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.

Maximize the objective function

\[ Z = \sum_i c_i v_i^k = c \cdot v^k \]

with the following constraints

\[ \frac{dx}{dt} = S \cdot v^k = 0 \]

\[ \alpha_j \leq v_j^k \leq \beta_j \]

\[ 1 \leq k \leq n \]

⇒ \( v^1, v^2, ..., v^n \) all have the same value of \( Z \)

Identifying Alternate Equivalent Optimal Solutions

- A function 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.

```matlab
% 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

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

---

**AOS_Example.xlsx**
Reactions with Changing Flux Identified Through Alternate Optimal Solutions for Growth on Succinate

<table>
<thead>
<tr>
<th>Reaction</th>
<th>Minimum Flux (mmol gDW⁻¹ hr⁻¹)</th>
<th>Maximum Flux (mmol gDW⁻¹ hr⁻¹)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MDH</td>
<td>13.56</td>
<td>20.06</td>
</tr>
<tr>
<td>ME1</td>
<td>0</td>
<td>6.49</td>
</tr>
<tr>
<td>ME2</td>
<td>7.17</td>
<td>13.67</td>
</tr>
<tr>
<td>NADTRHD</td>
<td>0</td>
<td>6.49</td>
</tr>
<tr>
<td>PPCK</td>
<td>3.93</td>
<td>10.42</td>
</tr>
<tr>
<td>PYK</td>
<td>0</td>
<td>6.49</td>
</tr>
</tbody>
</table>
Reactions with Changing Flux Identified Through Alternate Optimal Solutions

- **MDH** (malate dehydrogenase)
  - \( \text{mal-L} + \text{nad} \iff \text{h} + \text{nadh} + \text{oaa} \)
- **ME1** (malic enzyme (NAD))
  - \( \text{mal-L} + \text{nad} \rightarrow \text{co2} + \text{nadh} + \text{pyr} \)
- **ME2** (malic enzyme (NADP))
  - \( \text{mal-L} + \text{nadp} \rightarrow \text{co2} + \text{nadph} + \text{pyr} \)
- **NADTRHD** (NAD transhydrogenase)
  - \( \text{nad} + \text{nadph} \rightarrow \text{nadh} + \text{nadp} \)
- **PPCK** (phosphoenolpyruvate carboxykinase)
  - \( \text{atp} + \text{oaa} \rightarrow \text{adp} + \text{co2} + \text{pep} \)
- **PYK** (pyruvate kinase)
  - \( \text{adp} + \text{h} + \text{pep} \rightarrow \text{atp} + \text{pyr} \)

All reactions are centered around energy and reducing power production.
Energy Production: Electron Transport Chain

Need to maintain the same amount of flux through the electron transport chain.

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

http://sbrg.ucsd.edu/Publications/Books/SB2LectureSlides
Visualizing the Alternate Optimal Solution Flux Vectors

