Flux Variability Analysis & Parsimonious Flux Balance Analysis

Learning Objectives

- Explain alternate optimal solutions,
- Explain flux variability analysis,
- Explain parsimonious flux balance analysis.

Lesson Outline

- Alternate Optimal Solutions
- Flux Variability Analysis
- Parsimonious FBA

Phenotypes

- Phenotype = A phenotype (from Greek phainein, meaning "to show", and typos, meaning "type") is the composite of an organism's **observable** characteristics or traits, such as its morphology, development, biochemical or physiological properties, phenology, behavior, and products of behavior.
- Silent phenotypes have the same overall cellular function but are based on different underlying reaction networks.

https://en.wikipedia.org/wiki/Phenotype

Alternate Equivalent Optimal Solutions

- The flux distributions calculated by FBA are often not unique. In many cases, it is necessary for a biological system to achieve the same objective value by using alternate equivalent optimal pathways, creating phenotypically different alternate optimal solutions (silent phenotypes).
- Requires the Mixed Integer Linear Programming (MILP) solver
- For large models there can be a very large number of alternate equivalent optimal solutions.

Same objective value for all alternate flux vectors

Maximize the objective function

$$Z = \sum_i c_i v_i^k = \mathbf{c} \cdot \mathbf{v}^k$$

with the following constraints

$$\frac{d\mathbf{x}}{dt} = \mathbf{S} \cdot \mathbf{v}^k = \mathbf{0}$$

$$\alpha_j \le v_j^k \le \beta_j$$

$$1 \le k \le n$$

Up to n alternate flux vectors

$\Rightarrow$ $v^1, v^2, \ldots, v^n$ all have the same value of Z

Reed, J. L. & Palsson, B. Ø. Genome-scale in silico models of *E. coli* have multiple equivalent phenotypic states: assessment of correlated reaction subsets that comprise network states. Genome Res. 14, 1797–1805 (2004).

Identifying Alternate Equivalent Optimal Solutions

- A function that that is provided by the Cobra Toolbox to identify alternate equivalent optimal solutions is called

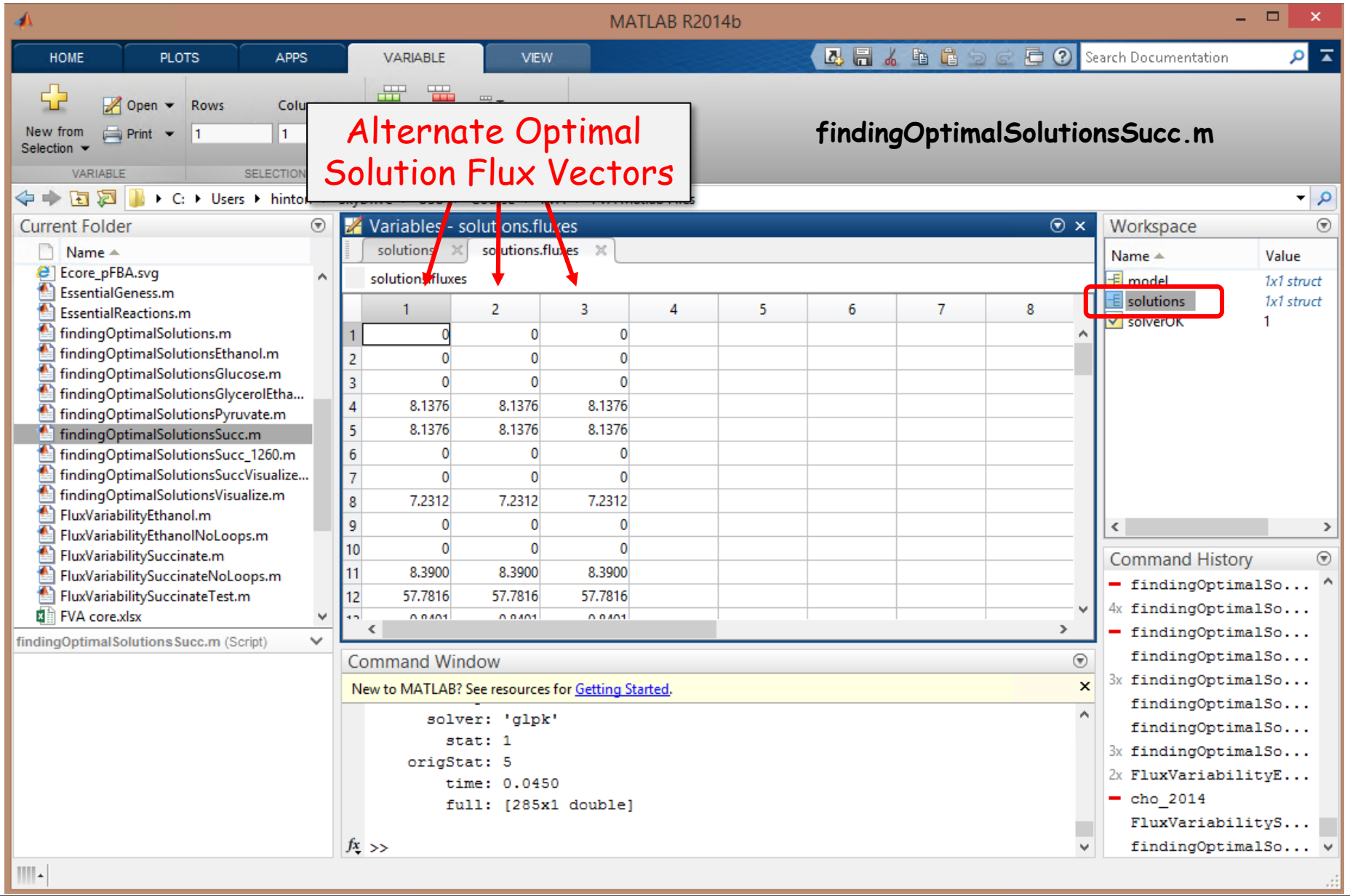
enumerateOptimalSolutions(model)

- In Matlab workspace a new structure called "solutions" is created that contains all the alternate equivalent optimal solutions.
- For large models, this computation will take a long time.

```
% findingOptimalSolutionsSucc.m

clear;

model = readCbModel('ecoli_core_model.mat');

model = changeRxnBounds(model,'EX_o2(e)',-40,'l');

model = changeRxnBounds(model,'EX_glc(e)',0,'l');

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

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

% List optimal solutions

changeCobraSolver('glpk','all') % gurobi can include loops

[solutions] = enumerateOptimalSolutions(model);
```

Reed, J. L. & Palsson, B. Ø. Genome-scale in silico models of E. coli have multiple equivalent phenotypic states: assessment of correlated reaction subsets that comprise network states. Genome Res. 14, 1797–1805 (2004).

Alternate Optimal Solutions Matlab Screenshot



Alternate Optimal Solution Non-zero Flux Values

Reaction	S1	S2	S3	Lower	Upper	Range
'ACONTa'	8.137641	8.137641	8.137641	8.137641	8.137641	0
'ACONTb'	8.137641	8.137641	8.137641	8.137641	8.137641	0
'AKGDH'	7.231221	7.231221	7.231221	7.231221	7.231221	0
'ATPM'	8.39	8.39	8.39	8.39	8.39	0
'ATPS4r'	57.78164	57.78164	57.78164	57.78164	57.78164	0
Biomass	0.840134	0.840134	0.840134	0.840134	0.840134	0
'CO2t'	-44.2477	-44.2477	-44.2477	-44.2477	-44.2477	0
'CS'	8.137641	8.137641	8.137641	8.137641	8.137641	0
'CYTBD'	66.55278	66.55278	66.55278	66.55278	66.55278	0
'ENO'	-3.49017	-3.49017	-3.49017	-3.49017	-3.49017	0
'EX_co2(e)'	44.24767	44.24767	44.24767	44.24767	44.24767	0
'EX_h2o(e)'	30.36754	30.36754	30.36754	30.36754	30.36754	0
'EX_h(e)'	-23.1469	-23.1469	-23.1469	-23.1469	-23.1469	0
'EX_nh4(e)'	-4.58108	-4.58108	-4.58108	-4.58108	-4.58108	0
'EX_o2(e)'	-33.2764	-33.2764	-33.2764	-33.2764	-33.2764	0
'EX_pi(e)'	-3.0906	-3.0906	-3.0906	-3.0906	-3.0906	0
'EX_succ(e)'	-20	-20	-20	-20	-20	0
'FBA'	-0.83568	-0.83568	-0.83568	-0.83568	-0.83568	0
'FBP'	0.835681	0.835681	0.835681	0.835681	0.835681	0

Reaction	S1	S2	S3	Lower	Upper	Range
'FUM'	27.23122	27.23122	27.23122	27.23122	27.23122	0
'GAPD'	-2.23333	-2.23333	-2.23333	-2.23333	-2.23333	0
'GLNS'	0.214822	0.214822	0.214822	0.214822	0.214822	0
'GLUDy'	-4.36626	-4.36626	-4.36626	-4.36626	-4.36626	0
'H2Ot'	-30.3675	-30.3675	-30.3675	-30.3675	-30.3675	0
'ICDHyr'	8.137641	8.137641	8.137641	8.137641	8.137641	0
'MDH'	13.56499	13.56499	20.05742	13.56499	20.05742	6.492425
'ME1'	0	6.492425	0	0	6.492425	6.492425
'ME2'	13.66623	7.173804	7.173804	7.173804	13.66623	6.492425
'NADH16'	39.32156	39.32156	39.32156	39.32156	39.32156	0
'NADTRHD'	6.492425	0	0	0	6.492425	6.492425
'NH4t'	4.581084	4.581084	4.581084	4.581084	4.581084	0
'O2t'	33.27639	33.27639	33.27639	33.27639	33.27639	0
'PDH'	11.2863	11.2863	11.2863	11.2863	11.2863	0
'PGI'	-0.17223	-0.17223	-0.17223	-0.17223	-0.17223	0
'PGK'	2.233329	2.233329	2.233329	2.233329	2.233329	0
'PGM'	3.490169	3.490169	3.490169	3.490169	3.490169	0
'PIt2r'	3.090602	3.090602	3.090602	3.090602	3.090602	0
'PPCK'	3.926283	3.926283	10.41871	3.926283	10.41871	6.492425
'PYK'	0	0	6.492425	0	6.492425	6.492425

AOS_Example.xlsx

Reactions with Changing Flux Identified Through Alternate Optimal Solutions for Growth on Succinate

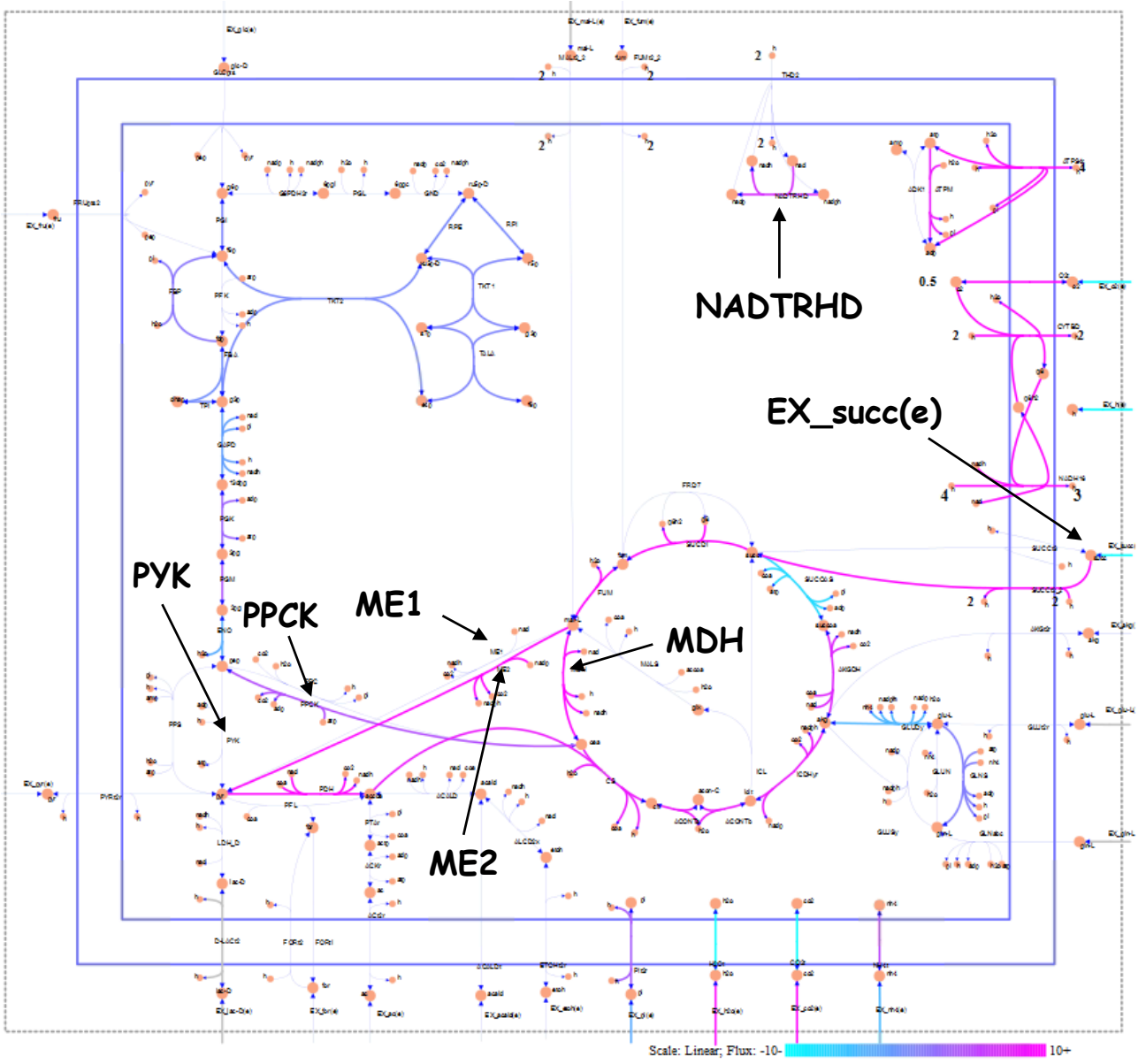
Reaction	Minimum Flux (mmol gDW^{-1} hr^{-1})	Maximum Flux (mmol gDW^{-1} hr^{-1})
MDH	13.56	20.06
ME1	0	6.49
ME2	7.17	13.67
NADTRHD	0	6.49
PPCK	3.93	10.42
PYK	0	6.49

findingOptimalSolutionsSucc.m

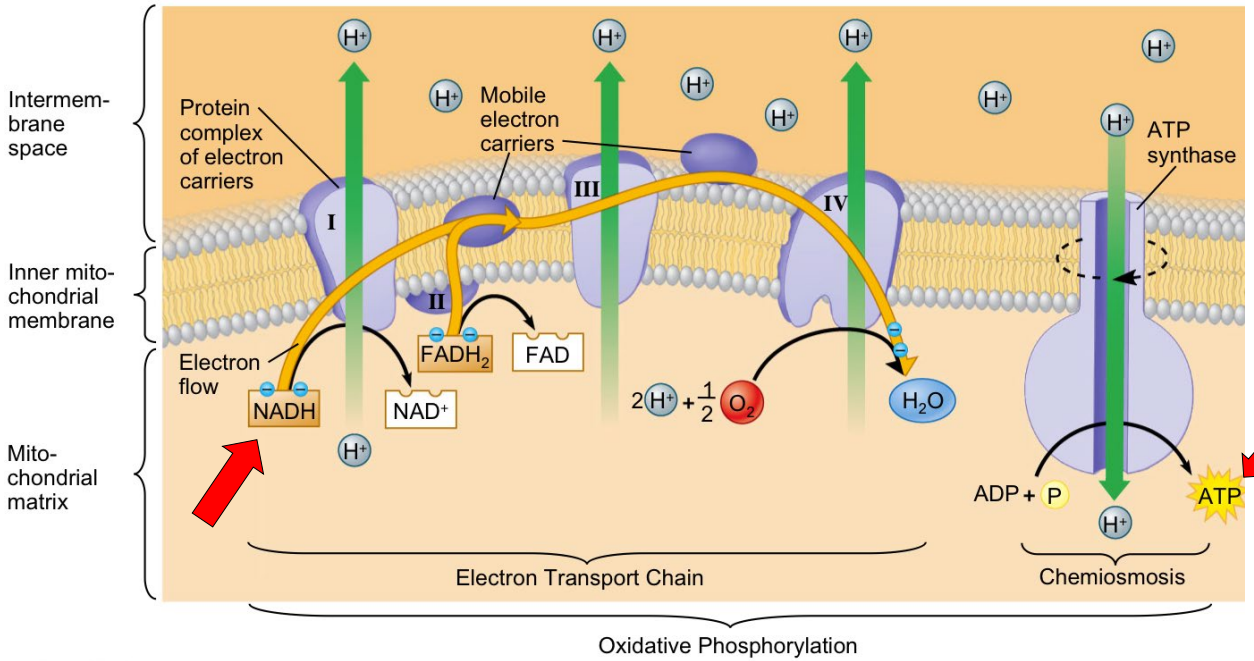
Reactions with Changing Flux Identified Through Alternate Optimal Solutions

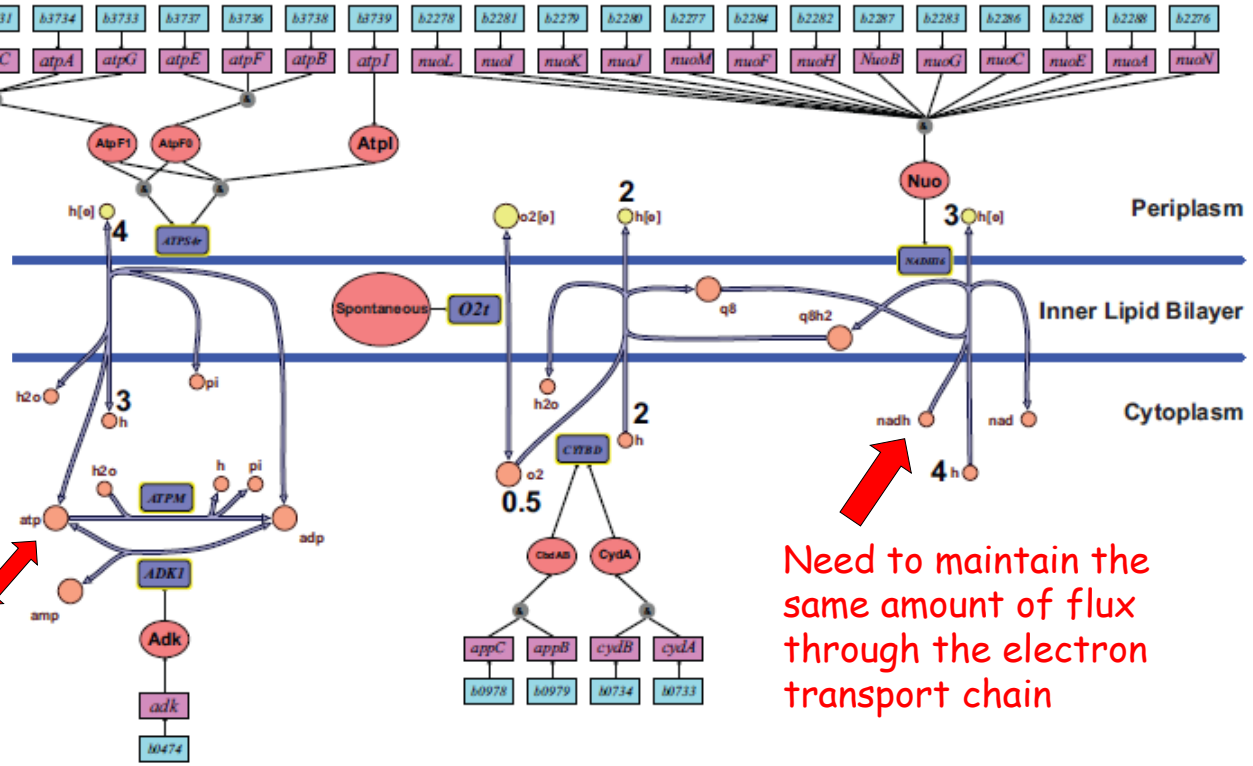
- **MDH** (malate dehydrogenase)
  - ✓ mal-L + nad <==> h + nadh + oaa
- **ME1** (malic enzyme (NAD))
  - ✓ mal-L + nad --> co2 + nadh + pyr
- **ME2** (malic enzyme (NADP))
  - ✓ mal-L + nadp --> co2 + nadph + pyr
- **NADTRHD** (NAD transhydrogenase)
  - ✓ nad + nadph --> nadh + nadp
- **PPCK** (phosphoenolpyruvate carboxykinase)
  - ✓ atp + oaa --> adp + co2 + pep
- **PYK** (pyruvate kinase)
  - ✓ adp + h + pep --> atp + pyr

