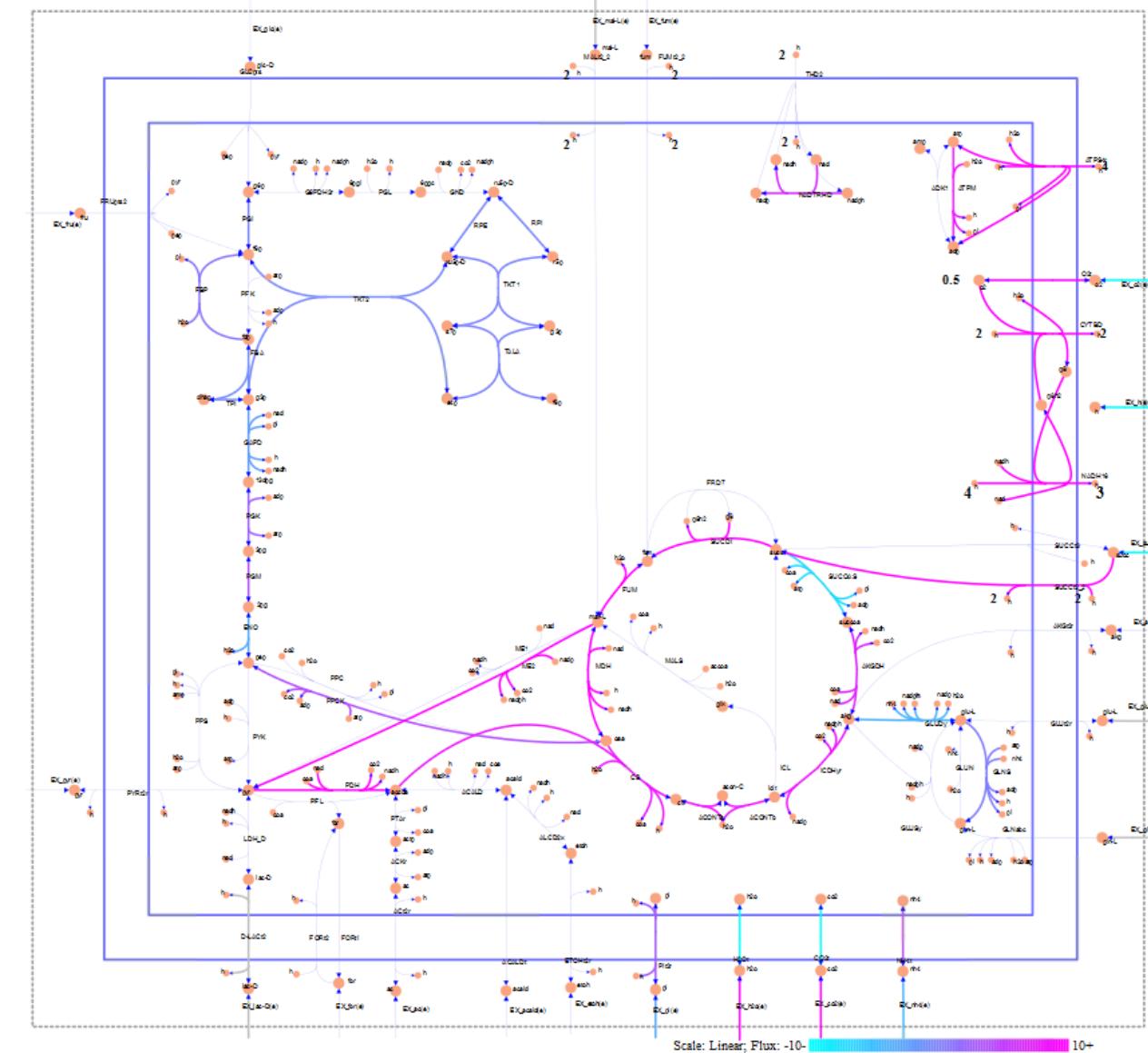
Growth can also be simulated under anaerobic conditions with any substrate by using changeRxnBounds to set the lower bound of the oxygen exchange reaction (EX\_o2(e)) to 0 mmol gDW $^{-1}$  hr $^{-1}$ , so no oxygen can enter the system. When this constraint is applied and succinate is the only organic 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.

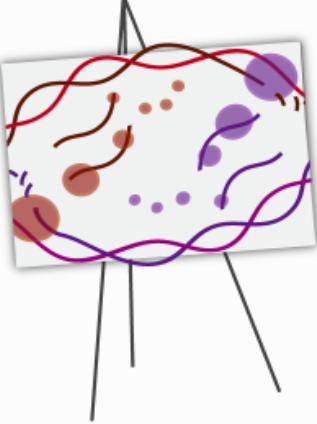
("What is flux balance analysis? - Supplementary tutorial")

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



# Using Escher Maps

[GitHub](#) [Docs](#) [What's new?](#)



# ESCHER

Build, share, and embed visualizations of metabolic pathways

Filter by organism

All

Map Model (Optional) Tool

Core metabolism (e\_coli\_core)