```matlab
% 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);
```
Alternate Optimal Solutions #1: Changing Fluxes

findingOptimalSolutionsSuccVisualize.m

ACONTa 8.13764
ACONTb 8.13764
AKGDH 7.23122
ATPM 8.39
ATPS4r 57.7816
ATPM 8.39
Biomass 0.840134
CO2t -44.2477
CS 8.13764
CYTBD 66.5528
ENO -3.49017
EX_co2(e) 44.2477
EX_h2o(e) 30.3675
EX_h(e) -23.1469
EX_nh4(e) -4.58108
EX_o2(e) -33.2764
EX_pi(e) -3.0906
EX_succ(e) -20

FBA -0.835681
FBP 0.835681
FUM 27.2312
GAPD -2.23333
GLNS 0.214822
GLUDy -4.36626
H2Ot -30.3675
ICDHyry 8.13764
MDH 13.565
ME1 0
ME2 13.6662
NADH16 39.3216
NADTRHD 6.49242
NADH 9.36626
NH4t 4.58108
O2t 33.2764
PDH 11.2863
PGI -0.172228
PGK 2.23333
PGM 3.49017
PIt2r 3.0906
PPCK 3.92628
RPE -0.603888
RPI -0.603888
SUCt2_2 20
SUCdi 27.2312
SUCA 7.2312
TALA -0.1503
TKT1 -0.1503
TKT2 -0.453588
TPI -0.835681
Other AOS Reactions
ME1 0
PYK 0

---

Constraint-based Metabolic Reconstructions & Analysis

Lesson: Flux Variability Analysis & Parsimonious Flux Balance Analysis

2019 H. Scott Hinton

Utah State University

BIE 5500/6500
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 (-2.233292), Bd: -1000 / 1000, glyceraldehyde-3-phosphate dehydrogenase, g3p[c] + nad[c] + p[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-Oxoglutarate 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

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

#68  NADTRHD (6.492425), Bd: 0 / 1000, NAD transhydrogenase, nadh[c] + nadph[c] → nadh[c] + succoa[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:


#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

#75  PGK (2.233292), Bd: -1000 / 1000, phosphoglycerate kinase, 3pg[c] + atp[c] → 13dpg[c] + adp[c]

#80  PPCK (3.926283), Bd: 0 / 1000, phosphoenolpyruvate carboxykinase, atp[c] + ooa[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] + p[c] → atp[c] + h2o[c] + 3 h[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
ENO    -3.49017 ME1   6.49242
EX_co2(e) 44.2477 ME2   7.1738
EX_h2o(e) 30.3675 NADH16 39.3216
EX_h(e)  -23.1469 NH4t  4.58108
EX_nh4(e) -4.58108 O2t   33.2764
EX_o2(e)  -33.2764 PDH  11.2863
EX_pi(e)  -3.0906 PGI  -0.172228
EX_succ(e) -20  PGK   2.23333

Other AOS Reactions

NADTRHD 0
PYK 0
Alternate Optimal Solutions #3: Changing Fluxes

findingOptimalSolutionsSuccVisualize.m

<table>
<thead>
<tr>
<th>Reaction</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACONTa</td>
<td>8.13764</td>
</tr>
<tr>
<td>ACONTb</td>
<td>8.13764</td>
</tr>
<tr>
<td>AKGDH</td>
<td>7.23122</td>
</tr>
<tr>
<td>ATPM</td>
<td>8.39</td>
</tr>
<tr>
<td>ATPS4r</td>
<td>57.7816</td>
</tr>
<tr>
<td>CO2t</td>
<td>-44.2477</td>
</tr>
<tr>
<td>CS</td>
<td>8.13764</td>
</tr>
<tr>
<td>CYTBD</td>
<td>66.5528</td>
</tr>
<tr>
<td>ENO</td>
<td>-3.49017</td>
</tr>
<tr>
<td>EX_co2(e)</td>
<td>44.2477</td>
</tr>
<tr>
<td>EX_h2o(e)</td>
<td>30.3675</td>
</tr>
<tr>
<td>EX_h(e)</td>
<td>-23.1469</td>
</tr>
<tr>
<td>EX_nh4(e)</td>
<td>-4.58108</td>
</tr>
<tr>
<td>EX_o2(e)</td>
<td>-33.2764</td>
</tr>
<tr>
<td>EX_pi(e)</td>
<td>-3.0906</td>
</tr>
<tr>
<td>EX_succ(e)</td>
<td>-20</td>
</tr>
</tbody>
</table>

Other AOS Reactions
- NADTRHD 0
- ME1 0
- ME1RHD 0
- Other AOS Reactions
- NADTRHD 0
- ME1 0
- ME1RHD 0

Utah State University  BIE 5500/6500  Lesson: Flux Variability Analysis & Parsimonious Flux Balance Analysis
Alternate Optimal Solutions

All three solutions produce the same amount of ATP by providing the same amount of NADH for the electron transport chain.
Reducing Alternate Optimal Solutions Code