All reactions are centered around energy and reducing power production



Energy Production: Electron Transport Chain

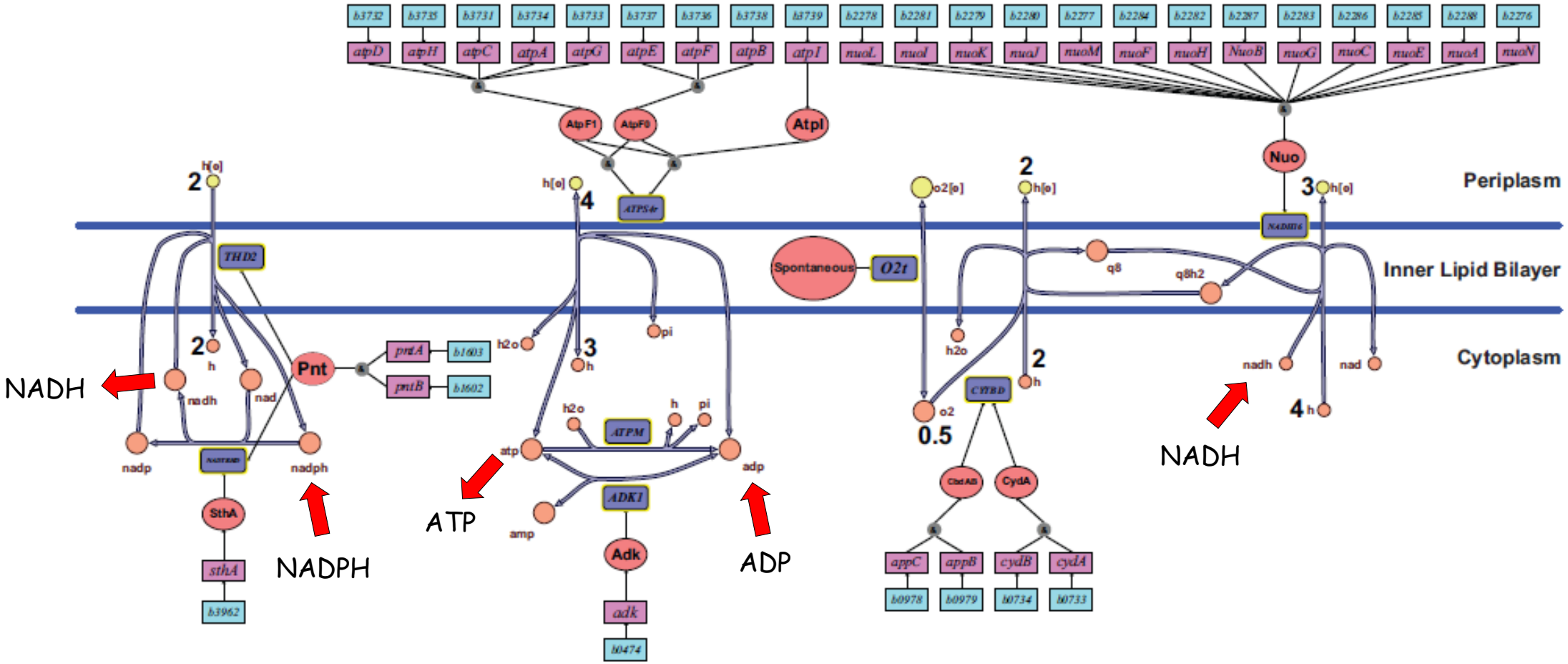




http://classconnection.s3.amazonaws.com/567/flashcards/203567/png/citric_acid_cycle1315513581579.png

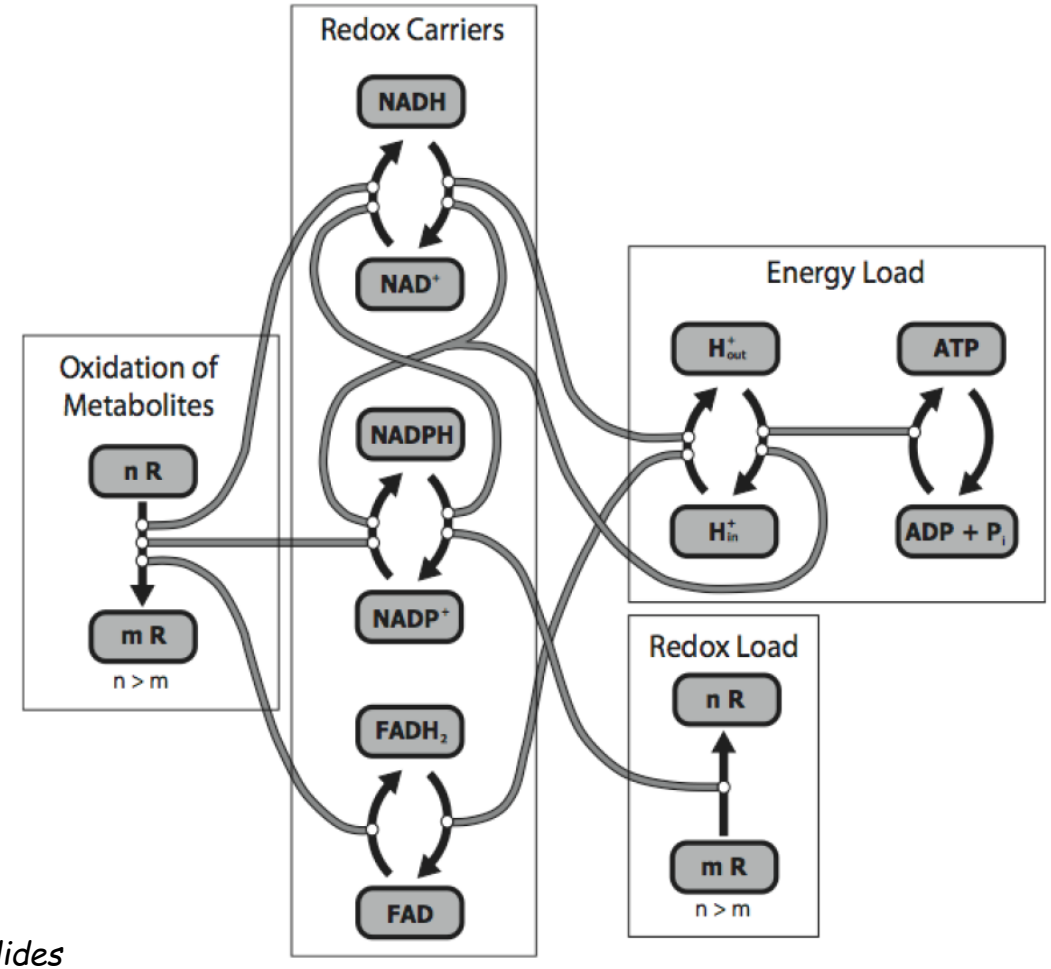
Reconstruction and Use of Microbial Metabolic Networks: the Core Escherichia coli Metabolic Model as an Educational Guide by Orth, Fleming, and Palsson (2010)

Oxidative Phosphorylation and Transfer of Reducing Equivalents



Reconstruction and Use of Microbial Metabolic Networks: the Core Escherichia coli Metabolic Model as an Educational Guide by Orth, Fleming, and Palsson (2010)

Redox Trafficking in the Core Metabolic Pathways: Cofactor View

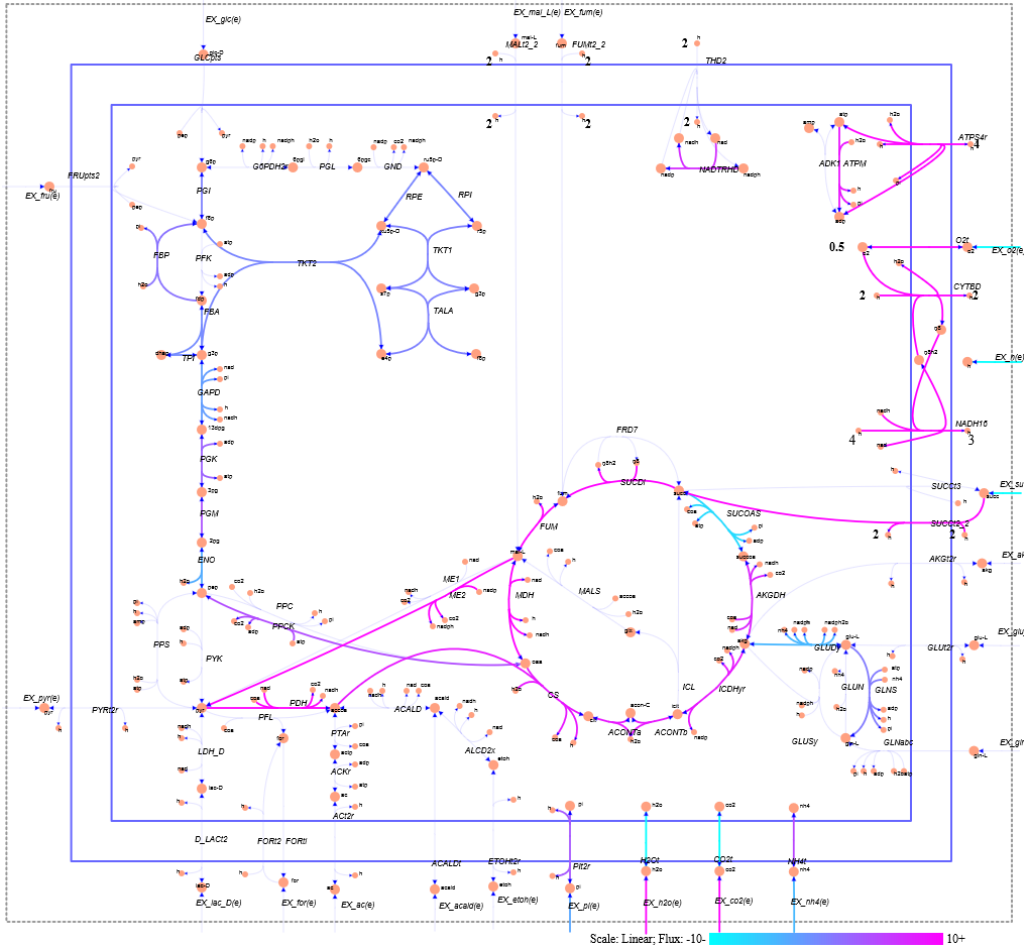


B. O. Palsson lectures on Systems Biology: Simulation of Dynamic Network States, Lecture #8.
http://sbrg.ucsd.edu/Publications/Books/SB2LectureSlides

Visualizing the Alternate Optimal Solution Flux Vectors

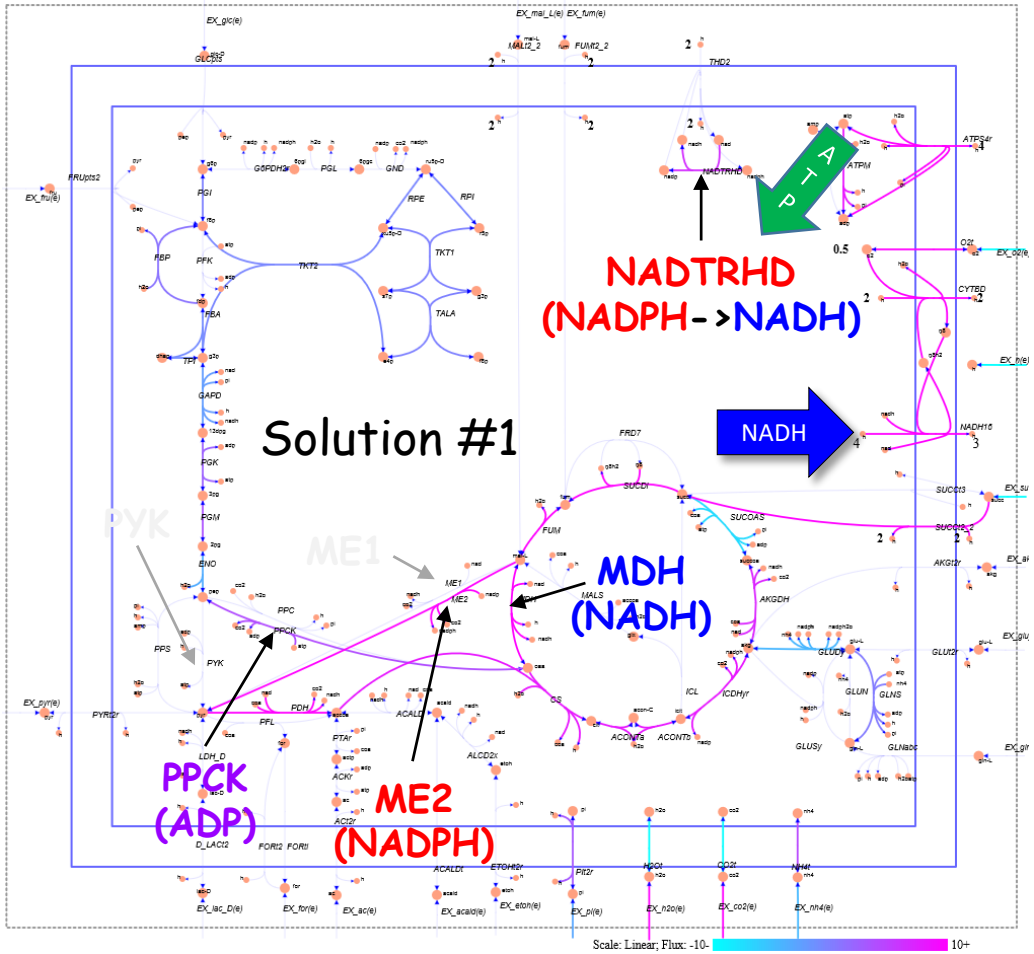
```
% findingOptimalSolutionsSuccVisualize.m
clear;

model = readCbModel('ecoli_core_model.mat');
model = changeRxnBounds(model,'EX_glc(e)',0,'l');
model = changeRxnBounds(model,'EX_o2(e)',-40,'l');
model = changeRxnBounds(model,'EX_succ(e)',-20,'l');
model = changeObjective(model,'Biomass_Ecoli_core_N(w/GAM)-Nmet2');

% List optimal solutions
changeCobraSolver('glpk','all');

[solutions] = enumerateOptimalSolutions(model);

v = solutions.fluxes(:,1); % Select which vector wanted to be mapped (1-3)
printFluxVector(model, FBAsolution.x, true)

map=readCbMap('ecoli_Textbook_ExportMap');
options.lb = -10;
options.ub = 10;
options.zeroFluxWidth = 0.1;
options.rxnDirMultiplier = 10;
drawFlux(map, model, v, options);
```

Solution #1



Alternate Optimal Solutions #1: Changing Fluxes

findingOptimalSolutionsSuccVisualize.m



ACONTa	8.13764	FBA	-0.835681	PGM	3.49017
ACONTb	8.13764	FBP	0.835681	PIt2r	3.0906
AKGDH	7.23122	FUM	27.2312	PPCK	3.92628
ATPM	8.39	GAPD	-2.23333	RPE	-0.603888
ATPS4r	57.7816	GLNS	0.214822	RPI	-0.603888
Biomass	0.840134	GLUDy	-4.36626	SUCCt2_2	20
CO2t	-44.2477	H2Ot	-30.3675	SUCDi	27.2312
CS	8.13764	ICDHyr	8.13764	SUCOAS	-7.23122
CYTBD	66.5528	MDH	13.565	TALA	-0.1503
ENO	-3.49017	ME2	13.6662	TKT1	-0.1503
EX_co2(e)	44.2477	NADH16	39.3216	TKT2	-0.453588
EX_h2o(e)	30.3675	NADTRHD	6.49242	TPI	-0.835681
EX_h(e)	-23.1469	NH4t	4.58108		
EX_nh4(e)	-4.58108	O2t	33.2764	Other AOS Reactions	
EX_o2(e)	-33.2764	PDH	11.2863	ME1	0
EX_pi(e)	-3.0906	PGI	-0.172228	PYK	0
EX_succ(e)	-20	PGK	2.23333		

ATP ADP NADH NADHP

Alternate Solution #1: NADH & ATP Node Fluxes

surfNet(model,'nadh[c]',0,flux,1)

Met #51 nadh[c], Nicotinamide adenine dinucleotide - reduced, C21H27N7O14P2, metCharges: -2

Consuming reactions with non-zero fluxes :

#49 GAPD (-2.233329), Bd: -1000 / 1000, glyceraldehyde-3-phosphate dehydrogenase,
g3p[c] + nad[c] + pi[c] <- 13dpg[c] + h[c] + nadh[c]

#67 NADH16 (39.32156), Bd: 0 / 1000, NADH dehydrogenase (ubiquinone-8 & 3 protons),
4 h[c] + nadh[c] + q8[c] -> 3 h[e] + nad[c] + q8h2[c]

Producing reactions with non-zero fluxes :

#8 AKGDH (7.231221), Bd: 0 / 1000, 2-Oxogluterate dehydrogenase,
akg[c] + coa[c] + nad[c] -> co2[c] + nadh[c] + succoa[c]

#13 Biomass_Ecoli_core_N(w/GAM)-Nmet2 (0.840134), Bd: 0 / 1000,
1.496 3pg[c] + 3.7478 accoa[c] + 59.81 atp[c] + 0.361 e4p[c] + 0.0709 f6p[c] + 0.129 g3p[c] + 0.205 g6p[c] + 0.2557 gln-L[c] + 4.9414 glu-L[c] + 59.81 h2o[c] + 3.547 nad[c] + 13.0279 nadph[c] + 1.7867 oaa[c] + 0.5191 pep[c] + 2.8328 pyr[c] + 0.8977 r5p[c] -> 59.81 adp[c] + 4.1182 akg[c] + 3.7478 coa[c] + 59.81 h[c] + 3.547 nadh[c] + 13.0279 nadp[c] + 59.81 pi[c]

#64 MDH (13.56499), Bd: -1000 / 1000, malate dehydrogenase,
mal-L[c] + nad[c] -> h[c] + nadh[c] + oaa[c]

#68 NADTRHD (6.492425), Bd: 0 / 1000, NAD transhydrogenase
nad[c] + nadph[c] -> nadh[c] + nadp[c]

#71 PDH (11.2863), Bd: 0 / 1000, pyruvate dehydrogenase,
coa[c] + nad[c] + pyr[c] -> accoa[c] + co2[c] + nadh[c]

surfNet(model,'atp[c]',0,flux,1)

Met #17 atp[c], **ATP**, C10H12N5O13P3, metCharges: -4

Consuming reactions with non-zero fluxes :

#11 ATPM (8.39), Bd: 8.39 / 8.39, ATP maintenance requirement,
atp[c] + h2o[c] -> adp[c] + h[c] + pi[c]

#13 Biomass_Ecoli_core_N(w/GAM)-Nmet2 (0.840134), Bd: 0 / 1000,
1.496 3pg[c] + 3.7478 accoa[c] + 59.81 atp[c] + 0.361 e4p[c] + 0.0709 f6p[c] + 0.129 g3p[c] + 0.205 g6p[c] + 0.2557 gln-L[c] + 4.9414 glu-L[c] + 59.81 h2o[c] + 3.547 nad[c] + 13.0279 nadph[c] + 1.7867 oaa[c] + 0.5191 pep[c] + 2.8328 pyr[c] + 0.8977 r5p[c] -> 59.81 adp[c] + 4.1182 akg[c] + 3.7478 coa[c] + 59.81 h[c] + 3.547 nadh[c] + 13.0279 nadp[c] + 59.81 pi[c]

#51 GLNS (0.214822), Bd: 0 / 1000, glutamine synthetase,
atp[c] + glu-L[c] + nh4[c] -> adp[c] + gln-L[c] + h[c] + pi[c]

#75 PGK (2.233329), Bd: -1000 / 1000, phosphoglycerate kinase,
3pg[c] + atp[c] -> 13dpg[c] + adp[c]

#80 PPCK (3.926283), Bd: 0 / 1000, phosphoenolpyruvate carboxykinase,
atp[c] + oaa[c] -> adp[c] + co2[c] + pep[c]

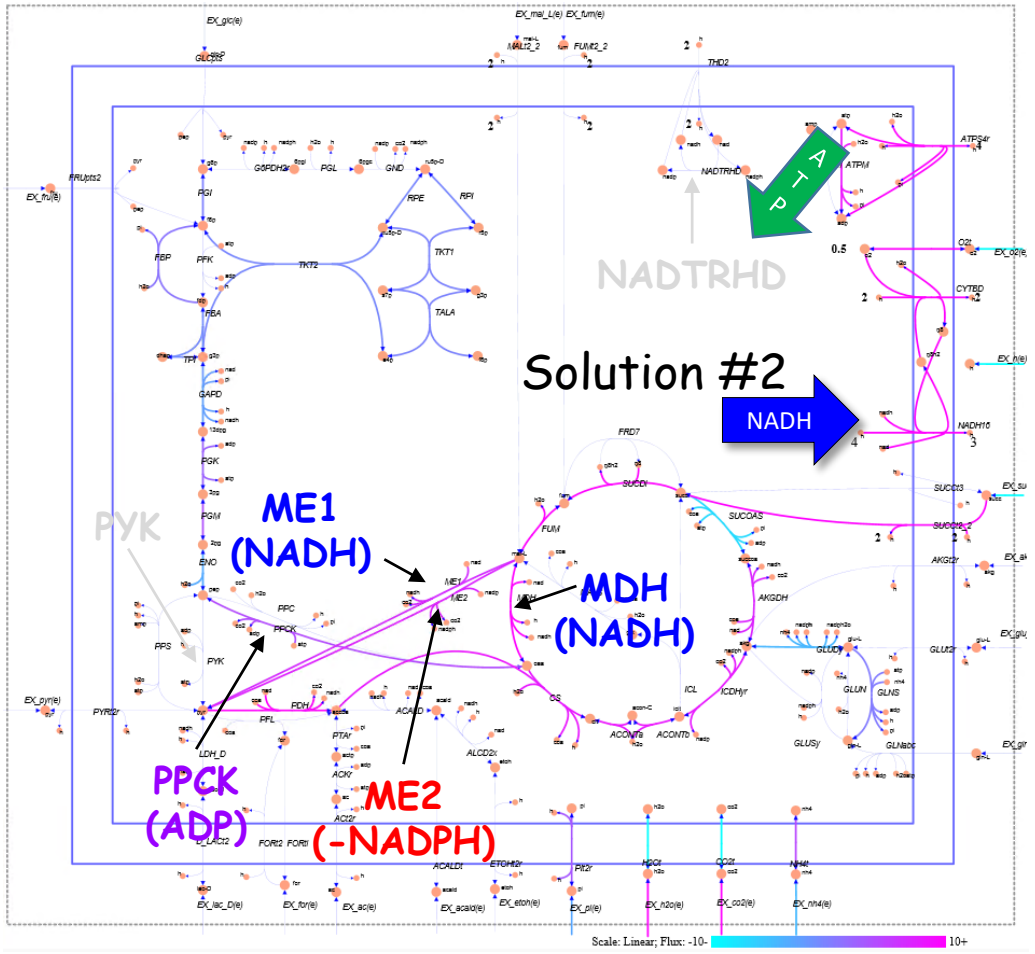
Producing reactions with non-zero fluxes :

#12 ATPS4r (57.78164), Bd: -1000 / 1000, ATP synthase (four protons for one ATP),
adp[c] + 4 h[e] + pi[c] -> atp[c] + h2o[c] + 3 h[c]

#90 SUCOAS (-7.231221), Bd: -1000 / 1000, succinyl-CoA synthetase (ADP-forming)
atp[c] + coa[c] + succ[c] <- adp[c] + pi[c] + succoa[c]