e\_coli\_core

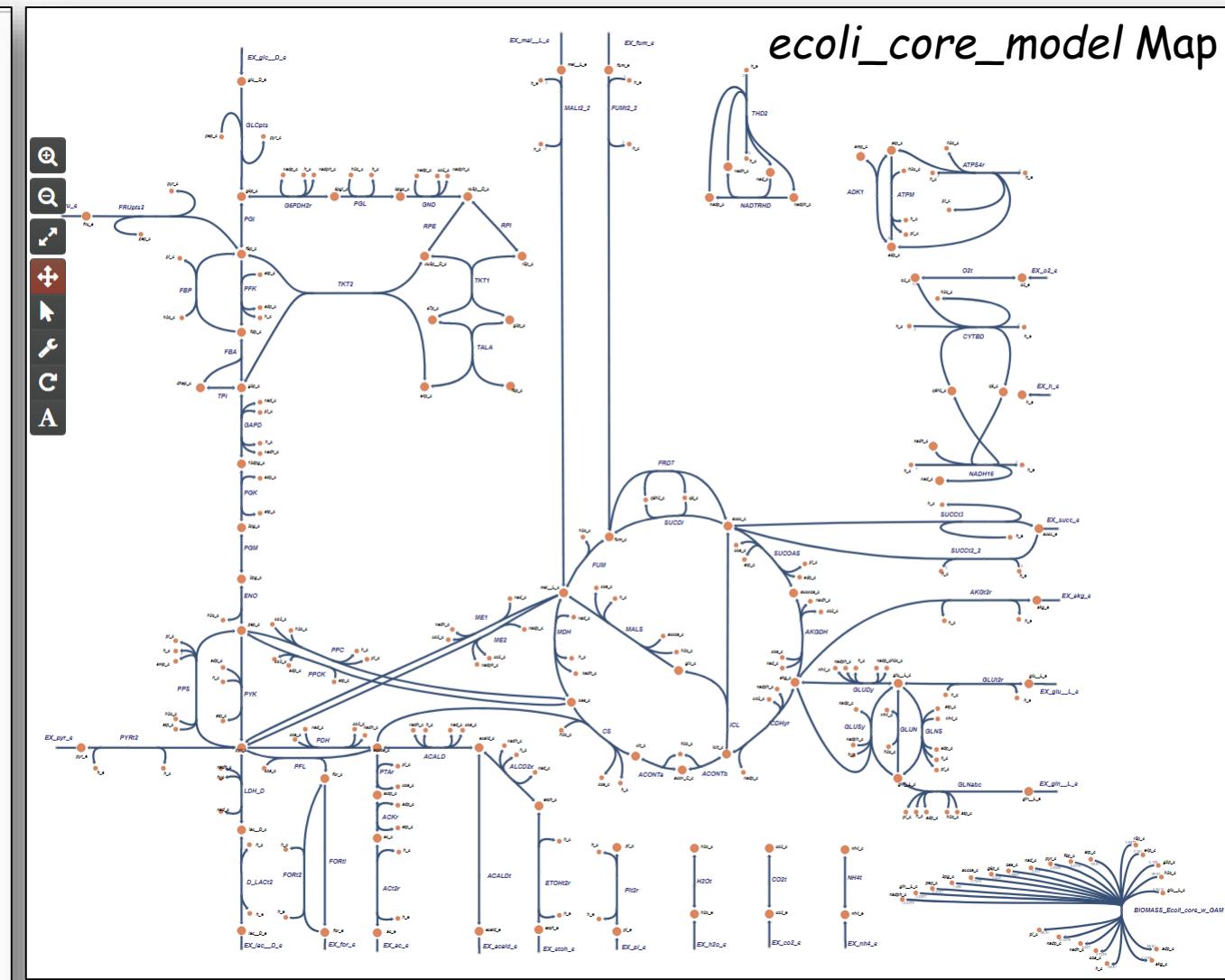
Builder

Options

Scroll to zoom (instead of scroll to pan)

Never ask before reloading

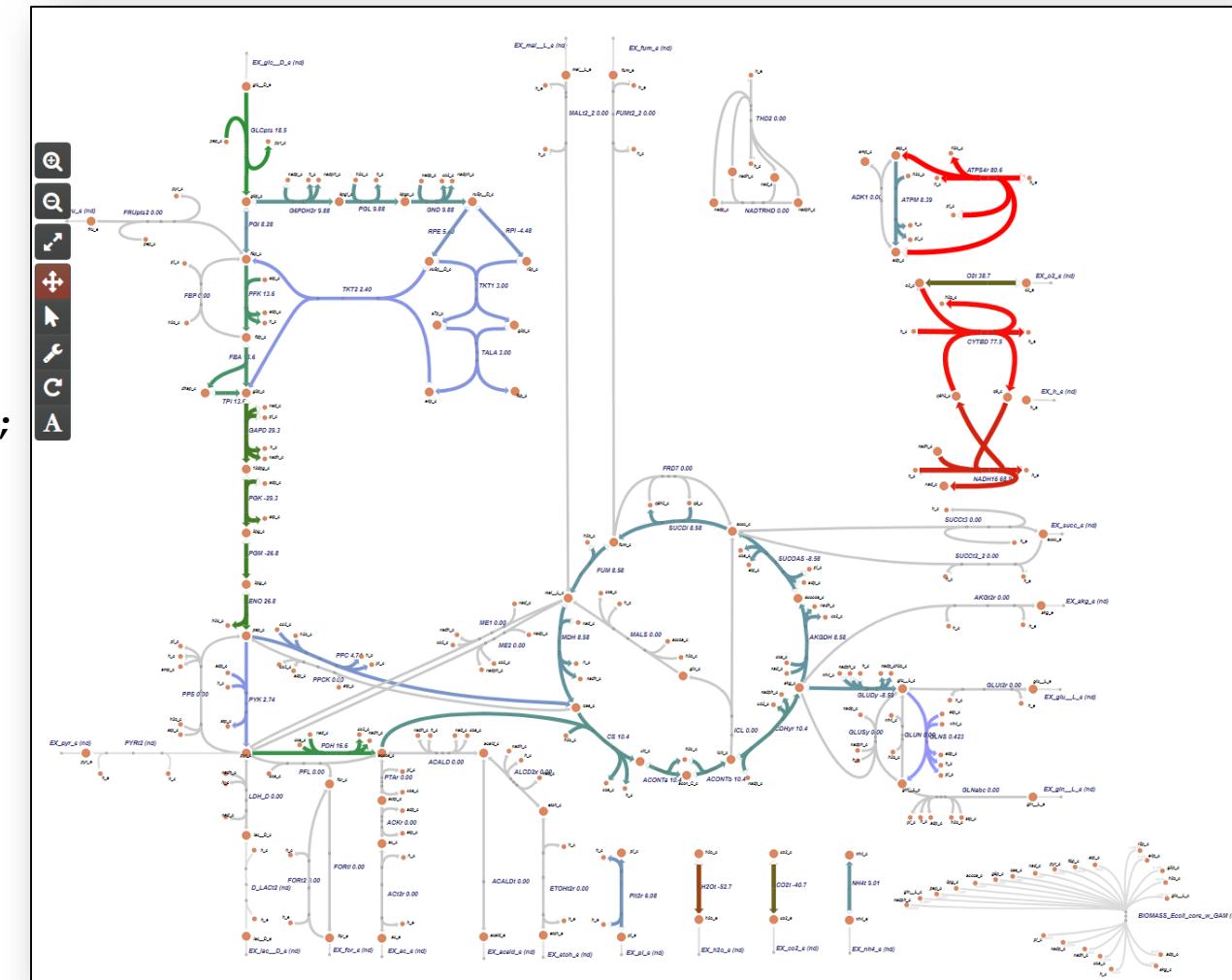
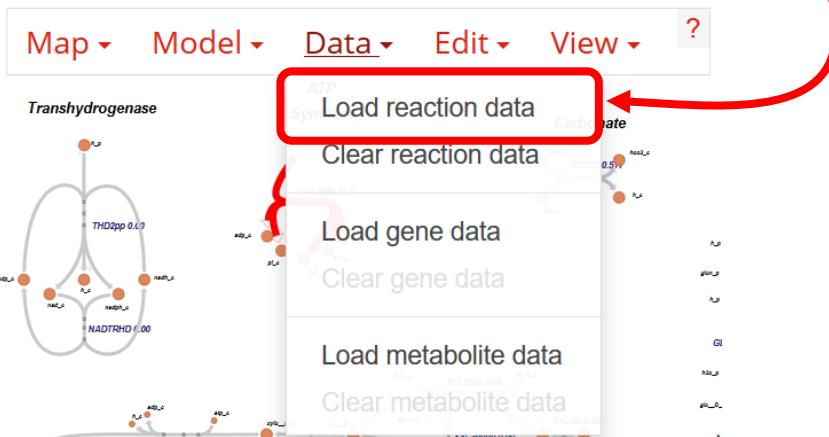
Load map





# Using Escher to Plot Flux Values

```
% Create table of flux values  
  
Reactions = model.rxns;  
  
Flux = round(FBAsolution.x,3);  
  
T = table(Reactions,Flux,'RowNames',model.rxns);  
  
% Write to CSV file  
  
writetable(T,'escher_flux.csv');
```



Zachary A. King, Andreas Dräger, Ali Ebrahim, Nikolaus Sonnenschein, Nathan E. Lewis, and Bernhard O. Palsson (2015) Escher: A web application for building, sharing, and embedding data-rich visualizations of biological pathways, PLOS Computational Biology 11(8): e1004321.



# "AerobicGlucoseBioMassEscher.m" with Escher Map Output

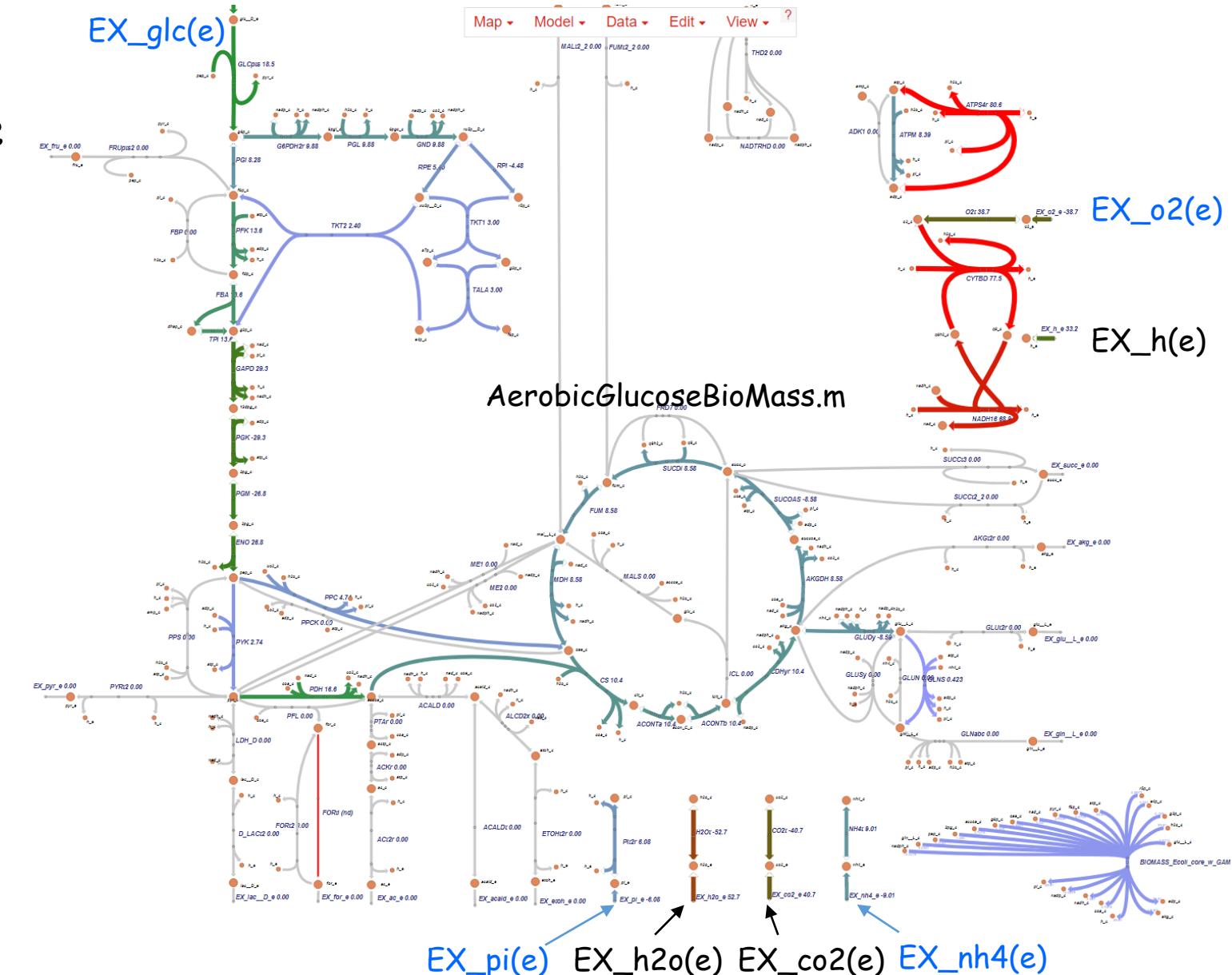
```
clear; % Clear Workspace  
  
Model = readCbModel('e_coli_core.mat'); % Load textbook model from BIGG database  
  
model = changeRxnBounds(model, 'EX_glc_D_e', -18.5, 'l'); % Set lower bound of glucose  
model = changeRxnBounds(model, 'EX_o2_e', -1000, 'l'); % Set lower bound of oxygen  
model = changeObjective(model, 'BIOMASS_Ecoli_core_w_GAM'); % Set objective function (Biomass)  
  
FBAsolution = optimizeCbModel(model, 'max', 0, 0) % Find optimized flux values  
  
  
Reactions = model.rxnns; % Create table of flux values  
Flux = round(solution.x, 3);  
T = table(Reactions, Flux, 'RowNames', model.rxnns);  
writetable(T, 'escher_flux.csv'); % Write to CSV file that can be read by Escher  
  
printFluxVector(model, FBAsolution.x, true); % Print flux values
```



# Aerobic Growth on Glucose (Escher Map)

## 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





# COBRA Toolbox v.3.0

- COBRA Toolbox v.3.0 Overview
- Base Functions
- Analysis Functions
- Data Integration Functions
- Design Functions
- Reconstruction Functions
- Visualization Functions
- COBRA Toolbox Examples



<https://opencobra.github.io/cobratoolbox/stable/index.html>



# 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 purpose of the BIGG Database?
- How do you plot flux on Escher models?
- How do you plot flux on legacy COBRA models?
- What is the purpose of the COBRA v.3.0 tutorials?