Alternate Optimal Solutions #2: Changing Fluxes

findingOptimalSolutionsSuccVisualize.m

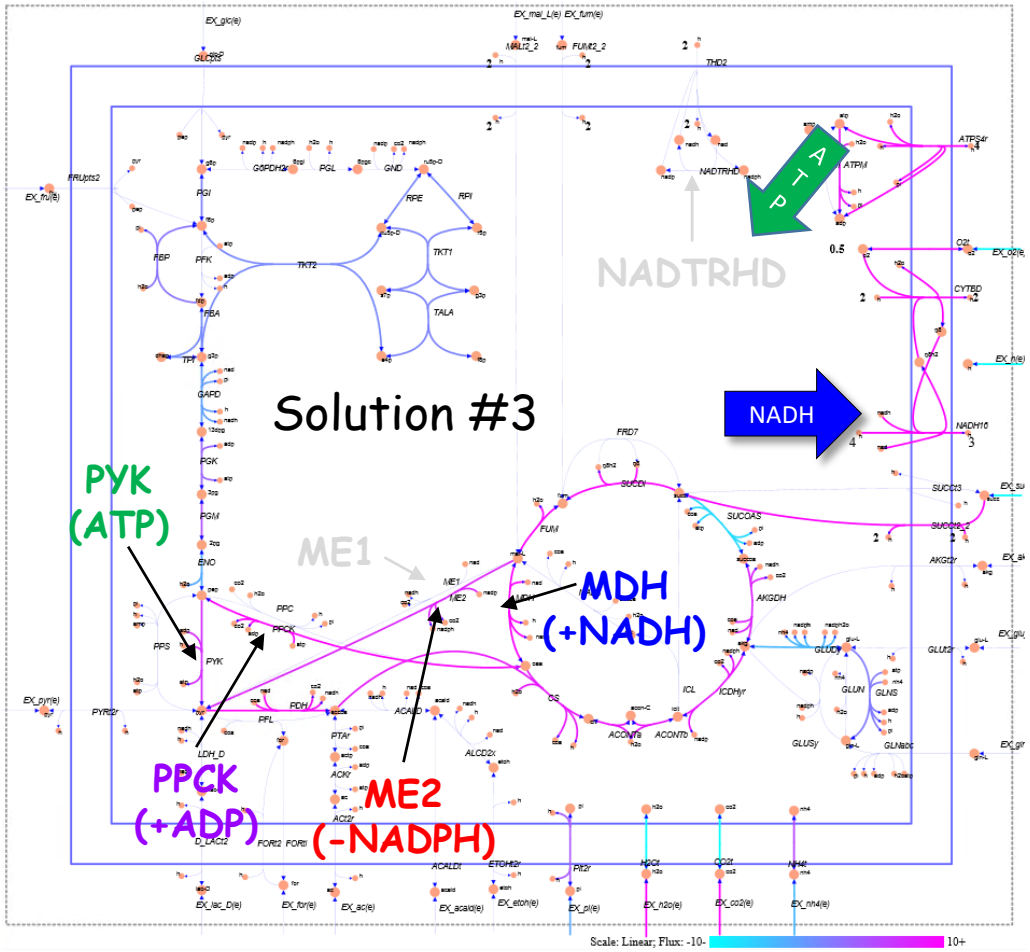


ACONTa	8.13764	FBA	-0.835681	PGM	3.49017
ACONTb	8.13764	FBP	0.835681	PIt2r	3.0906
AKGDH	7.23122	FUM	27.2312	PPCK	3.92628
ATPM	8.39	GAPD	-2.23333	RPE	-0.603888
ATPS4r	57.7816	GLNS	0.214822	RPI	-0.603888
Biomass	0.840134	GLUDy	-4.36626	SUCCt2_2	20
CO2t	-44.2477	H2Ot	-30.3675	SUCDi	27.2312
CS	8.13764	ICDHyr	8.13764	SUCOAS	-7.23122
CYTBD	66.5528	MDH	13.565	TALA	-0.1503
ENO	-3.49017	ME1	6.49242	TKT1	-0.1503
EX_co2(e)	44.2477	ME2	7.1738	TKT2	-0.453588
EX_h2o(e)	30.3675	NADH16	39.3216	TPI	-0.835681
EX_h(e)	-23.1469	NH4t	4.58108		
EX_nh4(e)	-4.58108	O2t	33.2764	Other AOS Reactions	
EX_o2(e)	-33.2764	PDH	11.2863	NADTRHD	0
EX_pi(e)	-3.0906	PGI	-0.172228	PYK	0
EX_succ(e)	-20	PGK	2.23333		

ATP ADP NADH NADHP

Alternate Optimal Solutions #3: Changing Fluxes

findingOptimalSolutionsSuccVisualize.m



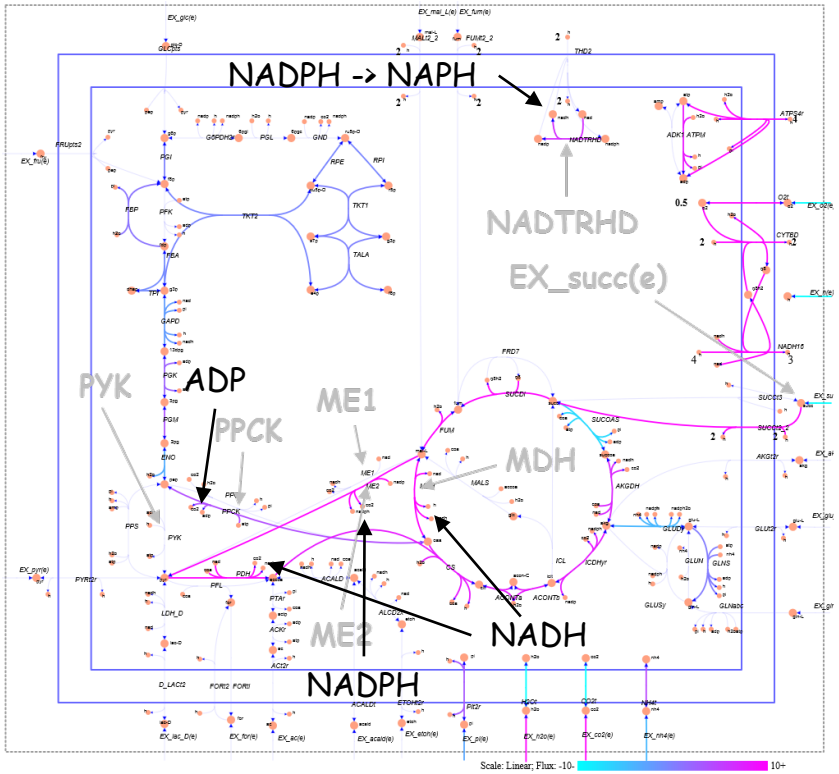
ACONTa	8.13764	FBA	-0.835681	PGM	3.49017
ACONTb	8.13764	FBP	0.835681	PIt2r	3.0906
AKGDH	7.23122	FUM	27.2312	PPCK	10.4187
ATPM	8.39	GAPD	-2.23333	PYK	6.49242
ATPS4r	57.7816	GLNS	0.214822	RPE	-0.603888
Biomass	0.840134	GLUDy	-4.36626	RPI	-0.603888
CO2t	-44.2477	H2Ot	-30.3675	SUCCt2_2	20
CS	8.13764	ICDHyr	8.13764	SUCDi	27.2312
CYTBD	66.5528	MDH	20.0574	SUCOAS	-7.23122
ENO	-3.49017	ME2	7.1738	TALA	-0.1503
EX_co2(e)	44.2477	NADH16	39.3216	TKT1	-0.1503
EX_h2o(e)	30.3675	NH4t	4.58108	TKT2	-0.453588
EX_h(e)	-23.1469	O2t	33.2764	TPI	-0.835681
EX_nh4(e)	-4.58108	PDH	11.2863		
EX_o2(e)	-33.2764	PGI	-0.172228	Other AOS Reactions	
EX_pi(e)	-3.0906	PGK	2.23333	NADTRHD	0
EX_succ(e)	-20			ME1	0

ATP ADP NADH NADHP

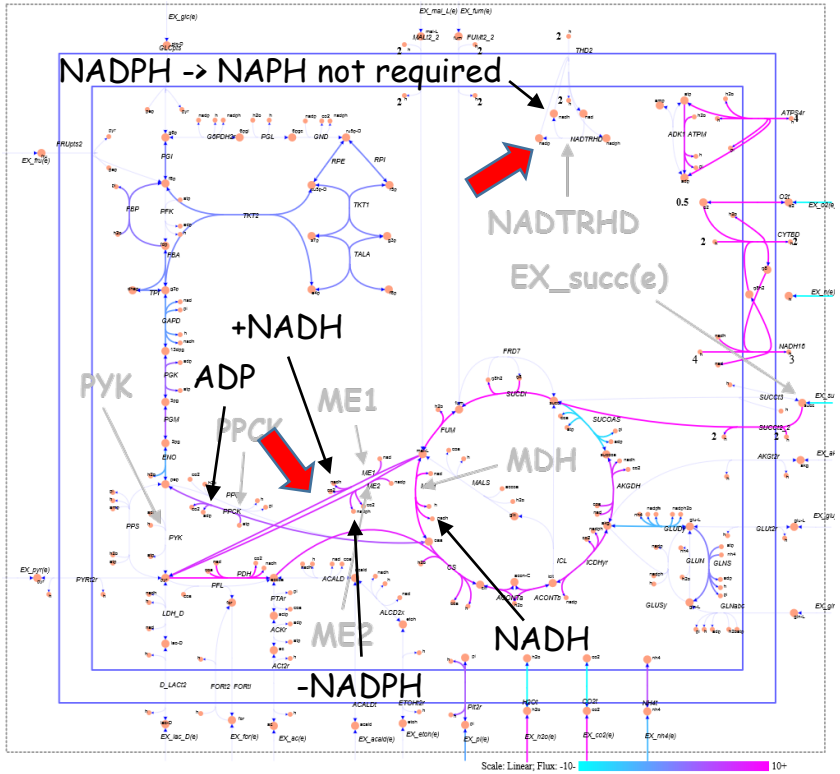
Alternate Optimal Solutions

findingOptimalSolutionsSuccVisualize.m

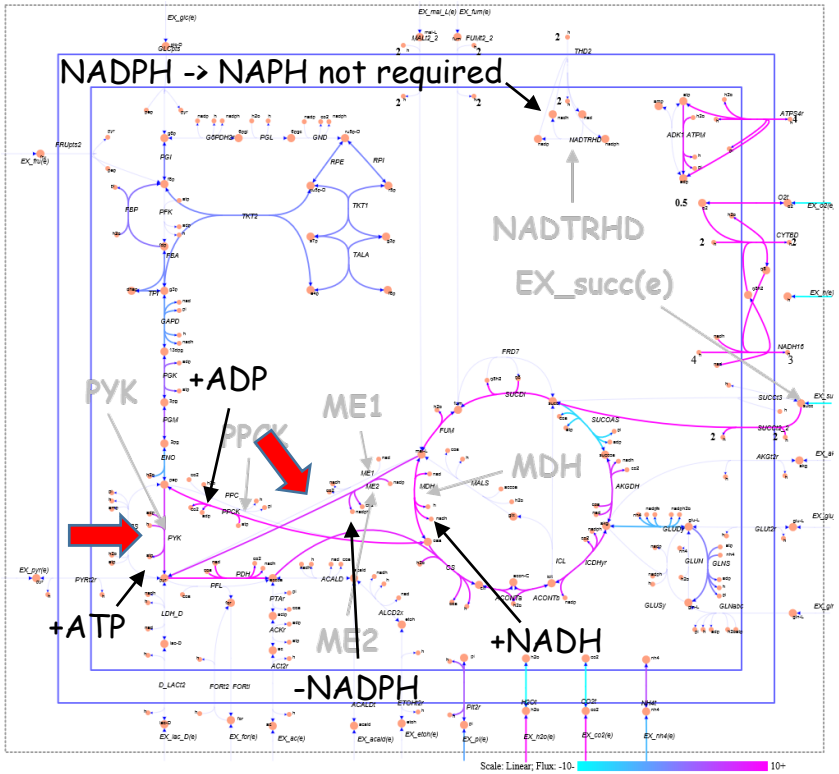
All three solution produce the same amount of ATP by providing the same amount of NADH for the electron transport chain



Solution #1



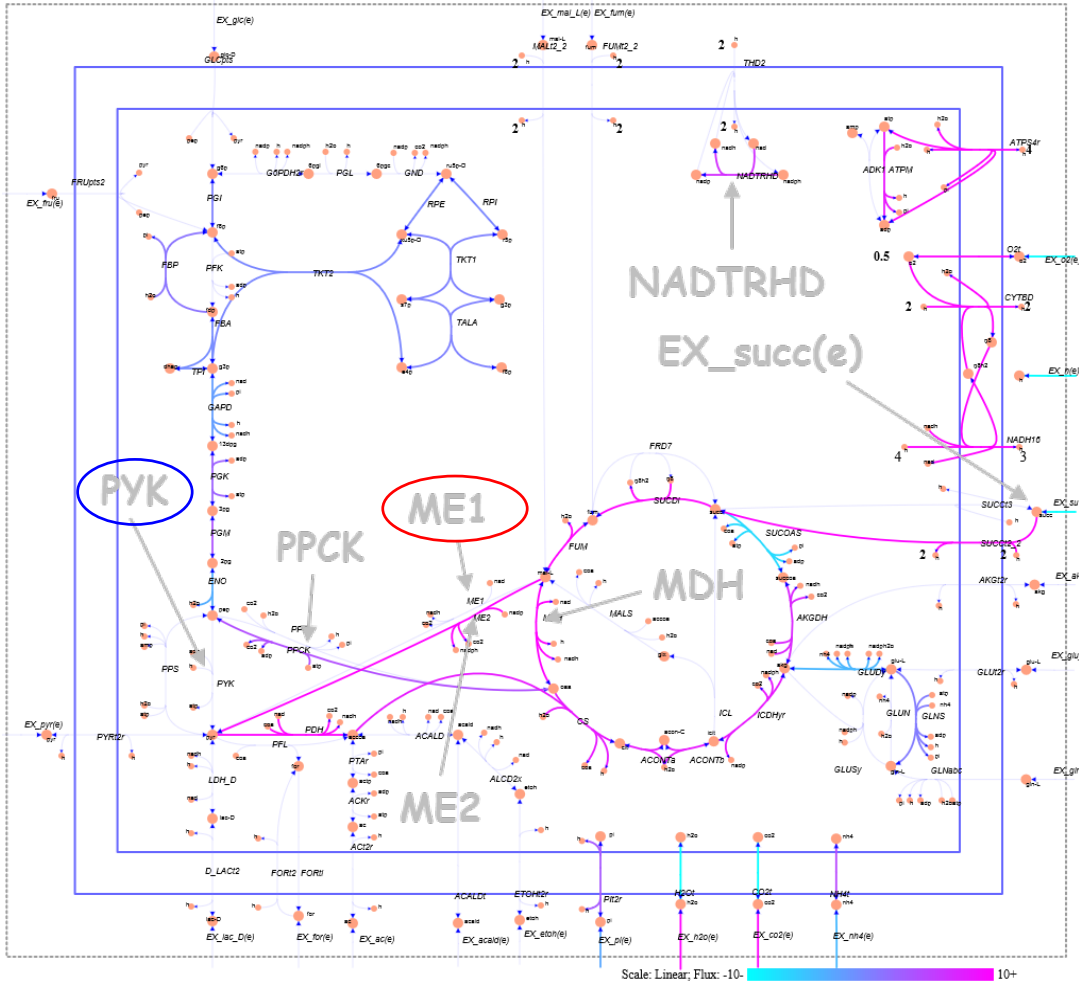
Solution #2



Solution #3

Reducing Alternate Optimal Solutions Code

```
% findingOptimalSolutionsSuccVisualizeOne.m
clear; clc;

model = readCbModel('ecoli_core_model.mat');
model = changeRxnBounds(model,'EX_glc(e)',0,'l');
model = changeRxnBounds(model,'EX_o2(e)',-40,'l');
model = changeRxnBounds(model,'EX_succ(e)',-20,'l');
model = changeObjective(model,'Biomass_Ecoli_core_N(w/GAM)-Nmet2');

model = changeRxnBounds(model,'ME1',0,'b');
% model = changeRxnBounds(model,'NADTRHD',0,'b');
model = changeRxnBounds(model,'PYK',0,'b');

% List optimal solutions
solverOK = changeCobraSolver('glpk','all');

[solutions] = enumerateOptimalSolutions(model);

v = solutions.fluxes(:,1); % Select which vector wanted to be mapped
(1-3)
printFluxVector(model, v, true)

map=readCbMap('ecoli_Textbook_ExportMap');
options.lb = -10;
options.ub = 10;
options.zeroFluxWidth = 0.1;
options.rxnDirMultiplier = 10;
drawFlux(map, model, v, options);
```

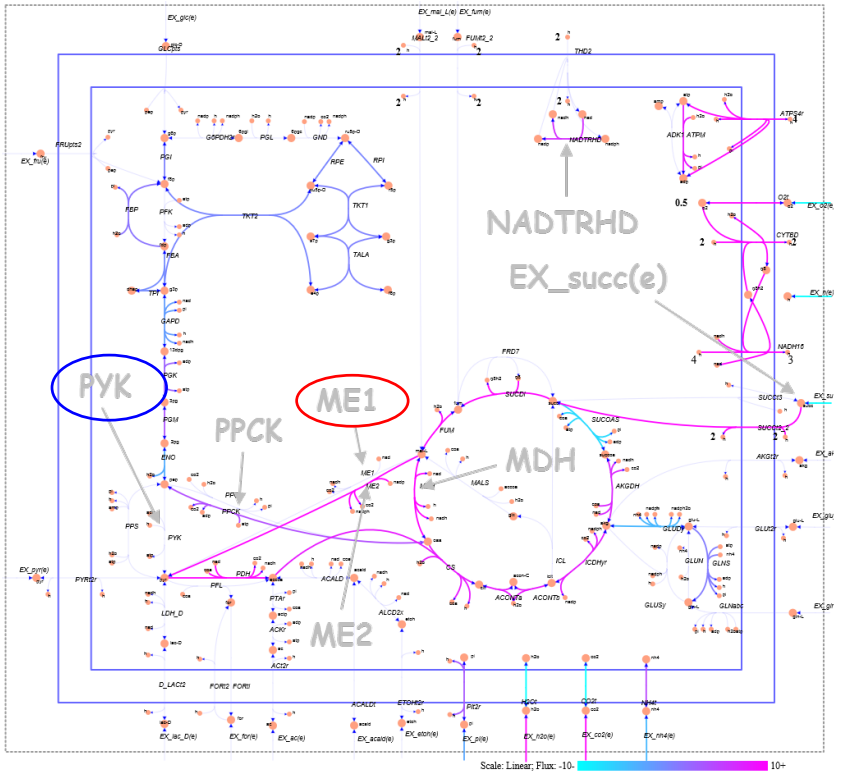


Solution #1

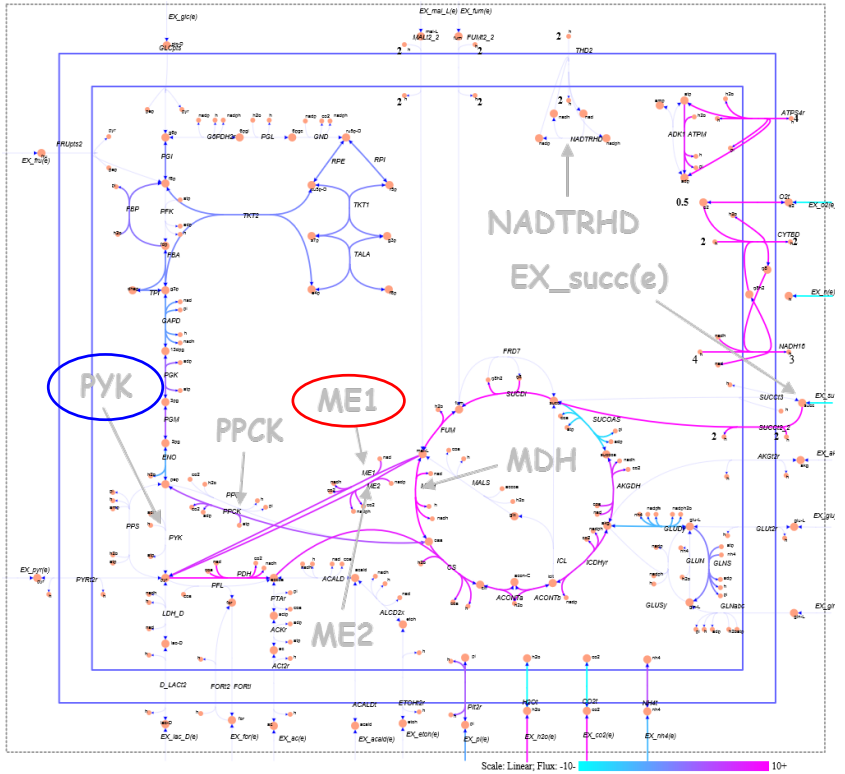
Reducing Alternate Optimal Solutions

findingOptimalSolutionsSuccVisualizeOne.m

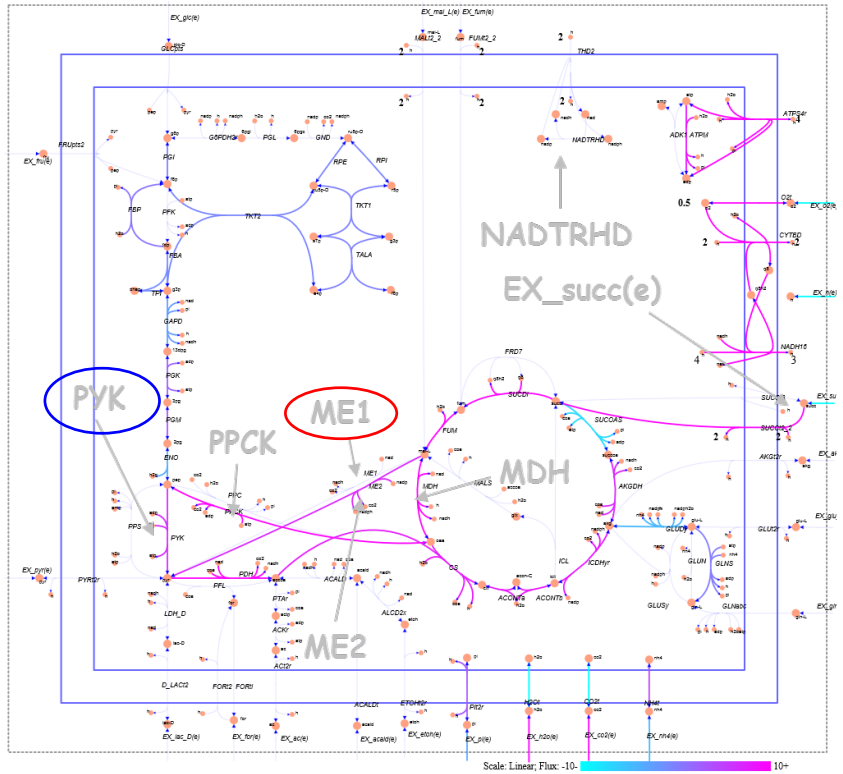
If both ME1 and PYK are set to zero (knocked out), then there will only be one optimal solution (in this simple model)