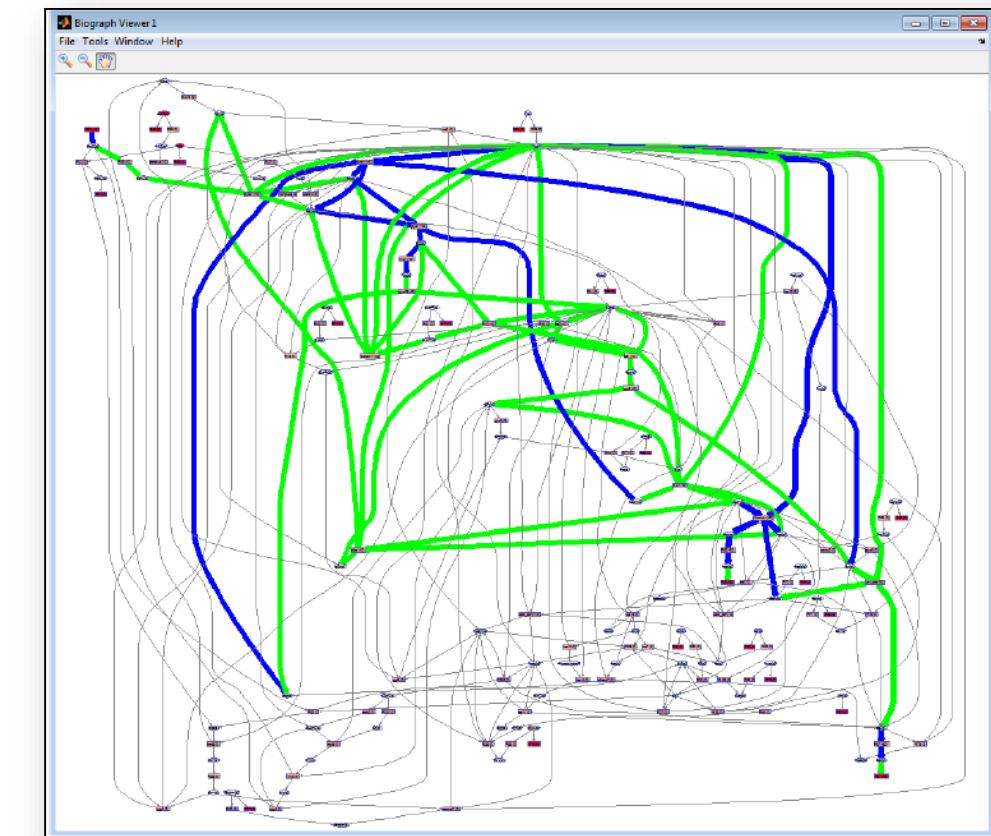


# Appendix



# Paint4Net

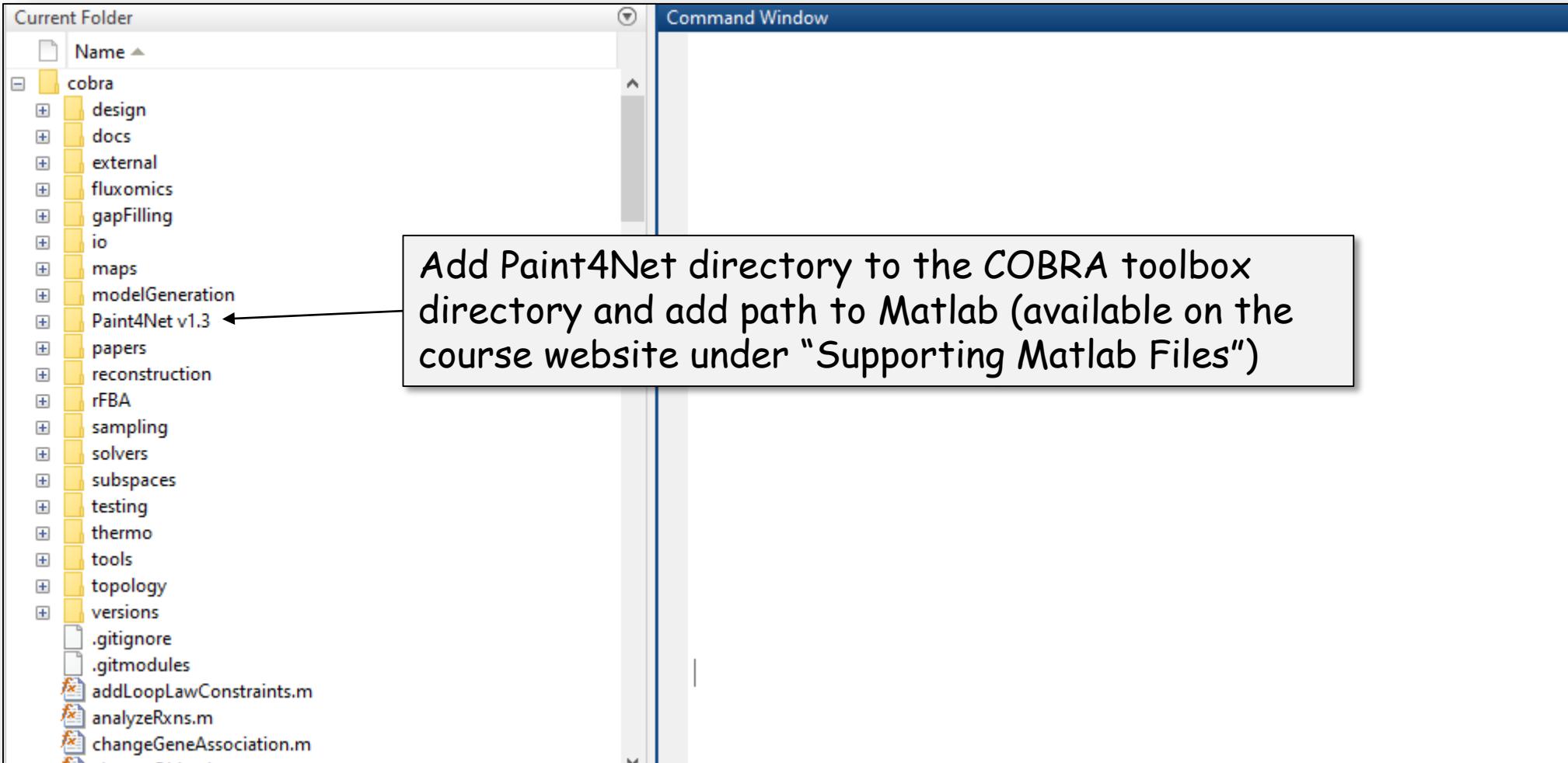
- Developed by Andrejs Kostromins
- Paint4Net is the COBRA Toolbox extension for visualization of constraints-based reconstruction and analysis (COBRA) models and reconstructions in the MATLAB environment.
- Uses the Bioinformatics toolbox to visualize COBRA models and reconstructions as a hypergraph.
- 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*.