Solution #1



Solution #2

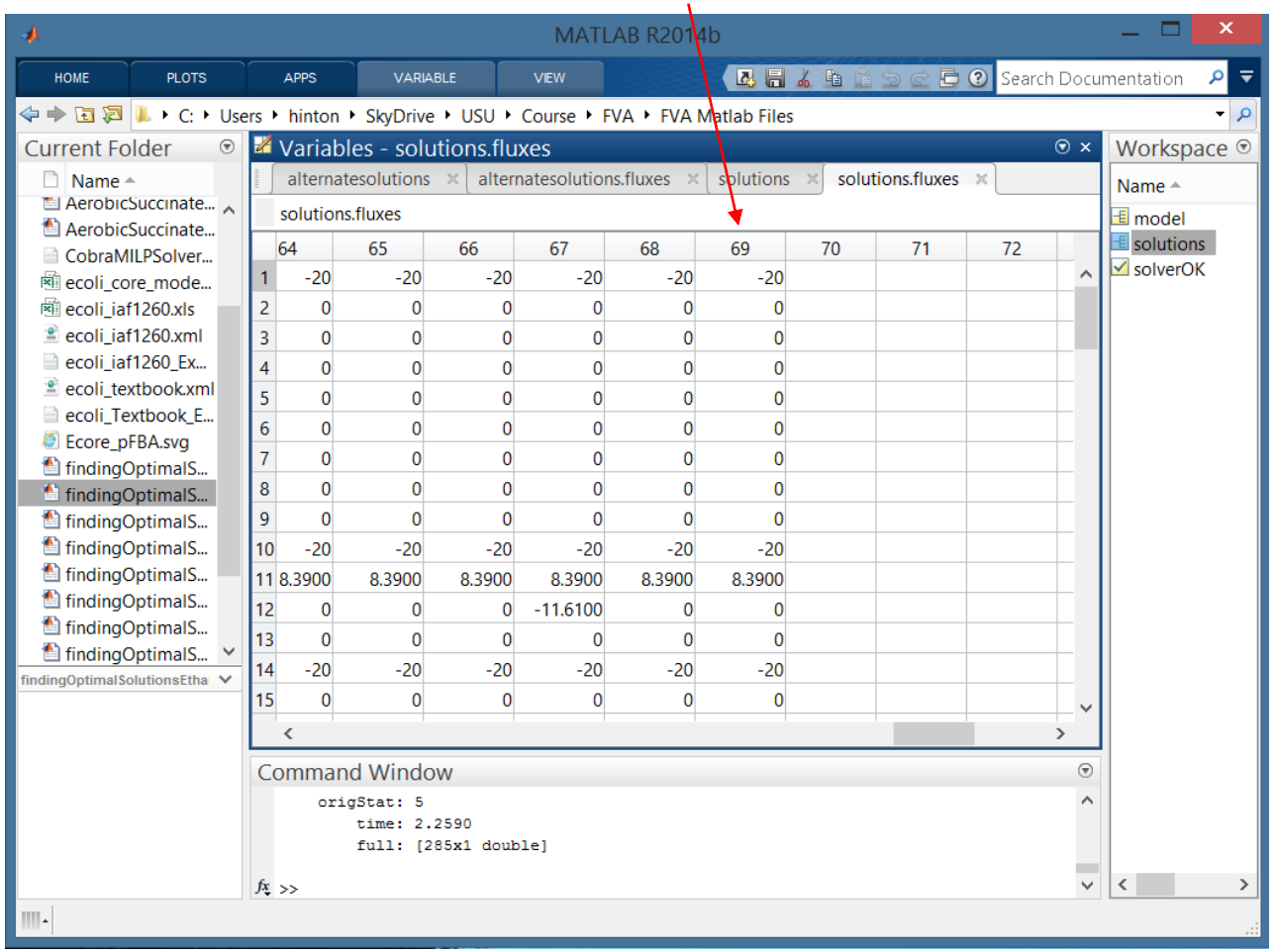


Solution #3

Alternate Optimal Solutions for Ethanol Production

```
% findingOptimalSolutionsEthanol.m
clear;

model = readCbModel('ecoli_core_model.mat');

model = changeRxnBounds(model,'EX_glc(e)',-10,'l');
model = changeRxnBounds(model,'EX_o2(e)',0,'l');

model = changeObjective(model,'EX_etoh(e)');

% List optimal solutions
solverOK = changeCobraSolver('glpk','all');
[solutions] = enumerateOptimalSolutions(model);
```

69 Alternate Optimal Flux Vectors



Review Questions

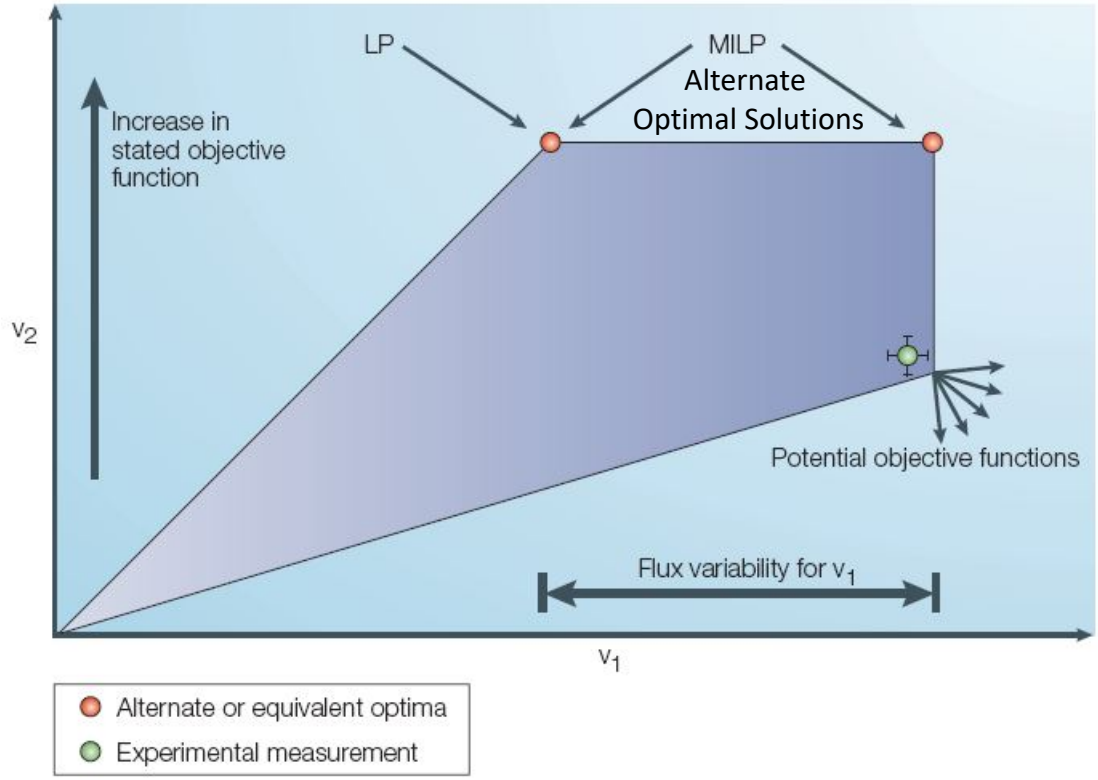
- What are alternate optimal solutions?
- What is the relationship between alternate optimal solutions and a cell's phenotype?
- What are silent phenotypes?
- How can you find the alternate optimal solutions using the Cobra Toolbox?
- How many alternate optimal solutions can there be for a given phenotype?
- How many alternate optimal solutions can there be for a carbon source?
- Do aerobic/anaerobic conditions impact the number alternate optimal solutions?
- Does the choice of objective function impact the number alternate optimal solutions?

Lesson Outline

- Alternate Optimal Solutions
- Flux Variability Analysis
- Parsimonious FBA

Flux Variability Analysis

- This method identifies the allowable range of flux values through a given reaction by finding the maximum and minimum possible fluxes through the particular reaction for a given **maximum objective value**.
- All reactions under test have the same objective value
- This analysis method begins by finding the optimal value of the objective function for a given set of constraints and then optimizes for the minimum and maximum flux values for each reaction (*1 + 2n* optimizations where *n* is the number of reactions).
- A method that can be used to identify alternate optimal pathways.

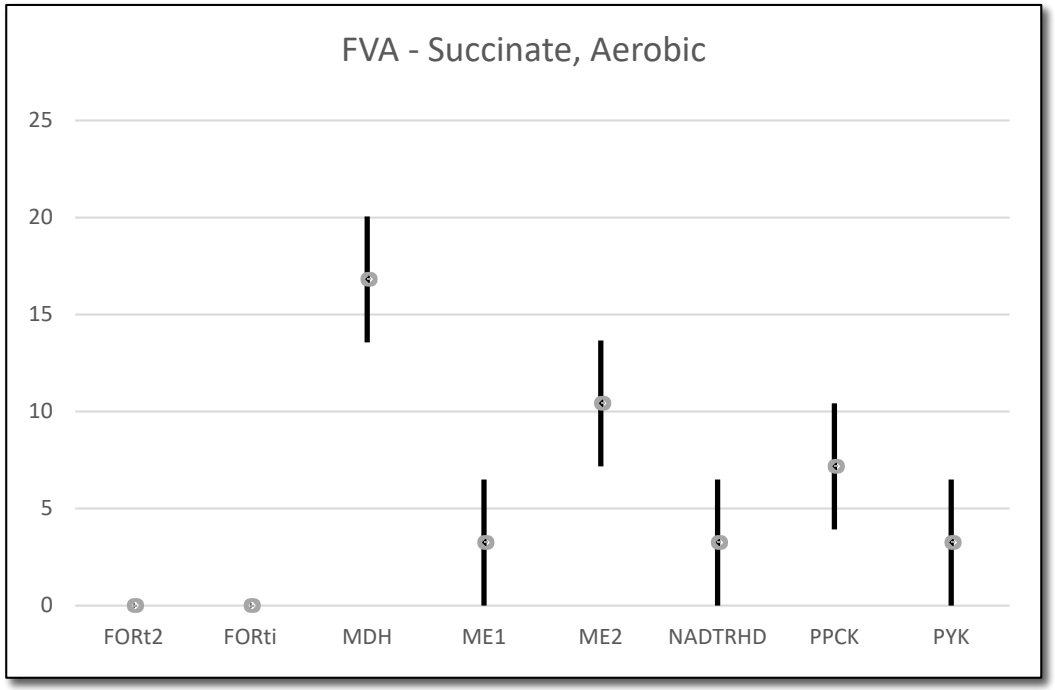


Price, N. D., J. L. Reed, et al. (2004). "Genome-scale models of microbial cells: evaluating the consequences of constraints." Nature reviews. Microbiology 2(11): 886-897.

Flux Variability Analysis Example

```
% FluxVariabilitySuccinate.m
% Load model
model = readCbModel('ecoli_core_model.mat');

% Change carbon source from glucose to succinate
model = changeRxnBounds(model,'EX_glc(e)',0,'l');
model = changeRxnBounds(model,'EX_succ(e)',-20,'l');

% Set optimization objective to Biomass_Ecoli_core_N(w/GAM)-Nmet2
model = changeObjective(model,'Biomass_Ecoli_core_N(w/GAM)-Nmet2');

% Perform flux variability analysis
[minFlux,maxFlux]=fluxVariability(model,100,'max',model.rxns,false,false);

% Print flux values
printFluxVector(model, [minFlux, maxFlux], true)
```

Does not allow loops



FVA_Succinate_Anaerobic.xlsx

Flux Variability Analysis Example Output (No Loops)

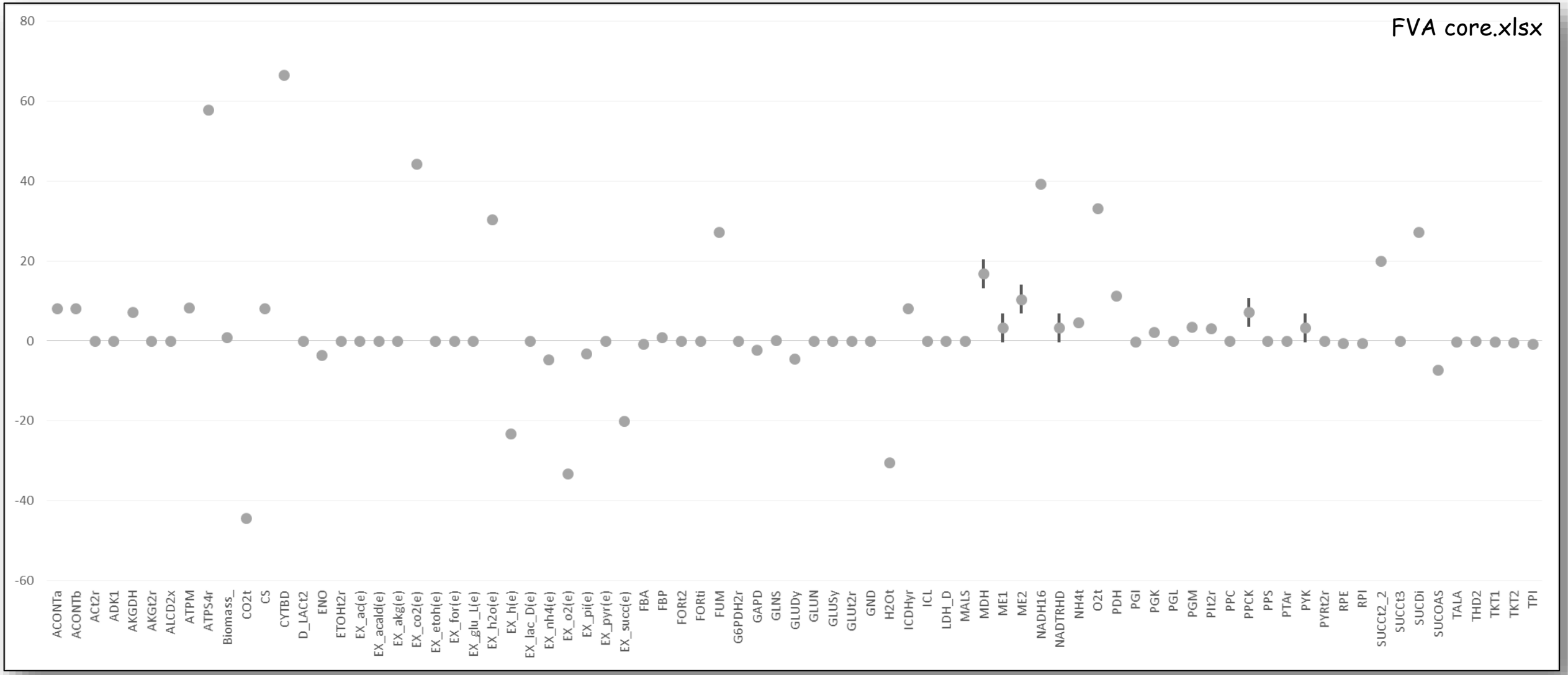
FVA_example1.xlsx

Reaction	Lower	Upper	Difference
ACALD	-5.07E-06	0	0
ACALDt	-5.07E-06	0	0
ACKr	-7.76E-06	0	0
ACONTa	8.13763	8.13764	0
ACONTb	8.13763	8.13764	0
ACt2r	-7.76E-06	0	0
ADK1	0	3.30E-05	0
AKGDH	7.23119	7.23122	0
AKGt2r	-3.07E-06	0	0
ALCD2x	-4.40E-06	0	0
ATPM	8.39	8.39	0
ATPS4r	57.7816	57.7817	0
Biomass	0.840134	0.840134	0
CO2t	-44.2477	-44.2477	0
CS	8.13763	8.13764	0
CYTBD	66.5528	66.5528	0
D_LACt2	-4.25E-06	0	0
ENO	-3.49018	-3.49017	0
ETOHt2r	-4.40E-06	0	0
EX_ac(e)	0	7.76E-06	0
EX_acald(e)	0	5.07E-06	0
EX_akg(e)	0	3.07E-06	0
EX_co2(e)	44.2477	44.2477	0
EX_etoh(e)	0	4.40E-06	0
EX_for(e)	0	2.20E-05	0
EX_glu_L(e)	0	2.75E-06	0
EX_h2o(e)	30.3675	30.3675	0
EX_h(e)	-23.1469	-23.1469	0

Reaction	Lower	Upper	Difference
EX_lac_D(e)	0	4.25E-06	0
EX_nh4(e)	-4.58109	-4.58108	0
EX_o2(e)	-33.2764	-33.2764	0
EX_pi(e)	-3.0906	-3.0906	0
EX_pyr(e)	0	5.07E-06	0
EX_succ(e)	-20	-20	0
FBA	-0.8357	-0.83568	0
FBP	0.835681	0.835714	0
FORt2	0	0.000132	0.00013187
FORti	0	0.000132	0.00013187
FUM	27.2312	27.2312	0
G6PDH2r	0	4.40E-05	0
GAPD	-2.23334	-2.23E+00	0
GLNS	0.214822	0.214855	0
GLUDy	-4.36626	-4.36623	0
GLUN	0	3.30E-05	0
GLUSy	0	3.30E-05	0
GLUt2r	-2.75E-06	0.00E+00	0
GND	0.00E+00	4.40E-05	0
H2Ot	-30.3675	-3.04E+01	0
ICDHyr	8.13761	8.13764	0
ICL	0	3.30E-05	0
LDH_D	-4.25E-06	0.00E+00	0
MALS	0.00E+00	3.30E-05	0
MDH	13.565	2.01E+01	6.49253
ME1	0	6.4925	6.4925
ME2	7.17372	13.6663	6.49254
NADH16	39.3216	39.3216	0

Reaction	Lower	Upper	Difference
NADTRHD	0	6.4925	6.4925
NH4t	4.58108	4.58109	0
O2t	33.2764	33.2764	0
PDH	11.2863	11.2863	0
PFK	0	3.30E-05	0
PFL	0	2.20E-05	0
PGI	-0.17227	-0.17223	0
PGK	2.23333	2.23334	0
PGL	0	4.40E-05	0
PGM	3.49017	3.49018	0
PIt2r	3.0906	3.0906	0
PPC	0	3.30E-05	0
PPCK	3.92625	10.4188	6.49254
PPS	0	3.30E-05	0
PTAr	0	7.76E-06	0
PYK	0	6.4925	6.4925
PYRt2r	-5.07E-06	0	0
RPE	-0.60389	-0.60386	0
RPI	-0.6039	-0.60389	0
SUCCt2_2	20	20	0
SUCCt3	0	4.40E-05	0
SUCDi	27.2312	27.2312	0
SUCOAS	-7.23122	-7.23119	0
TALA	-0.1503	-0.15029	0
THD2	0	6.59E-05	0
TKT1	-0.1503	-0.15029	0
TKT2	-0.45359	-0.45357	0
TPI	-0.8357	-0.83568	0

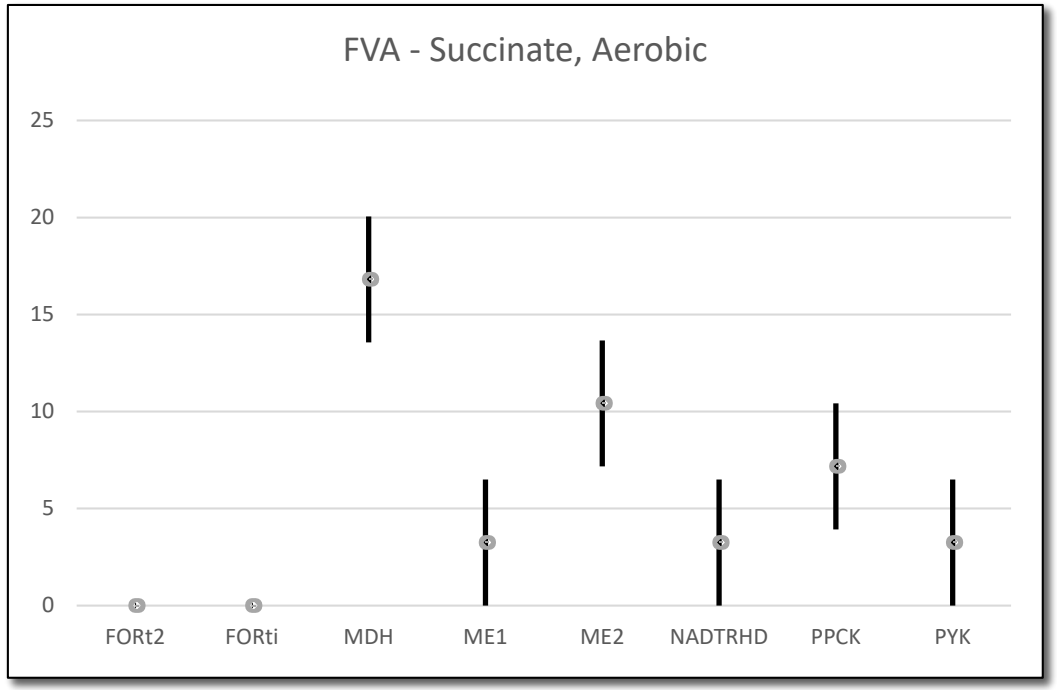
Flux Variability Analysis Example



Flux Variability Analysis Example (Continued)

Reaction	Minimum Flux (mmol gDW^{-1} hr^{-1})	Maximum Flux (mmol gDW^{-1} hr^{-1})
MDH	13.56	20.06
ME1	0	6.49
ME2	7.17	13.67
NADTRHD	0	6.49
PPCK	3.93	10.42
PYK	0	6.49

Variable Reactions For Growth On Succinate
(Same as Alternate Optimal Solutions Example)

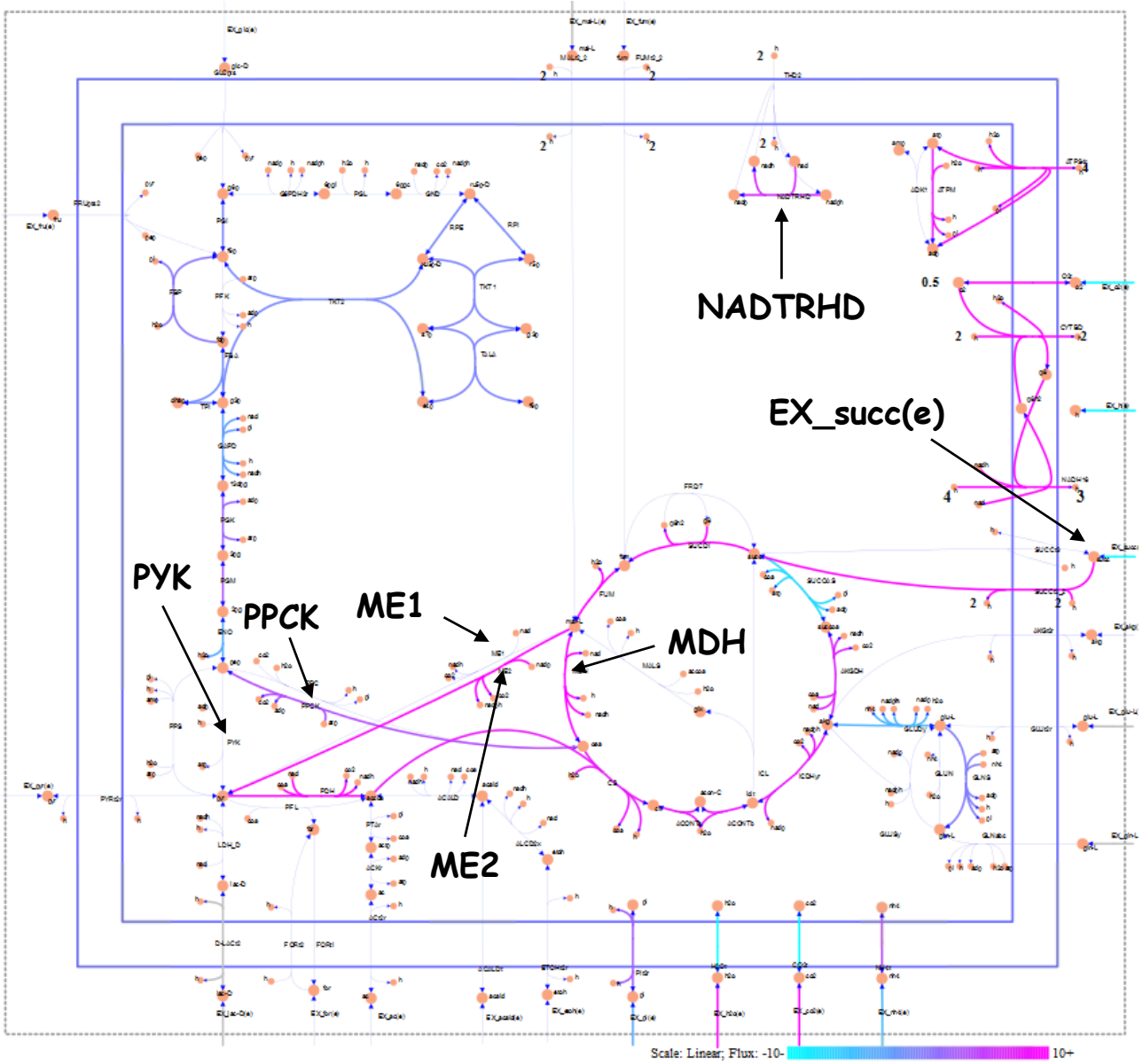


FVA_Succinate_Anaerobic.xlsx

Fluxes Identified Through Flux Variability Analysis

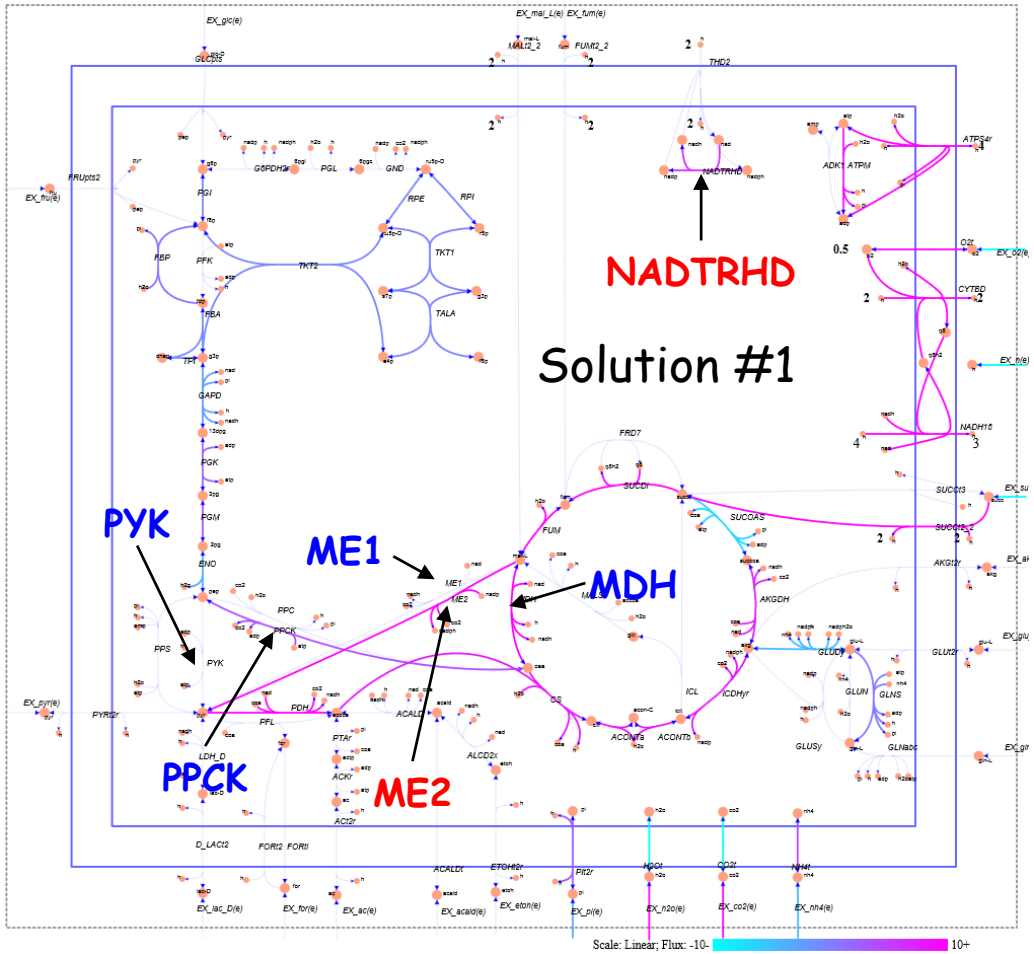
(AerobicSuccinateBioMass.m)

- **MDH** (malate dehydrogenase)
- **ME1** (malic enzyme (NAD))
- **ME2** (malic enzyme (NADP))
- **NADTRHD** (NAD transhydrogenase)
- **PPCK** (phosphoenolpyruvate carboxykinase)
- **PYK** (pyruvate kinase)



Flux Variability Analysis #1

findingOptimalSolutionsSuccVisualize.m

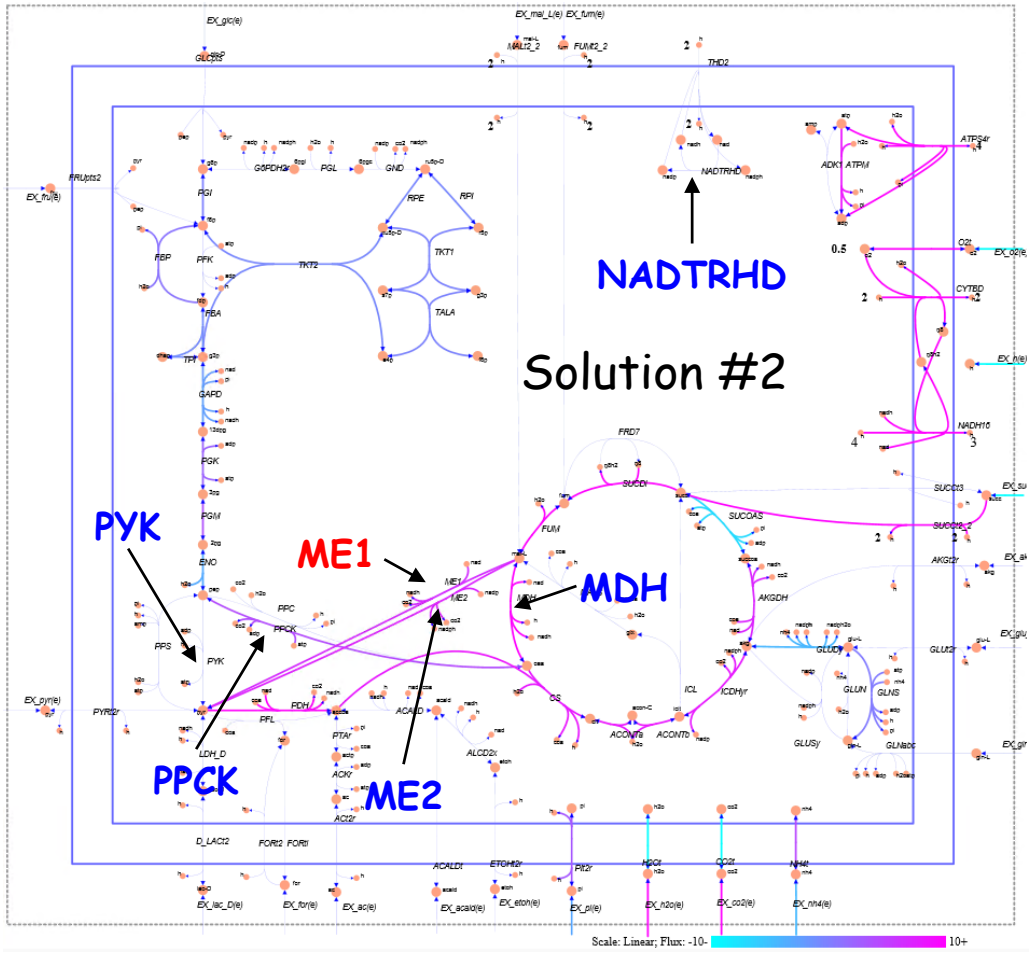


ACONTa	8.13764	FBA	-0.835681	PGM	3.49017
ACONTb	8.13764	FBP	0.835681	PIt2r	3.0906
AKGDH	7.23122	FUM	27.2312	PPCK	3.92628
ATPM	8.39	GAPD	-2.23333	RPE	-0.603888
ATPS4r	57.7816	GLNS	0.214822	RPI	-0.603888
Biomass	0.840134	GLUDy	-4.36626	SUCCt2_2	20
CO2t	-44.2477	H2Ot	-30.3675	SUCDi	27.2312
CS	8.13764	ICDHyr	8.13764	SUCOAS	-7.23122
CYTBD	66.5528	MDH	13.565	TALA	-0.1503
ENO	-3.49017	ME2	13.6662	TKT1	-0.1503
EX_co2(e)	44.2477	NADH16	39.3216	TKT2	-0.453588
EX_h2o(e)	30.3675	NADTRHD	6.49242	TPI	-0.835681
EX_h(e)	-23.1469	NH4t	4.58108		
EX_nh4(e)	-4.58108	O2t	33.2764	Other FVA Reactions	
EX_o2(e)	-33.2764	PDH	11.2863	ME1	0
EX_pi(e)	-3.0906	PGI	-0.172228	PYK	0
EX_succ(e)	-20	PGK	2.23333		

FVA Upper Bound; Lower Bound.

Flux Variability Analysis #2

findingOptimalSolutionsSuccVisualize.m

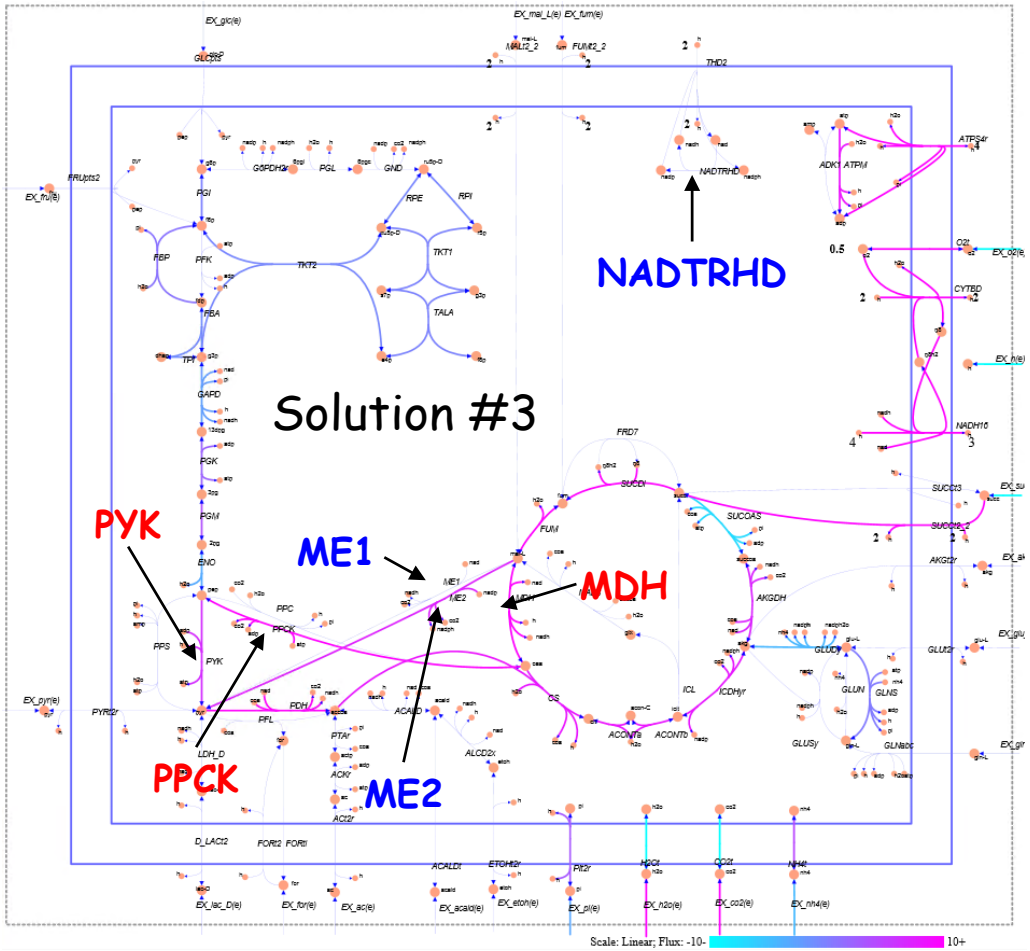


ACONTa	8.13764	FBA	-0.835681	PGM	3.49017
ACONTb	8.13764	FBP	0.835681	PIt2r	3.0906
AKGDH	7.23122	FUM	27.2312	PPCK	3.92628
ATPM	8.39	GAPD	-2.23333	RPE	-0.603888
ATPS4r	57.7816	GLNS	0.214822	RPI	-0.603888
Biomass	0.840134	GLUDy	-4.36626	SUCCt2_2	20
CO2t	-44.2477	H2Ot	-30.3675	SUCDi	27.2312
CS	8.13764	ICDHyr	8.13764	SUCOAS	-7.23122
CYTBD	66.5528	MDH	13.565	TALA	-0.1503
ENO	-3.49017	ME1	6.49242	TKT1	-0.1503
EX_co2(e)	44.2477	ME2	7.1738	TKT2	-0.453588
EX_h2o(e)	30.3675	NADH16	39.3216	TPI	-0.835681
EX_h(e)	-23.1469	NH4t	4.58108		
EX_nh4(e)	-4.58108	O2t	33.2764	Other FVA Reactions	
EX_o2(e)	-33.2764	PDH	11.2863	NADTRHD	0
EX_pi(e)	-3.0906	PGI	-0.172228	PYK	0
EX_succ(e)	-20	PGK	2.23333		

FVA Upper Bound; Lower Bound.

Flux Variability Analysis #3

findingOptimalSolutionsSuccVisualize.m

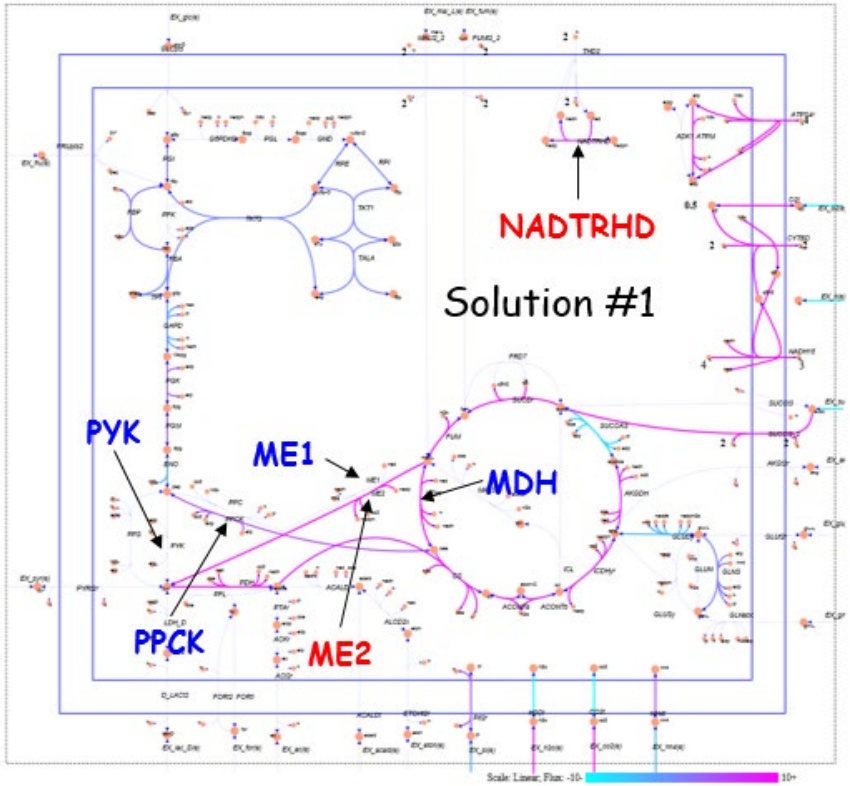


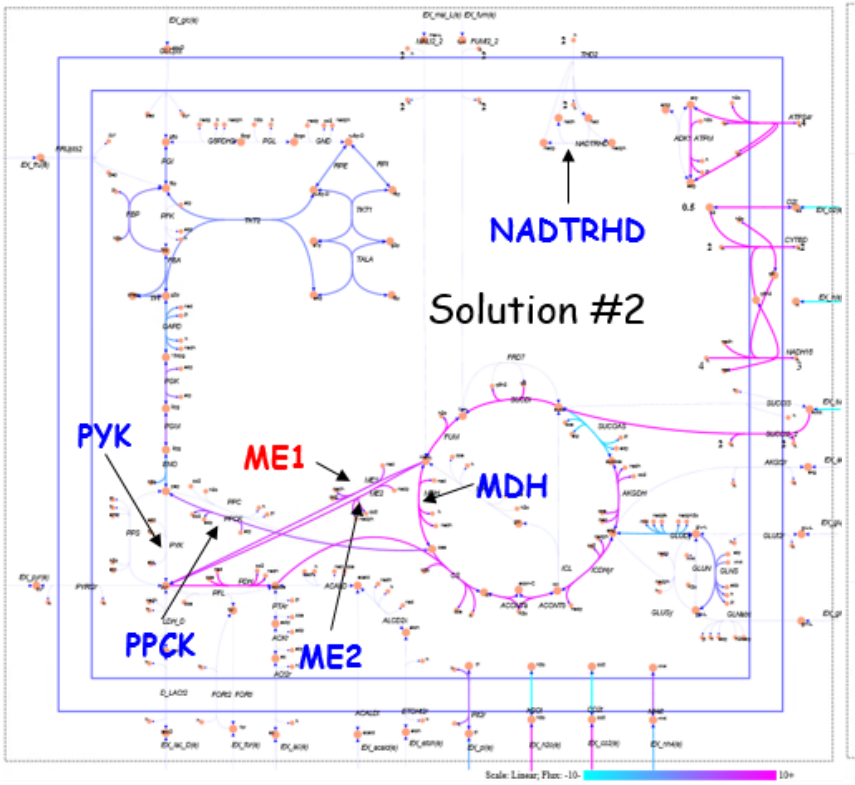
ACONTa	8.13764	FBA	-0.835681	PGM	3.49017
ACONTb	8.13764	FBP	0.835681	PIt2r	3.0906
AKGDH	7.23122	FUM	27.2312	PPCK	10.4187
ATPM	8.39	GAPD	-2.23333	PYK	6.49242
ATPS4r	57.7816	GLNS	0.214822	RPE	-0.603888
Biomass	0.840134	GLUDy	-4.36626	RPI	-0.603888
CO2t	-44.2477	H2Ot	-30.3675	SUCCt2_2	20
CS	8.13764	ICDHyr	8.13764	SUCDi	27.2312
CYTBD	66.5528	MDH	20.0574	SUCOAS	-7.23122
ENO	-3.49017	ME2	7.1738	TALA	-0.1503
EX_co2(e)	44.2477	NADH16	39.3216	TKT1	-0.1503
EX_h2o(e)	30.3675	NH4t	4.58108	TKT2	-0.453588
EX_h(e)	-23.1469	O2t	33.2764	TPI	-0.835681
EX_nh4(e)	-4.58108	PDH	11.2863		
EX_o2(e)	-33.2764	PGI	-0.172228	Other FVA Reactions	
EX_pi(e)	-3.0906	PGK	2.23333	NADTRHD	0
EX_succ(e)	-20			ME1	0

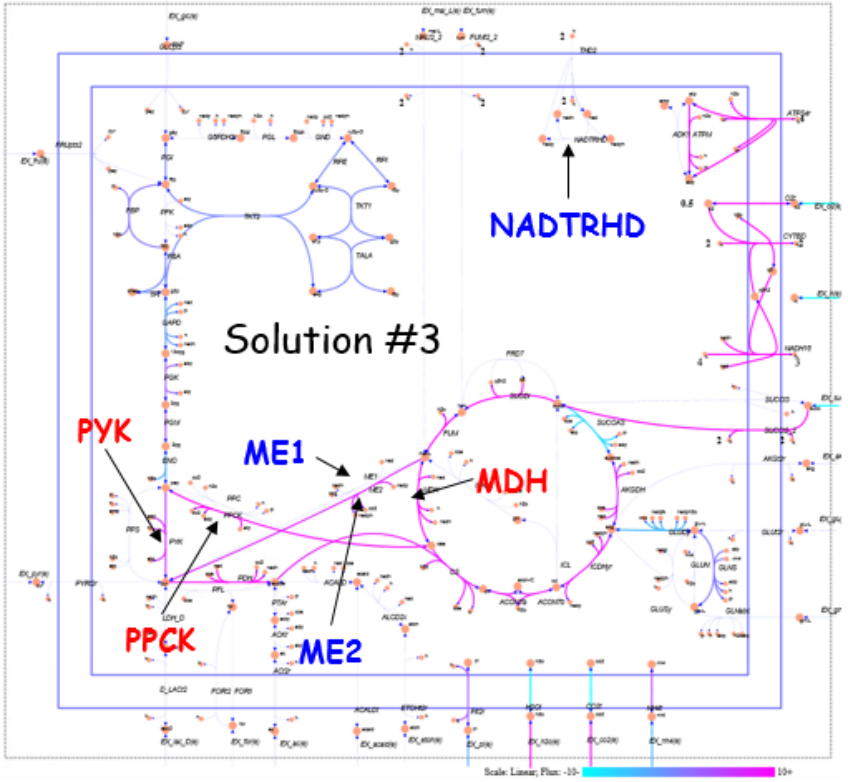
FVA Upper Bound; Lower Bound.