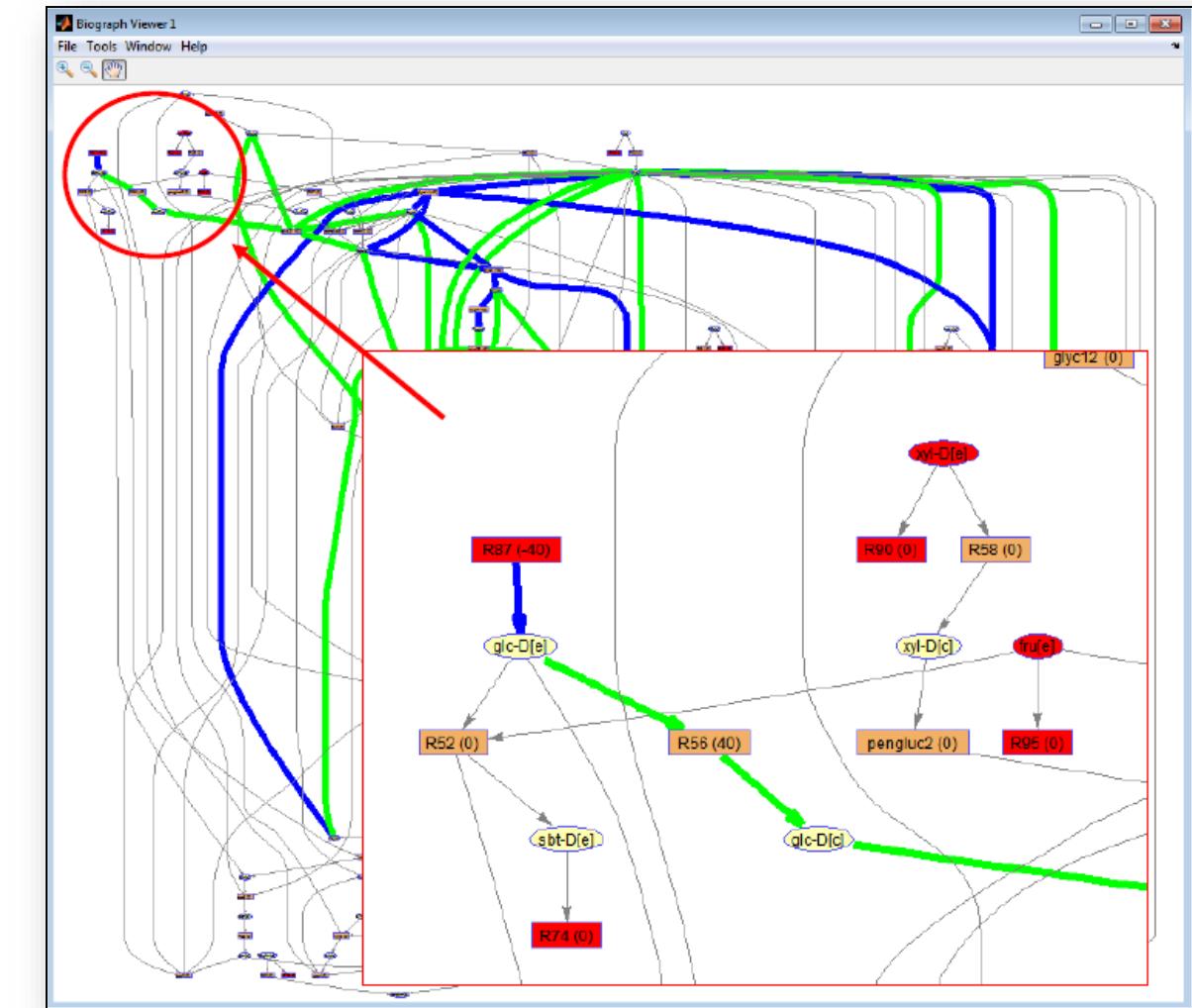
# Integrating Paint4Net with the COBRA Toolbox





# “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);

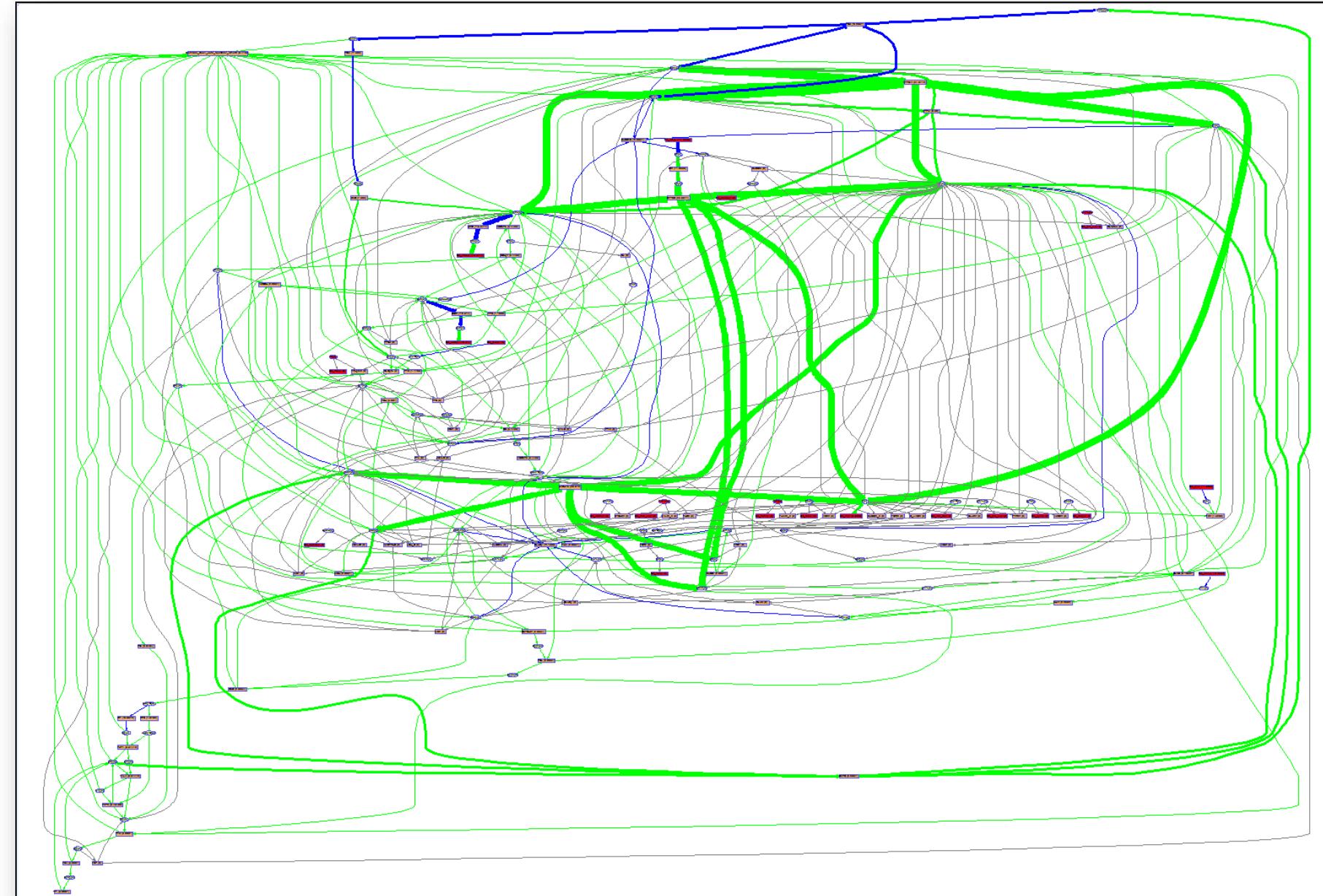
cofactors =
{'amp[c]','atp[c]','adp[c]','pi[c]','nad[c]','nadh[c]','nadph[c]','nadp[c]','h[c]','h2o[c]','co2[c]'