Three Alternate Optimal Solutions

All three solution produce the same amount of NADH for the electron transport chain





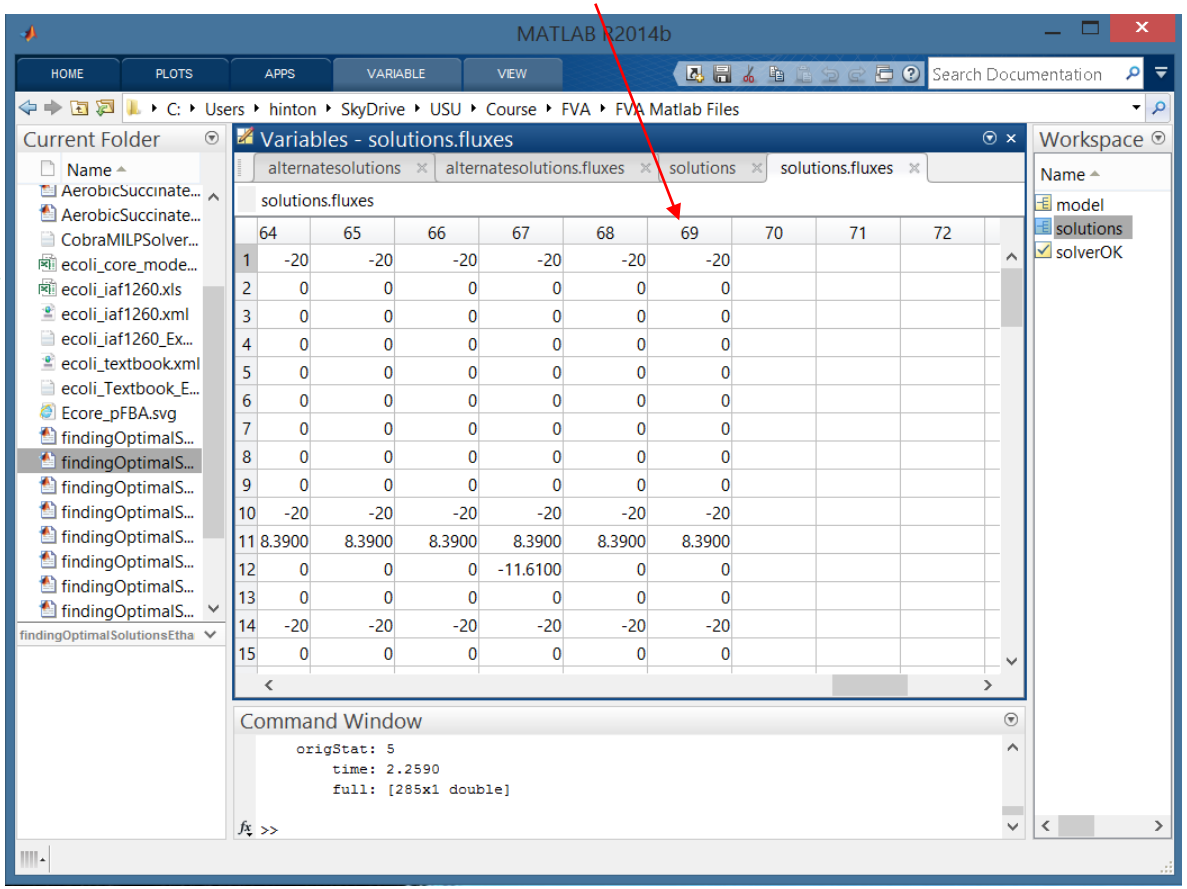


FVA Upper Bound; Lower Bound.

Flux Variability Analysis for Maximum Ethanol Production

```
% FluxVariabilityEthanol.m
clear; clc;
% Load the E.coli core model
model = readCbModel('ecoli_core_model.mat');
model = changeRxnBounds(model,'EX_glc(e)',0,'l');
model = changeRxnBounds(model,'EX_succ(e)',-20,'l');
model = changeObjective(model,'EX_etoh(e)'); % Optimize for maximum ethanol

% Perform flux variability analysis
[minFluxL,maxFluxL]=fluxVariability(model,100,'max',model.rxns,false,false)
% Print flux values
Difference = abs(maxFlux - minFlux);
FluxDifference = Difference;
n = length(Difference);
for i=1:n % Set small values of flux to zero
    if Difference(i) < 0.0001
        FluxDifference(i) = 0;
    end
end
printFluxVector(model, [minFlux, maxFlux, FluxDifference])
```

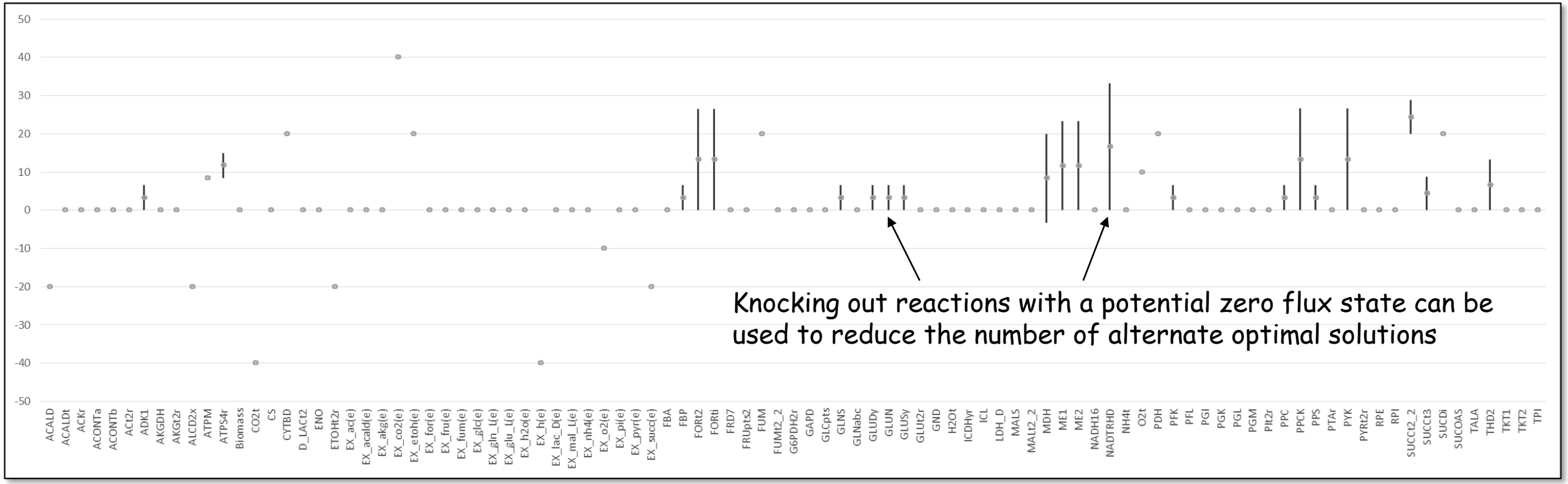
69 Alternate Optimal Flux Vectors



findingOptimalSolutionsEthanol.m

FVA Chart for Ethanol Production

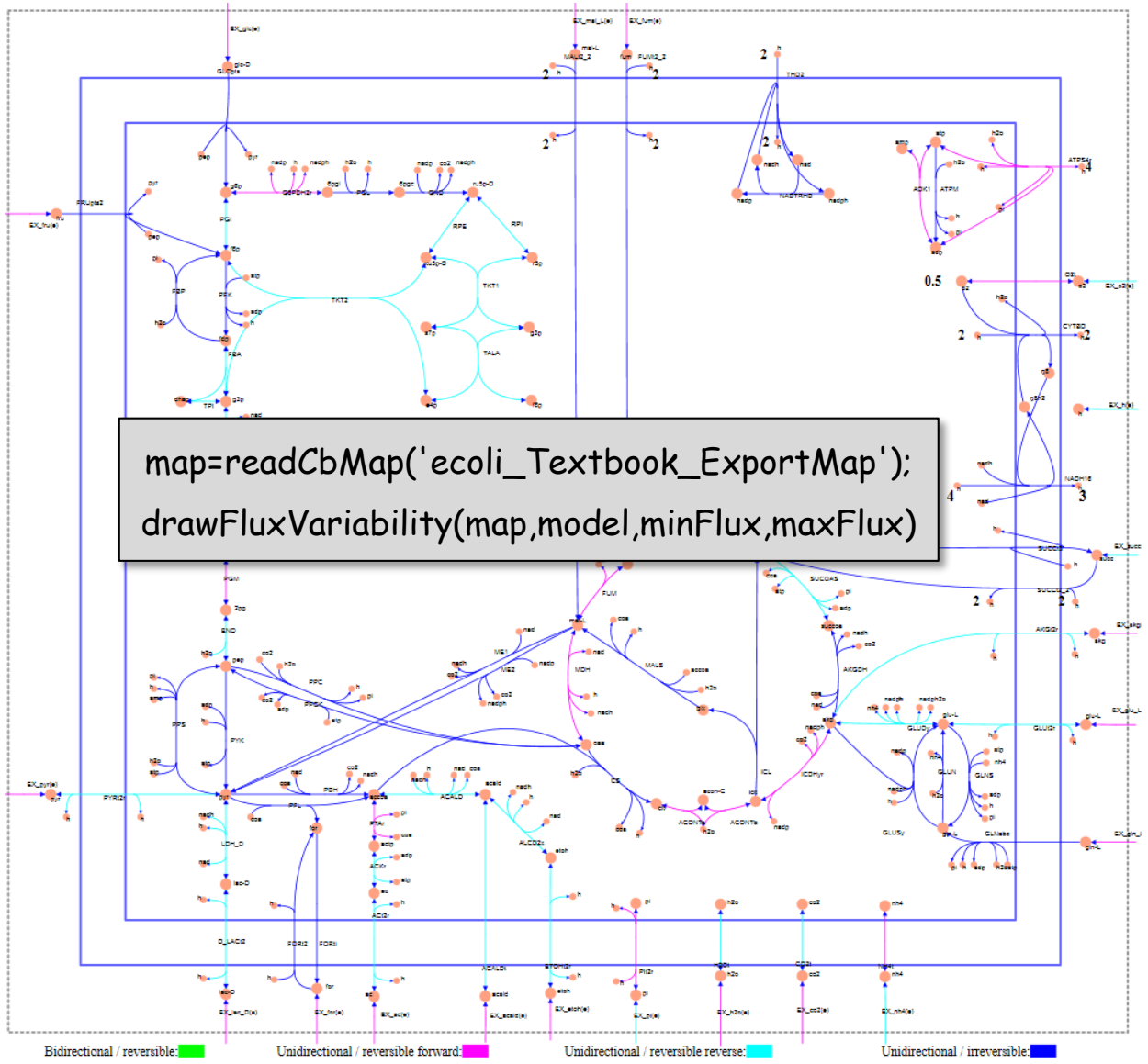
FVA Ethanol Production.xlsx



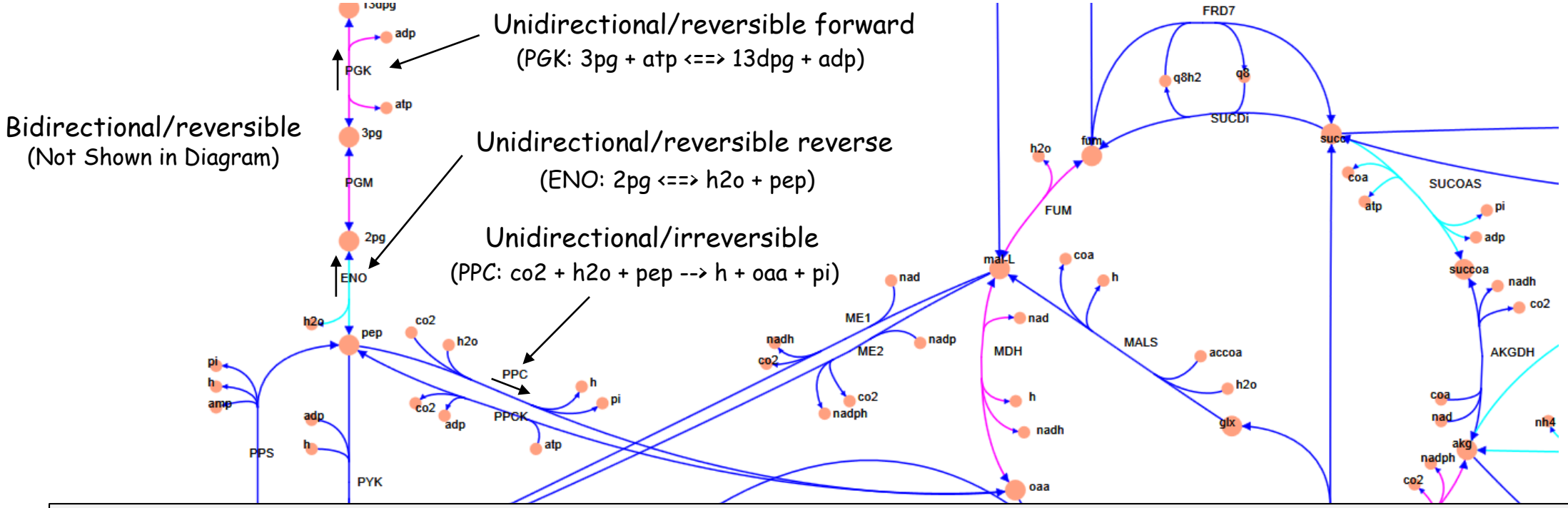
Flux Variability Map

FluxVariabilitySuccinate.m

```
map=readCbMap('ecoli_Textbook_ExportMap');
drawFluxVariability(map,model,minFlux,maxFlux)
```



Flux Variability Map – Close-up

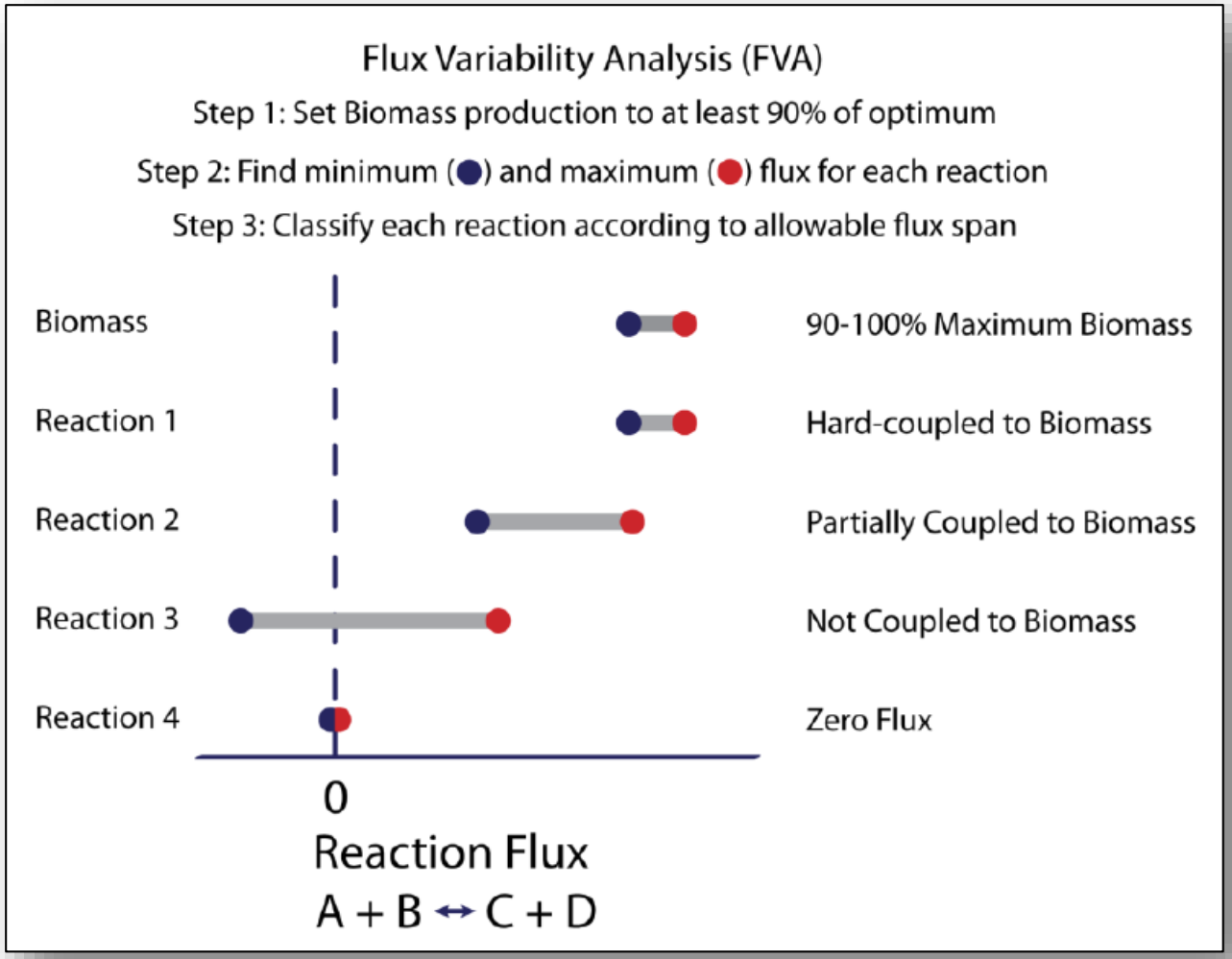


- Bidirectional/reversible = calculated flux change is bidirectional and changes directions/Stoichiometry is reversible
- Unidirectional/reversible forward = calculated flux change is unidirectional in direction of listed Stoichiometry/Stoichiometry is reversible
- Unidirectional/reversible reverse = calculated flux change is unidirectional in opposite direction of listed Stoichiometry/Stoichiometry is reversible
- Unidirectional/irreversible = calculated flux change is unidirectional/Stoichiometry is irreversible

FVA Classifications

FVA can be used to classify the reactions in a metabolic network. Assuming a biomass production rate greater than 90% of the optimal growth rate;

- A reaction was classified as "**Hard-coupled to biomass**" if the flux varied exactly with biomass production.
- "**Partially coupled to biomass**" included reactions that were required to have a non-zero flux, but were more flexible in the range.
- Reactions were classified as "**Not coupled to biomass**" if they could have a zero or non-zero flux while maintaining 90% biomass.
- Reactions were considered "**zero flux**" if they could maintain a flux in other conditions, but could not in growth conditions.



Lewis, N. E., K. K. Hixson, et al. (2010). "Omic data from evolved E. coli are consistent with computed optimal growth from genome-scale models-Supplementary Material." Molecular Systems Biology 6: 390.

FVA Classifications

```
% FVASuccinateClassificationSimple.m
clear; clc;

% Input the E.coli core model
model = readCbModel('ecoli_core_model.mat');

% Change carbon source from glucose to succinate
model = changeRxnBounds(model,'EX_glc(e)',0,'l');
model = changeRxnBounds(model,'EX_succ(e)',-20,'l');

% Set optimization objective to Biomass_Ecoli_core_N(w/GAM)-Nmet2
model = changeObjective(model,'Biomass_Ecoli_core_N(w/GAM)-Nmet2');

% Perform flux variability analysis classification
[minFlux,maxFlux]=fluxVariability(model,90,'max',model.rxns,false,false);

BioMassID = findRxnIDs(model, 'Biomass_Ecoli_core_N(w/GAM)-Nmet2');
BiomassRatio = minFlux(BioMassID)/maxFlux(BioMassID);

% Find hard-coupled reactions
HCReactions = {};
n = length(maxFlux);
j = 1;
for i=1:n
    if (minFlux(i) == BiomassRatio*maxFlux(i)) && (maxFlux(i) > 0)
        HCReactions(j) = model.rxns(i);
        j = j+1;
    end
end
HardCoupledReactions = transpose(HCReactions)

% Find partially-coupled reactions
PCReactions = {};
n = length(maxFlux);
j = 1;
for i=1:n
    if (minFlux(i) > 0 ) && (minFlux(i) < BiomassRatio*maxFlux(i))
        PCReactions(j) = model.rxns(i);
        j = j+1;
    end
end
PartiallyCoupledReactions = transpose(PCReactions)

% Find not-coupled reactions
NCReactions = {};
n = length(maxFlux);
j = 1;
for i=1:n
    if (minFlux(i) <= 0 ) && (minFlux(i) < maxFlux(i))
        NCReactions(j) = model.rxns(i);
        j = j+1;
    end
end
NotCoupledReactions = transpose(NCReactions)

% Find zero-flux reactions
ZFReactions = {};
n = length(maxFlux);
j = 1;
for i=1:n
    if (minFlux(i) == 0 ) && (maxFlux(i) == 0)
        ZFReactions(j) = model.rxns(i);
        j = j+1;
    end
end
ZeroFluxReactions = transpose(ZFReactions)
```

FVASuccinateClassificationSimple.m

FVA Classifications for Succinate Growth

Hard-Coupled Reactions

'Biomass_Ecoli_core_N(w/GAM)-Nmet2'

Partially-Coupled Reactions

'ACONTa'
'ACONTb'
'ATPS4r'
'Biomass
'CS'
'CYTBD'
'EX_co2(e)'
'EX_h2o(e)'
'FBP'
'FUM'
'GLNS'
'ICDHyr'
'MDH'
'NADH16'
'NH4t'
'O2t'
'PDH'
'PGK'
'PGM'
'PIt2r'
'SUCCt2_2'
'SUCDi'

Not-Coupled Reactions

'ACALD'
'ACALDt'
'ACKr'
'ACt2r'
'ADK1'
'AKGDH'
'AKGt2r'
'ALCD2x'
'CO2t'
'D_LACt2'
'ENO'
'ETOHt2r'
'EX_ac(e)'
'EX_acald(e)'
'EX_akg(e)'
'EX_etoh(e)'
'EX_for(e)'
'EX_glu_L(e)'
'EX_h(e)'
'EX_lac_D(e)'
'EX_nh4(e)'
'EX_o2(e)'
'EX_pi(e)'
'EX_pyr(e)'
'EX_succ(e)'
'FBA'
'FORt2'
'FORti'
'G6PDH2r'
'GAPD'
'GLUDy'
'GLUN'
'GLUSy'
'GLUt2r'
'GND'
'H2Ot'
'ICL'
'LDH_D'
'MALS'
'ME1'
'ME2'
'NADTRHD'
'PFK'
'PFL'
'PGI'
'PGL'
'PGL'
'PPC'
'PPCK'
'PPS'
'PTAr'
'PYK'
'PYRt2r'
'RPE'
'RPI'
'SUCCt3'
'SUCOAS'
'TALA'
'THD2'
'TKT1'
'TKT2'
'TPI'

Zero-flux Reactions

'EX_fru(e)'
'EX_fum(e)'
'EX_glc(e)'
'EX_gln_L(e)'
'EX_mal_L(e)'
'FRD7'
'FRUpts2'
'FUMt2_2'
'GLCpts'
'GLNabc'
'MALt2_2'

FVASuccinateClassification.m

Review Questions

- What is flux variability analysis?
- What is the relationship between the value of the objective function and the flux values calculated through flux variability analysis?
- How is flux variability analysis related to alternate optimal flux vectors?
- How can you implement flux variability analysis using the Cobra Toolbox?
- Does flux variability analysis identify the specific alternate optimal solutions?
- What is the value of knowing which reactions carry flux, which reactions carry no flux, and which reactions span a range of flux values?
- Explain the different FVA classifications; hard-coupled, partially-coupled, not-coupled, and no-flux reactions?

Lesson Outline

- Alternate Optimal Solutions
- Flux Variability Analysis
- Parsimonious FBA

singleRxnDeletion, singleGeneDeletion

- A cobra toolbox function that performs single reaction or gene deletion (knockout) analysis
  - ✓ `[grRatio,grRateKO,grRateWT,hasEffect,delRxns,hasEffect] = singleRxnDeletion(model)`
  - ✓ `[grRatio,grRateKO,grRateWT,delRxns,hasEffect] = singleGeneDeletion(model)`
- grRatio - Computed growth rate ratio between the model with a deleted reaction/gene and the original model without any deletions
- grRateKO - Growth rate of model with a reaction deletion/gene (1/h)
- grRateWT - Growth rate of the original model (1/h)
- hasEffect - Does a reaction deletion/gene affect anything
- delRxn - Deleted reactions/genes
- fluxSolution - FBA/MOMA/lMOMA fluxes for models with reaction/gene deletions
- Typically, if the grRatio is below a certain tolerance, tol, then the reaction/gene is categorized as essential

Essential Reactions

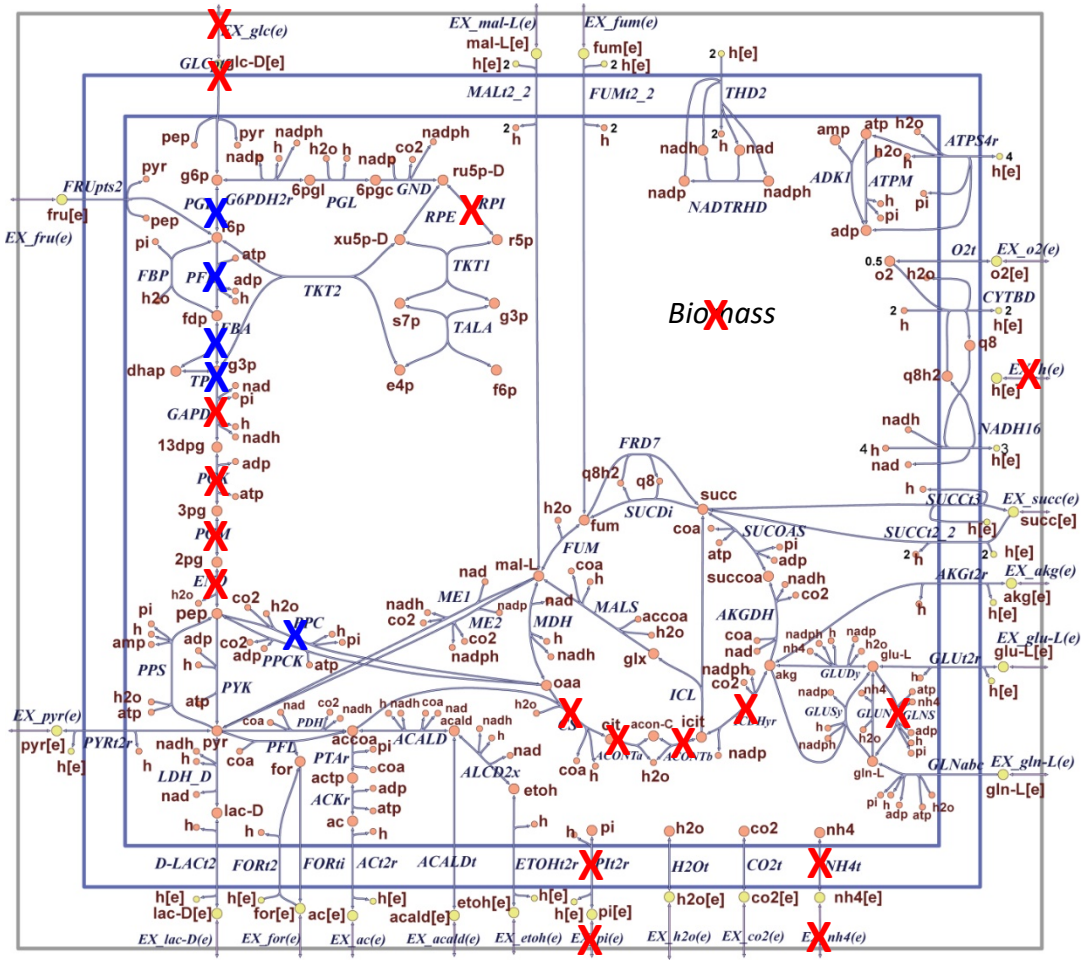
Essential reactions, metabolic genes necessary for *in silico* growth in the given media;

```
% EssentialReactions.m
clear; clc;

% Load the E.coli core model
model = readCbModel('ecoli_core_model.mat');

model = changeRxnBounds(model,'EX_glc(e)',-10,'l');
model = changeRxnBounds(model,'EX_o2(e)',-30,'l');

tol = 1e-6; % Growth rate lower limit
RxnRatio = singleRxnDeletion(model);
RxnRatio(isnan(RxnRatio))=0; % Replace NaN with 0
EssentialRxns = model.rxns(RxnRatio<tol)
```

*RxnRatio = Computed growth rate ratio between deletion strain and wild type

Aerobic
'ACONTa'
'ACONTb'
'Biomass
'CS'
'ENO'
'EX_glc(e)'
'EX_h(e)'
'EX_nh4(e)'
'EX_pi(e)'
'GAPD'
'GLCpts'
'GLNS'
'ICDHyr'
'NH4t'
'PGK'
'PGM'
'PIt2r'
'RPI'

Anaerobic
'ACONTa'
'ACONTb'
'Biomass'
'CS'
'ENO'
'EX_glc(e)'
'EX_h(e)'
'EX_nh4(e)'
'EX_pi(e)'
'FBA'
'GAPD'
'GLCpts'
'GLNS'
'ICDHyr'
'NH4t'
'PFK'
'PGI'
'PGK'
'PGM'
'PIt2r'
'PPC'
'RPI'
'TPI'

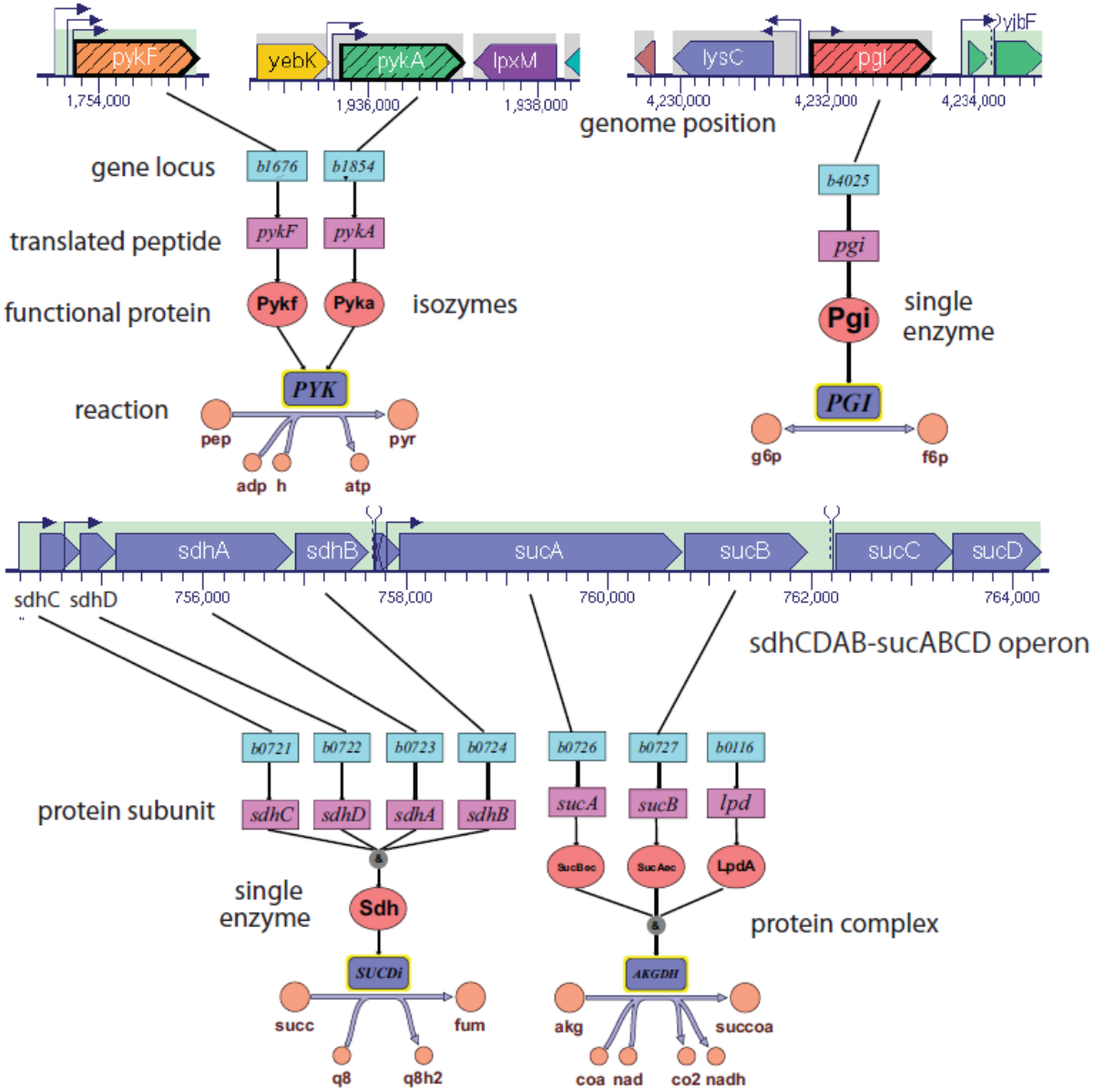


Essential Genes

Essential genes, metabolic genes necessary for *in silico* growth in the given media;

```
% EssentialGenes.m
clear; clc;


% Load the E.coli core model
model = readCbModel('ecoli_core_model.mat');

model = changeRxnBounds(model,'EX_glc(e)',-10,'l');
model = changeRxnBounds(model,'EX_o2(e)',-30,'l');

tol = 1e-6; % Growth rate lower limit
grRatio = singleGeneDeletion(model);
grRatio(isnan(grRatio))=0;
EssentialGenes = model.genes(grRatio<tol)

% List reactions associated with genes
[results ListResults] = findRxnsFromGenes(model, EssentialGenes);
```

Aerobic
'b0451'
'b0720'
'b1136'
'b1779'
'b2415'
'b2416'
'b2779'
'b2926'
Anaerobic
'b0451'
'b0720'
'b1136'
'b1779'
'b2415'
'b2416'
'b2779'
'b2926'
'b3919'
'b3956'
'b4025'



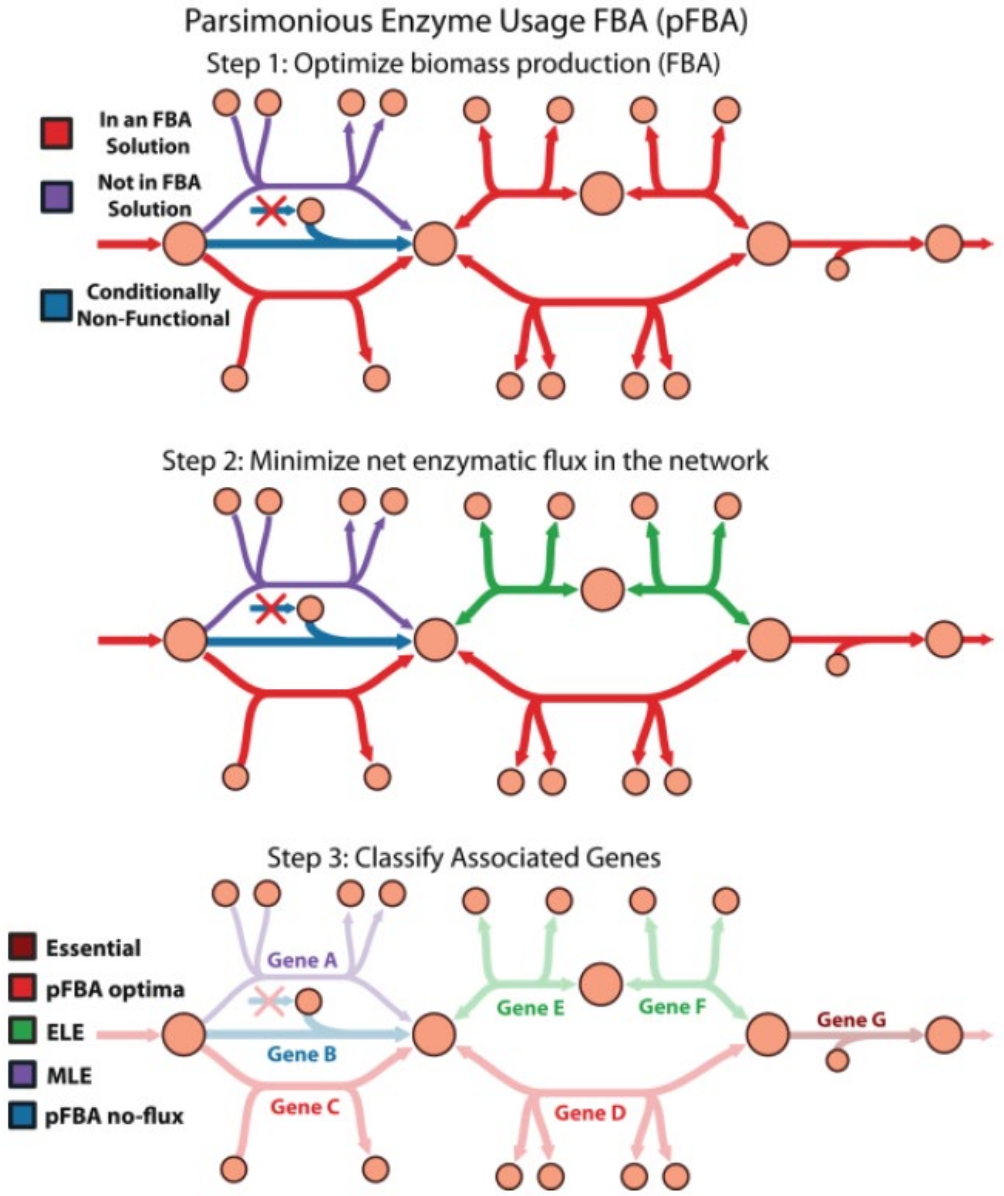
Parsimonious FBA

- Flux parsimony - minimize the total material flow required to achieve an objective.
- The underlying assumption is that, under growth pressure, there is a selection for strains that can process the growth substrate the most rapidly and efficiently while using the minimum amount of enzyme.
- Genes are classified into six categories:
  1. essential genes, metabolic genes necessary for *in silico* growth in the given media;
  2. pFBA optima, non-essential genes contributing to the optimal growth rate and minimum gene-associated flux;
  3. enzymatically less efficient (ELE), genes requiring more flux through enzymatic steps than alternative pathways that meet the same predicted growth rate;
  4. metabolically less efficient (MLE), genes requiring a growth rate reduction if used;
  5. pFBA no-flux, genes that are unable to carry flux in the experimental conditions; and
  6. Blocked, genes that are only associated with the reactions that cannot carry a flux under any condition ("blocked" reactions).
- A map showing the category of each gene can be created.

Lewis, N. E., K. K. Hixson, et al. (2010). "Omic data from evolved E. coli are consistent with computed optimal growth from genome-scale models." Molecular Systems Biology 6: 390.

Parsimonious Enzyme Usage

- Gene A, classified as MLE, represents an enzyme that uses a suboptimal co-factor to catalyze a reaction, thereby reducing the growth rate if used.
- Gene B, classified as pFBA no-flux, cannot carry a flux in this example since it is unable to take up or produce a necessary precursor metabolite.
- Genes E and F in this example require two different enzymes to catalyze the same transformation which Gene D can do alone; therefore they are classified as ELE.
- Gene G is essential, since its removal will stop the flux through all pathways.
- Genes C and D represent the most efficient (topologically and metabolically) pathway and therefore are part of the pFBA optima.



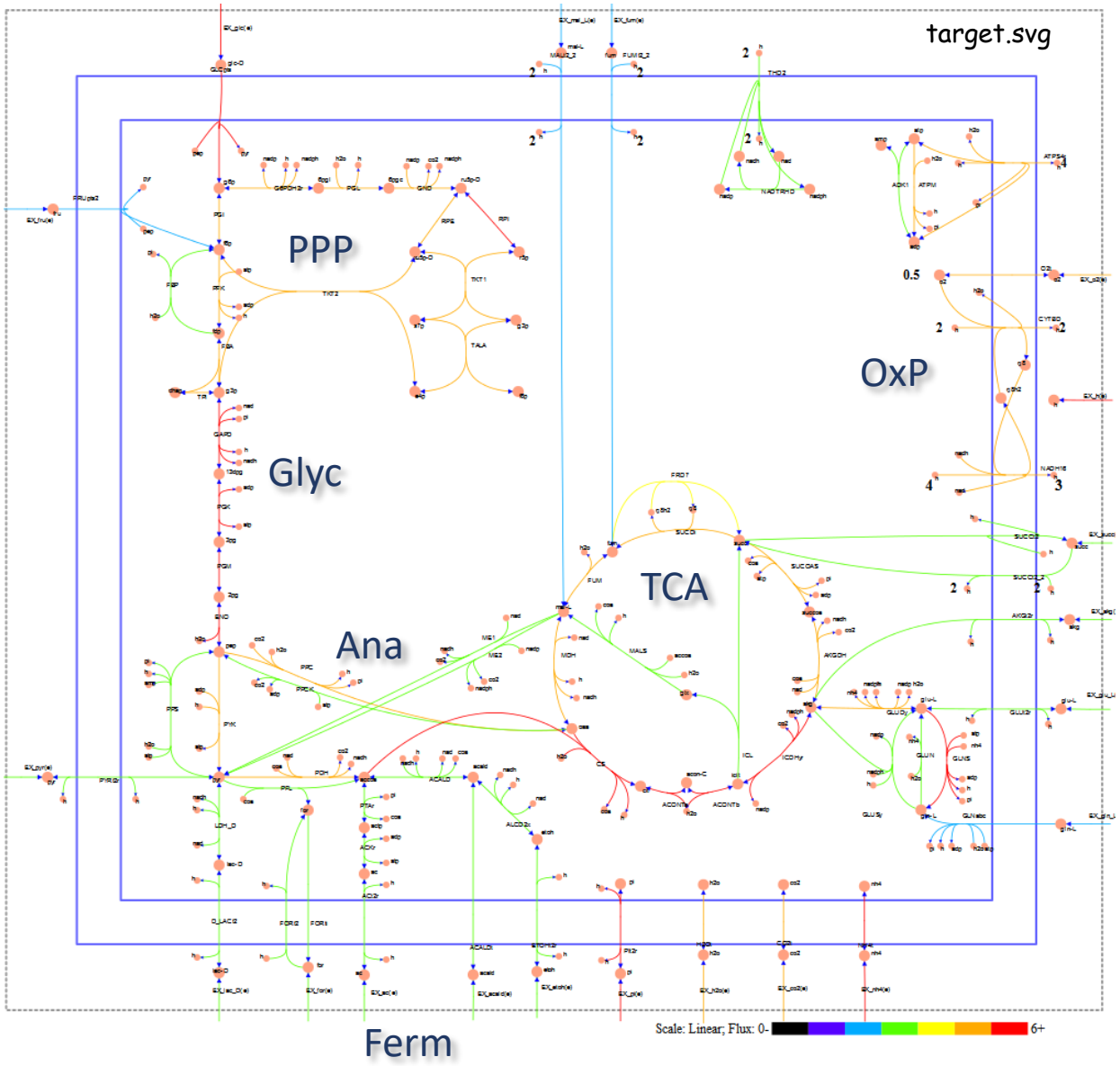
Lewis, N. E., K. K. Hixson, et al. (2010). "Omic data from evolved E. coli are consistent with computed optimal growth from genome-scale models." Molecular Systems Biology 6: 390.