% 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);
```

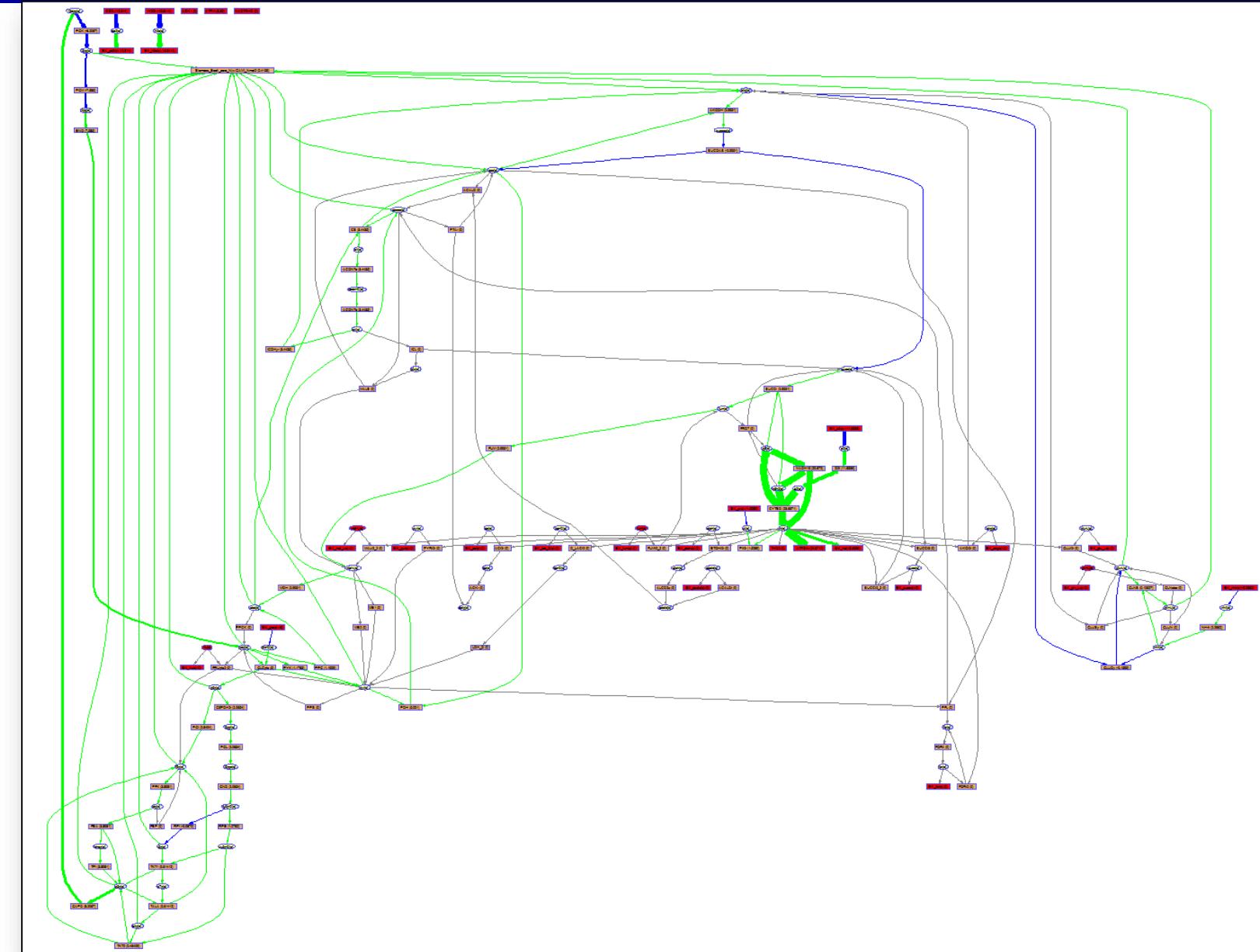


## Textbook Model With Cofactors





## 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]'}

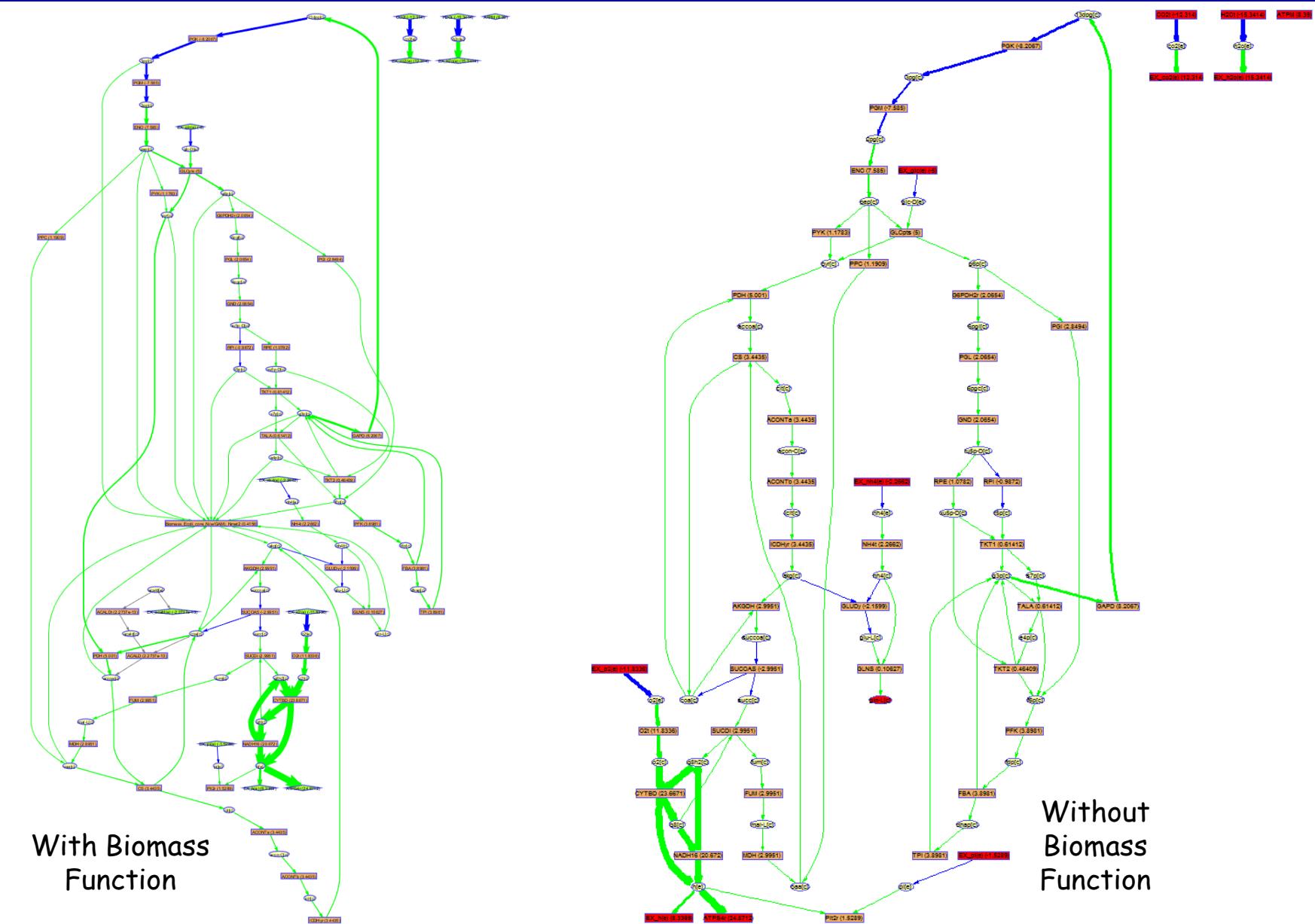
[involvedMets,deadEnds]= draw_by_rxn (model,fluxReactions,'true','struc',{''},cofactors,FBAsolution.x);
```