Parsimonious FBA Example

pFBA_Ecoli_Core.m (Aerobic)

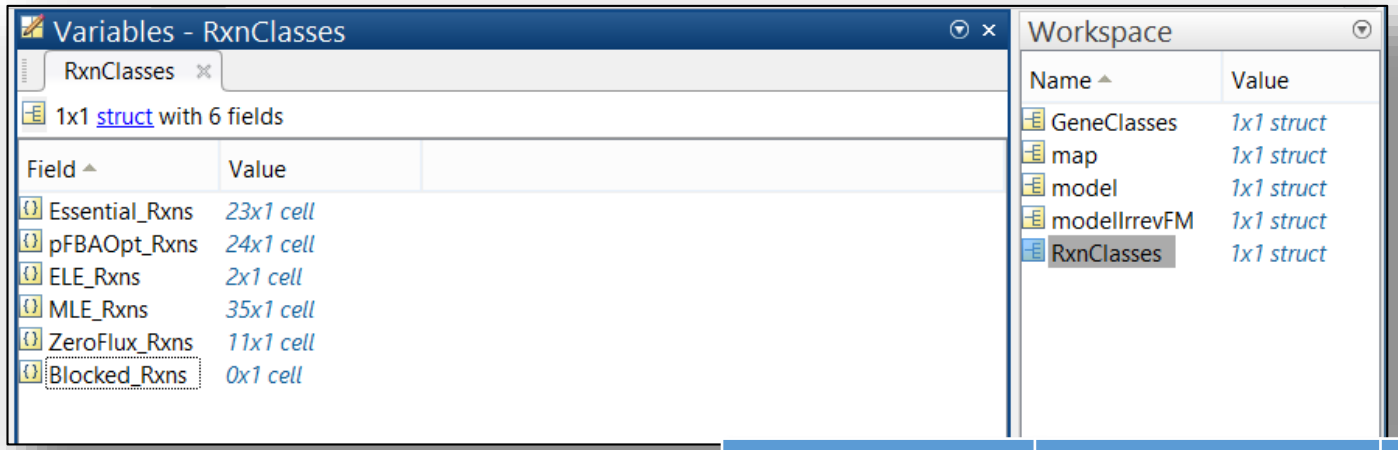
```
% pFBA_Ecoli_Core.m

clear;

model=readCbModel('ecoli_textbook');

model = changeRxnBounds(model,'EX_glc(e)',-10,'l');
model = changeRxnBounds(model,'EX_o2(e)',-0 or -30,'l');

map=readCbMap('ecoli_Textbook_ExportMap');

[GeneClasses RxnClasses modelIrrevFM] = pFBA(model,
          'geneoption',0, 'tol',1e-7)
```

Red = Essential reactions,
Orange = pFBA optima reaction
Yellow = ELE reactions,
Green = MLE reactions,
Blue = zero flux reactions,
Purple = blocked reactions,
Black = not classified



pFBA Classification

pFBA_Ecoli_Core.m (Anaerobic)



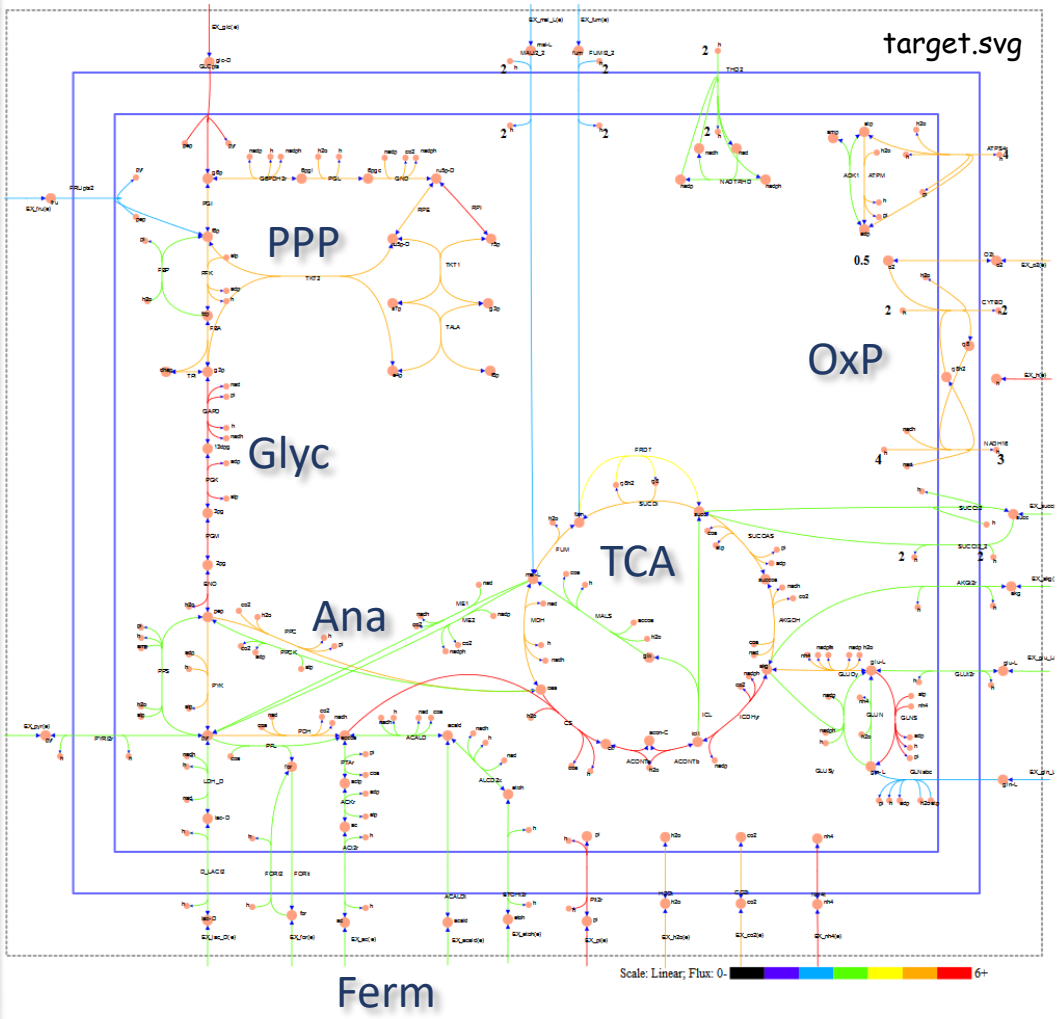
1. Essential genes, metabolic reactions/genes necessary for *in silico* growth in the given media;
2. pFBA optima, non-essential reactions/genes contributing to the optimal growth rate and minimum gene-associated flux;
3. Enzymatically Less Efficient (ELE), reactions/genes requiring more flux through enzymatic steps than alternative pathways that meet the same predicted growth rate;
4. Metabolically Less Efficient (MLE), reactions/genes requiring a growth rate reduction if used;
5. pFBA no-flux, reactions/genes that are unable to carry flux in the experimental conditions; and
6. Blocked, reactions/genes that are only associated with the reactions that cannot carry a flux under any condition ("blocked" reactions).

Essential		pFBA Optima		Enzymatically Less Efficient	Metabolically Less Efficient			pFBA No-flux	Blocked
'ACONTa'	'GLNS'	'ACALD'	'EX_h2o(e)'	'FRD7'	'ACALDt'	'FORt2'	'ME2'	'CYTBD'	
'ACONTb'	'ICDHyr'	'ACKr'	'FORti'	'SUCDi'	'ADK1'	'FUM'	'NADH16'	'EX_fru(e)'	
Biomass'	'NH4t'	'ACt2r'	'GLUDy'		'AKGDH'	'G6PDH2r'	'NADTRHD'	'EX_fum(e)'	
'CS'	'PFK'	'ALCD2x'	'H2Ot'		'AKGt2r'	'GLUN'	'PDH'	'EX_gln_L(e)'	
'ENO'	'PGI'	'ATPM'	'PFL'		'D_LACt2'	'GLUSy'	'PGL'	'EX_mal_L(e)'	
'EX_glc(e)'	'PGK'	'ATPS4r'	'PTAr'		'EX_acald(e)'	'GLUt2r'	'PPCK'	'EX_o2(e)'	
'EX_h(e)'	'PGM'	'CO2t'	'PYK'		'EX_akg(e)'	'GND'	'PPS'	'FRUpts2'	
'EX_nh4(e)'	'PIt2r'	'ETOHt2r'	'RPE'		'EX_glu_L(e)'	'ICL'	'PYRt2r'	'FUMt2_2'	
'EX_pi(e)'	'PPC'	'EX_ac(e)'	'TALA'		'EX_lac_D(e)'	'LDH_D'	'SUCCt2_2'	'GLNabc'	
'FBA'	'RPI'	'EX_co2(e)'	'THD2'		'EX_pyr(e)'	'MALS'	'SUCCt3'	'MALt2_2'	
'GAPD'	'TPI'	'EX_etoh(e)'	'TKT1'		'EX_succ(e)'	'MDH'	'SUCOAS'	'O2t'	
'GLCpts'		'EX_for(e)'	'TKT2'		'FBP'	'ME1'			

Parsimonious FBA Data

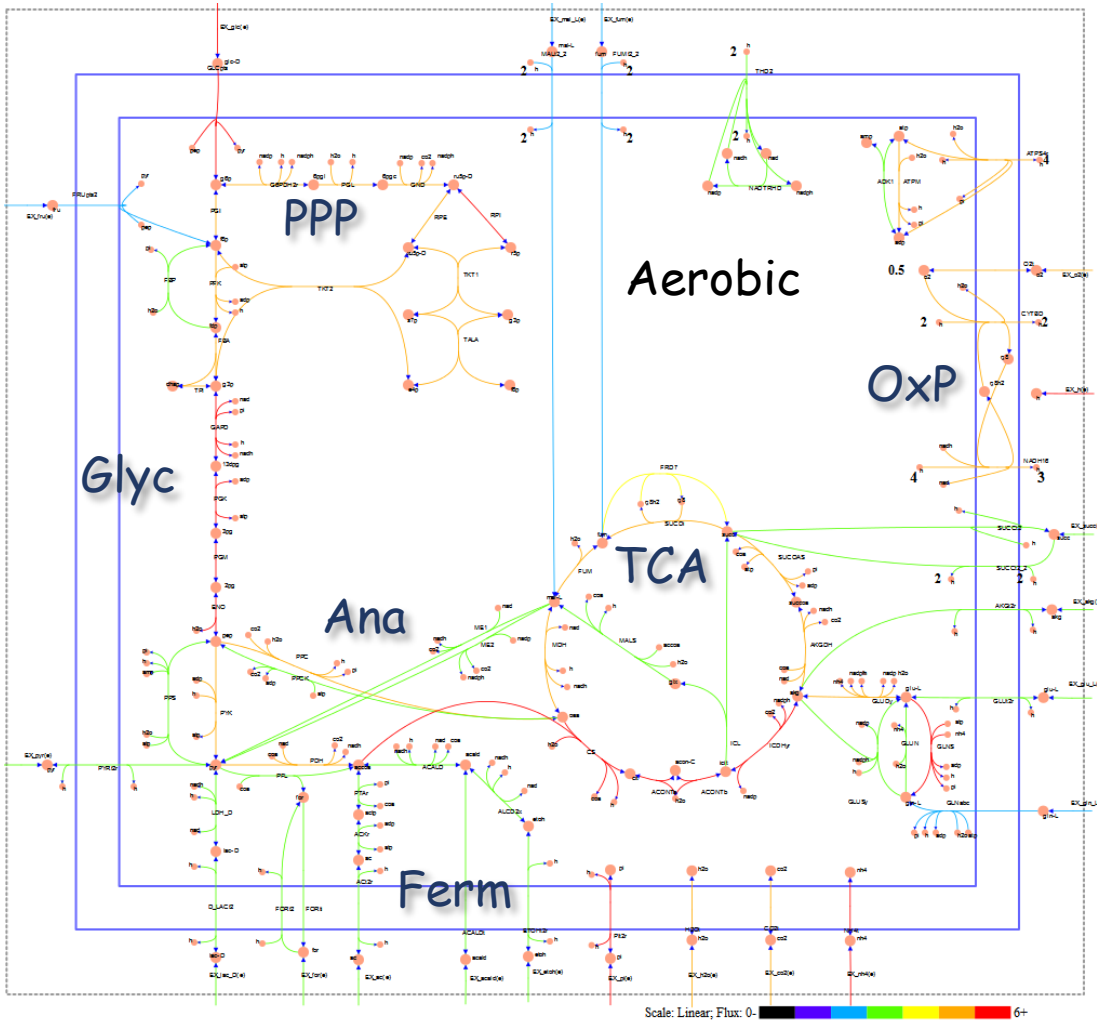
Essential	pFBA Optima	Enzymatically Less Efficient	Metabolically Less Efficient		pFBA No-flux	Blocked
'ACONTa'	'ACALD'	'FRD7'	'ACALDt'	'ME2'	'CYTBD'	
'ACONTb'	'ACKr'	'SUCDi'	'ADK1'	'NADH16'	'EX_fru(e)'	
Biomass'	'ACt2r'		'AKGDH'	'NADTRHD'	'EX_fum(e)'	
'CS'	'ALCD2x'		'AKGt2r'	'PDH'	'EX_gln_L(e)'	
'ENO'	'ATPM'		'D_LACt2'	'PGL'	'EX_mal_L(e)'	
'EX_glc(e)'	'ATPS4r'		'EX_acald(e)'	'PPCK'	'EX_o2(e)'	
'EX_h(e)'	'CO2t'		'EX_akg(e)'	'PPS'	'FRUpts2'	
'EX_nh4(e)'	'ETOHt2r'		'EX_glu_L(e)'	'PYRt2r'	'FUMt2_2'	
'EX_pi(e)'	'EX_ac(e)'		'EX_lac_D(e)'	'SUCCt2_2'	'GLNabc'	
'FBA'	'EX_co2(e)'		'EX_pyr(e)'	'SUCCt3'	'MALt2_2'	
'GAPD'	'EX_etoh(e)'		'EX_succ(e)'	'SUCOAS'	'O2t'	
'GLCpts'	'EX_for(e)'		'FBP'			
'GLNS'	'EX_h2o(e)'		'FORt2'			
'ICDHyr'	'FORti'		'FUM'			
'NH4t'	'GLUDy'		'G6PDH2r'			
'PFK'	'H2Ot'		'GLUN'			
'PGI'	'PFL'		'GLUSy'			
'PGK'	'PTAr'		'GLUt2r'			
'PGM'	'PYK'		'GND'			
'PIt2r'	'RPE'		'ICL'			
'PPC'	'TALA'		'LDH_D'			
'RPI'	'THD2'		'MALS'			
'TPI'	'TKT1'		'MDH'			
	'TKT2'		'ME1'			

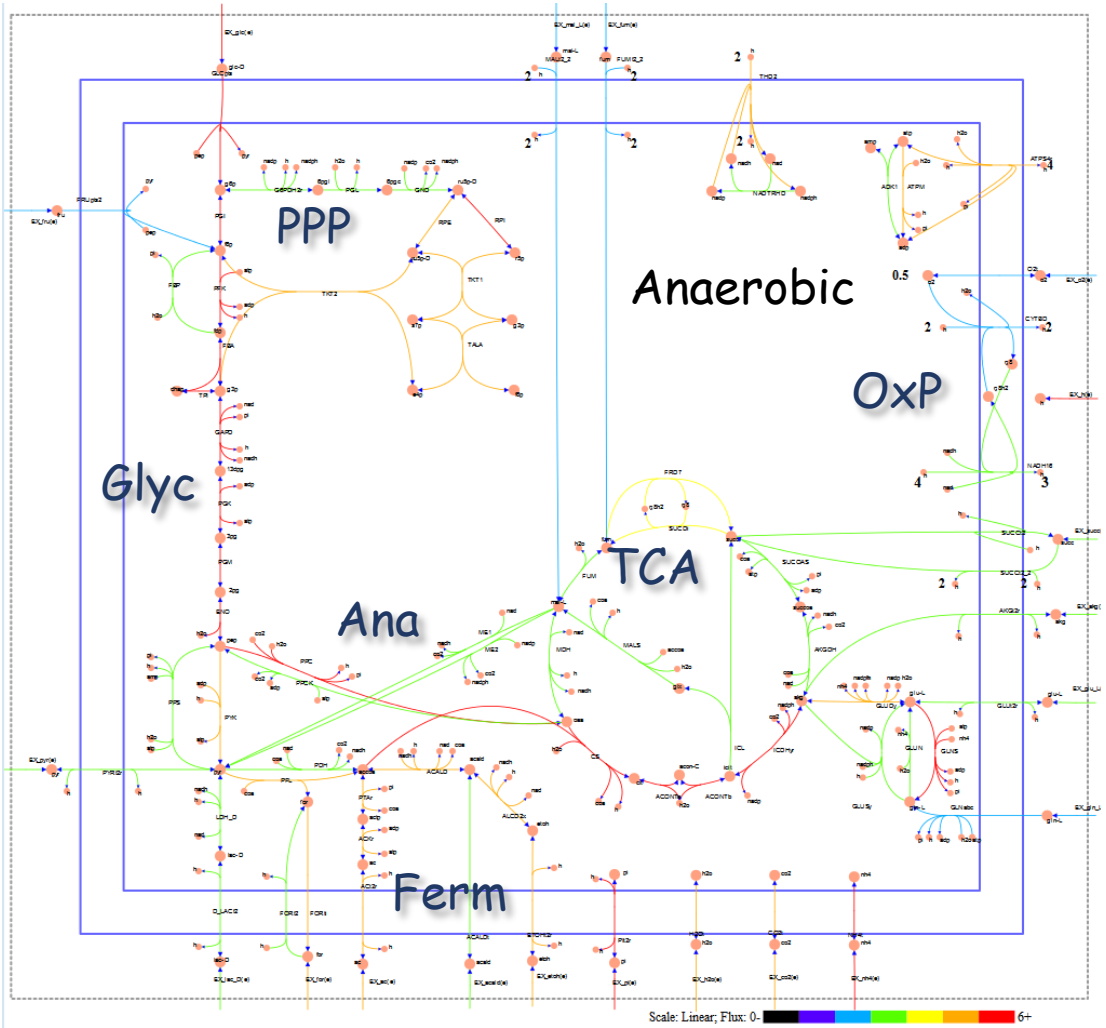
pFBA_Ecoli_Core.m (Anaerobic)



Parsimonious FBA Maps

pFBA_Ecoli_Core.m

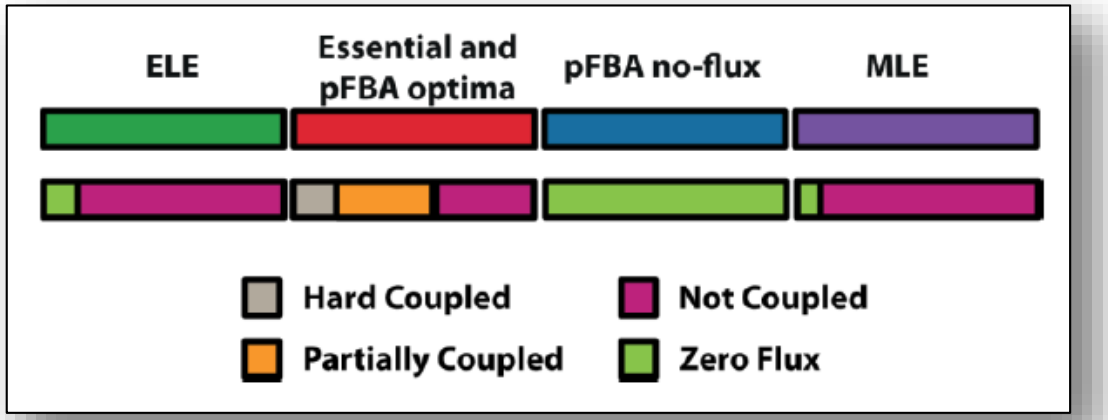




pFBA vs. FVA Reaction Classes

The pFBA genes were mapped to the FVA reaction classes.

- The hard-coupled and partially-coupled reactions were all associated with the essential and pFBA optima genes, and the
- pFBA no-flux genes were all within the FVA zero-flux reactions.
- FVA Zero Flux reactions were identified in the other pFBA classes since some Zero-Flux reactions are catalyzed by genes which may be active for alternative, functional reactions.



Lewis, N. E., K. K. Hixson, et al. (2010). "Omic data from evolved E. coli are consistent with computed optimal growth from genome-scale models-Supplementary Material." Molecular Systems Biology 6: 390.

Review Questions

- Why do they call it parsimonious flux balance analysis?
- What are essential genes/reactions?
- What are pFBA optima genes/reactions?
- What are enzymatically less efficient (ELE) genes/reactions?
- What are metabolically less efficient genes/reactions?
- What are pFBA no-flux genes/reactions?
- What are blocked genes/reactions?
- What is the difference between pFBA optima genes/reactions, enzymatically less efficient (ELE) genes/reactions and metabolically less efficient (MLE), genes/reactions?
- How can you implement parsimonious flux balance analysis using the Cobra Toolbox
- How can parsimonious flux balance analysis be used to metabolically engineer a cell?

Lesson Outline

- Alternate Optimal Solutions
- Flux Variability Analysis
- Parsimonious FBA

New Cobra Toolbox Functions

```
% Changing solver type
changeCobraSolver('glpk','all')

% Finding alternate optimal solutions
[solutions] = enumerateOptimalSolutions(model); % Finding alternate optimal solutions

% Flux Variability Analysis
[minFlux,maxFlux]=fluxVariability(model,100,'max',model.rxns,false,false); % Flux Variability Analysis

% Single reaction deletion
[grRatio,grRateKO,grRateWT,hasEffect,delRxns,hasEffect] = singleRxnDeletion(model)

% Single gene deletion
[grRatio,grRateKO,grRateWT,delRxns,hasEffect] = singleGeneDeletion(model)

% Parsimonious Flux Balance Analysis
[GeneClasses RxnClasses modelIrrevFM] = pFBA(model,'geneoption',0, 'tol',1e-7)
```

References

Flux Variability Analysis

1. Reed, J. L. & Palsson, B. Ø. Genome-scale in silico models of *E. coli* have multiple equivalent phenotypic states: assessment of correlated reaction subsets that comprise network states. Genome Res. 14, 1797–1805 (2004).
2. Mahadevan, R. and C. H. Schilling (2003). "The effects of alternate optimal solutions in constraint-based genome-scale metabolic models." Metabolic engineering 5(4): 264-276
3. Phalakornkule, C. et al. A MILP-based flux alternative generation and NMR experimental design strategy for metabolic engineering. Metab. Eng. 3, 124–137 (2001).
4. Lee, S., Phalakornkule, C., Domach, M. M. & Grossmann, I. E. Recursive MILP model for finding all the alternate optima in LP models for metabolic networks.Comp. Chem. Eng. 24, 711–716 (2000).

Parsimonious Analysis

1. Lewis, N. E., K. K. Hixson, et al. (2010). "Omic data from evolved *E. coli* are consistent with computed optimal growth from genome-scale models." Molecular Systems Biology 6: 390.