# 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'};

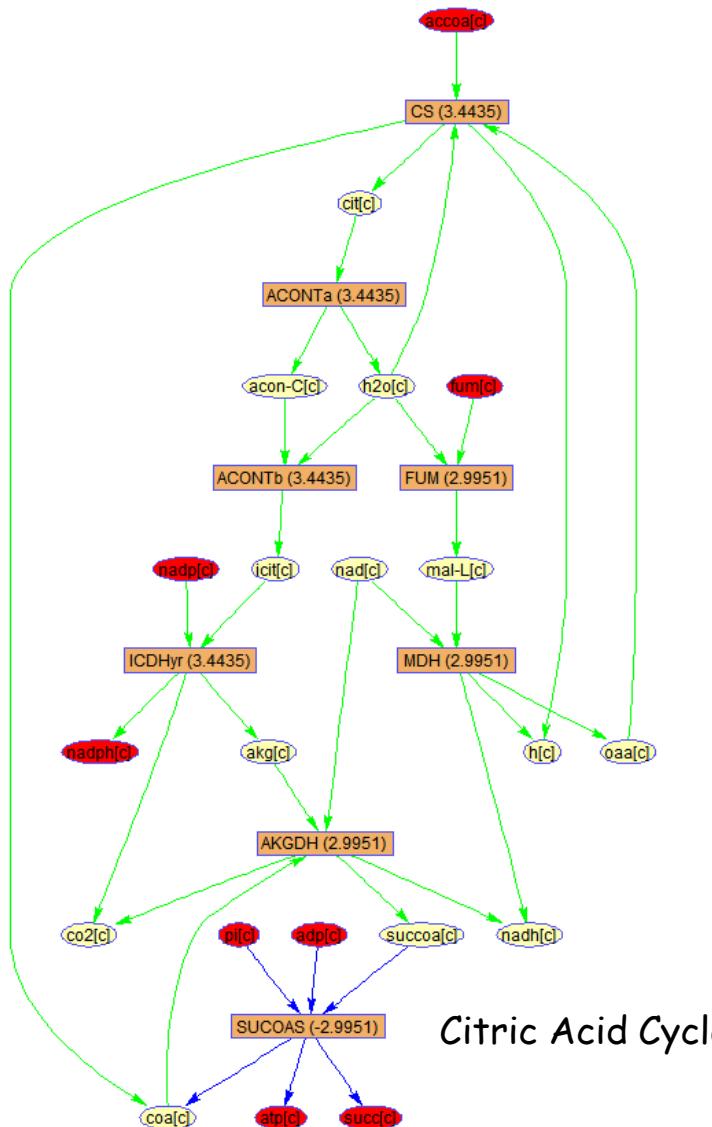
fluxReactions = model.rxn(ismember(model.subSystems,includedSubSystems));

% Plot multiple subsystem

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

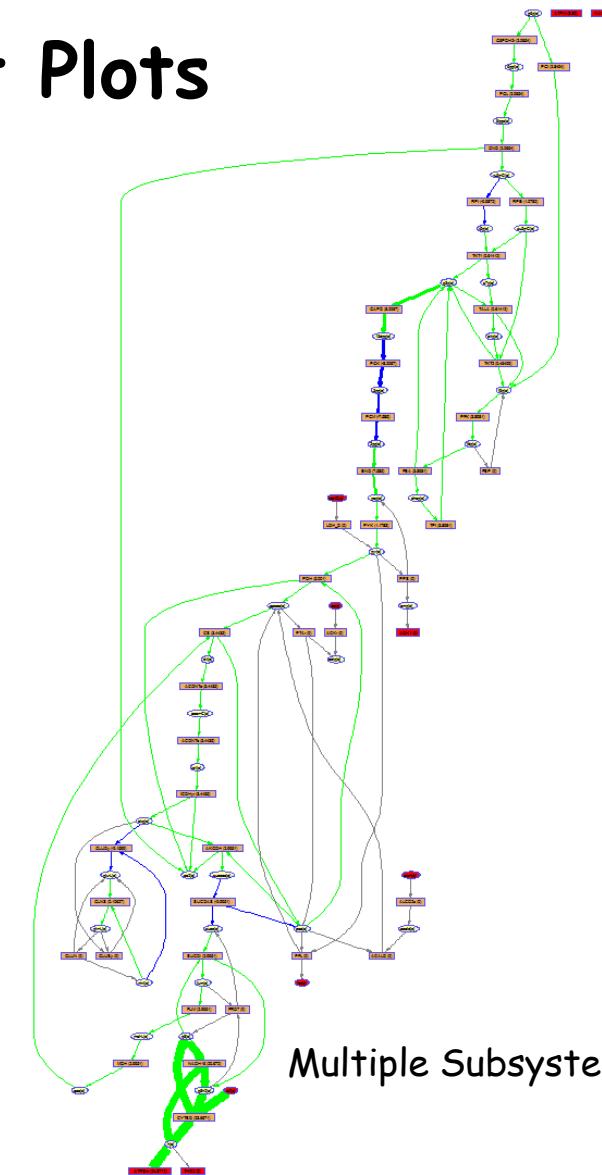


# Paint4Net Plots



Citric Acid Cycle

Paint4Net\_core\_subSystem.m

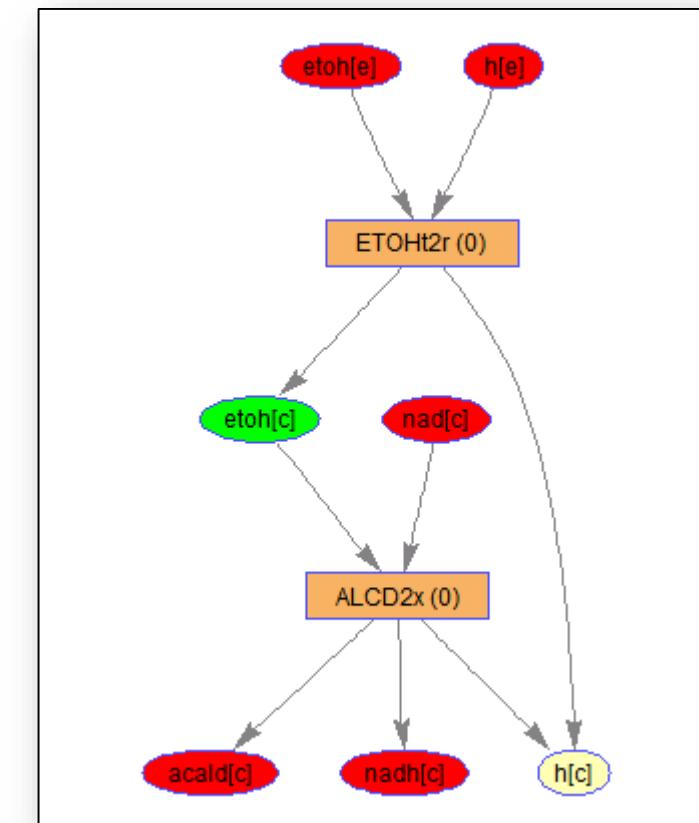


Multiple Subsystems



# 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);

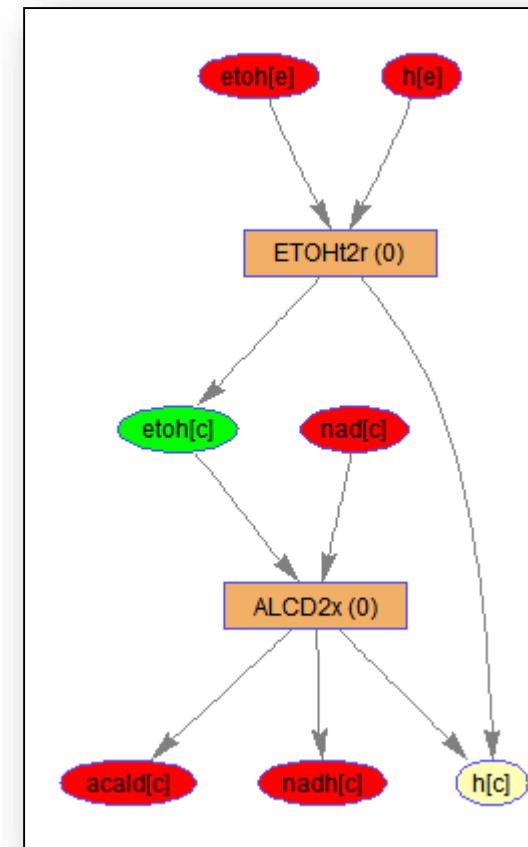
% Plot connectivity to 'etoh[c]', remove cofactors. Radius = 1

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

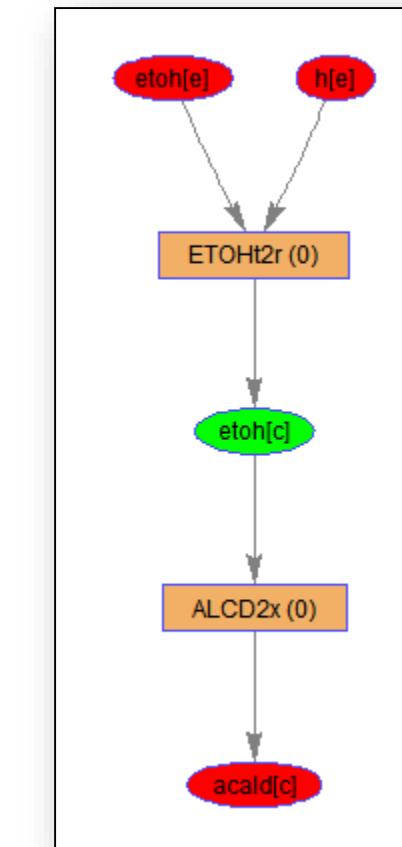


# 'etoh[c]' Connectivity (Radius = 1)

VisualizeMets\_Textbook.m



With Cofactors

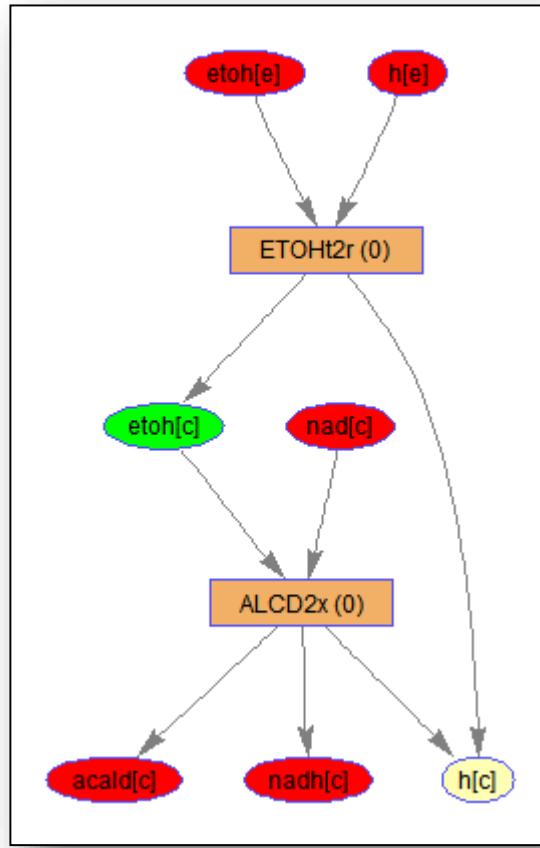


Without Cofactors

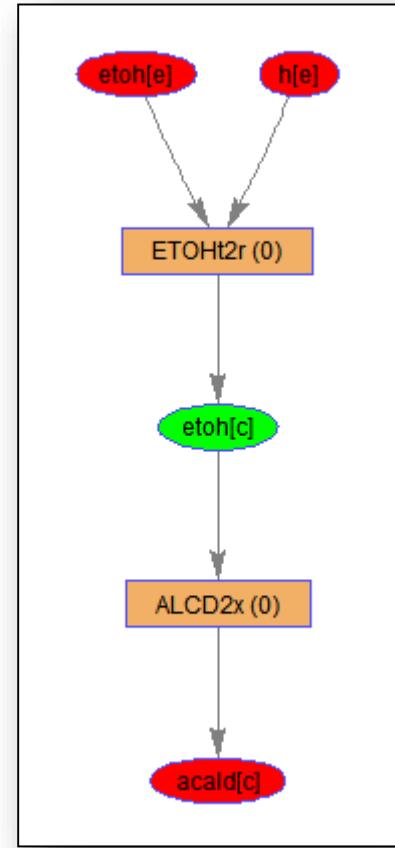


## 'etoh[c]' Connectivity (Radius = 1)

VisualizeMets\_Textbook.m



With Cofactors



Without Cofactors