```matlab
% 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);
```
Reducing Alternate Optimal Solutions

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

% 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
### Flux Variability Analysis Example Output (No Loops)

<table>
<thead>
<tr>
<th>Reaction</th>
<th>Lower</th>
<th>Upper</th>
<th>Difference</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACALD</td>
<td>-5.07E-06</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>ACALDt</td>
<td>-5.07E-06</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>ACKr</td>
<td>-7.76E-06</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>ACONt</td>
<td>8.13763</td>
<td>8.13764</td>
<td>0</td>
</tr>
<tr>
<td>ACONtb</td>
<td>8.13763</td>
<td>8.13764</td>
<td>0</td>
</tr>
<tr>
<td>ActOr</td>
<td>-7.76E-06</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>ADK1</td>
<td>0</td>
<td>3.30E-05</td>
<td>0</td>
</tr>
<tr>
<td>AKGDH</td>
<td>7.23119</td>
<td>7.23122</td>
<td>0</td>
</tr>
<tr>
<td>AKGt2r</td>
<td>-3.07E-06</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>ALCD2x</td>
<td>-4.40E-06</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>ATPM</td>
<td>8.39</td>
<td>8.39</td>
<td>0</td>
</tr>
<tr>
<td>ATPS4r</td>
<td>57.7816</td>
<td>57.7817</td>
<td>0</td>
</tr>
<tr>
<td>Biomass</td>
<td>0.840134</td>
<td>0.840134</td>
<td>0</td>
</tr>
<tr>
<td>CO2t</td>
<td>-44.2477</td>
<td>-44.2477</td>
<td>0</td>
</tr>
<tr>
<td>CS</td>
<td>8.13763</td>
<td>8.13764</td>
<td>0</td>
</tr>
<tr>
<td>CYTBD</td>
<td>66.5528</td>
<td>66.5528</td>
<td>0</td>
</tr>
<tr>
<td>D_LACT2</td>
<td>-4.25E-06</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>ENO</td>
<td>-3.49018</td>
<td>-3.49017</td>
<td>0</td>
</tr>
<tr>
<td>ETOHt2r</td>
<td>-4.40E-06</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>EX_ac(e)</td>
<td>0</td>
<td>7.76E-06</td>
<td>0</td>
</tr>
<tr>
<td>EX_acald(e)</td>
<td>0</td>
<td>5.07E-06</td>
<td>0</td>
</tr>
<tr>
<td>EX_akk(e)</td>
<td>0</td>
<td>3.07E-06</td>
<td>0</td>
</tr>
<tr>
<td>EX_co2(e)</td>
<td>44.2477</td>
<td>44.2477</td>
<td>0</td>
</tr>
<tr>
<td>EX_etoH(e)</td>
<td>0</td>
<td>4.40E-06</td>
<td>0</td>
</tr>
<tr>
<td>EX_for(e)</td>
<td>0</td>
<td>2.20E-05</td>
<td>0</td>
</tr>
<tr>
<td>EX_glu_L(e)</td>
<td>0</td>
<td>2.75E-06</td>
<td>0</td>
</tr>
<tr>
<td>EX_h2o(e)</td>
<td>30.3675</td>
<td>30.3675</td>
<td>0</td>
</tr>
<tr>
<td>EX_H(e)</td>
<td>-23.1469</td>
<td>-23.1469</td>
<td>0</td>
</tr>
</tbody>
</table>

### Reaction Output Example

<table>
<thead>
<tr>
<th>Reaction</th>
<th>Lower</th>
<th>Upper</th>
<th>Difference</th>
</tr>
</thead>
<tbody>
<tr>
<td>EX_lac_D(e)</td>
<td>0</td>
<td>4.25E-06</td>
<td>0</td>
</tr>
<tr>
<td>EX_nh4(e)</td>
<td>-4.58109</td>
<td>-4.58108</td>
<td>0</td>
</tr>
<tr>
<td>EX_02(e)</td>
<td>-33.2764</td>
<td>-33.2764</td>
<td>0</td>
</tr>
<tr>
<td>EX_pi(e)</td>
<td>-3.0906</td>
<td>-3.0906</td>
<td>0</td>
</tr>
<tr>
<td>EX_pyr(e)</td>
<td>0</td>
<td>5.07E-06</td>
<td>0</td>
</tr>
<tr>
<td>EX_succ(e)</td>
<td>-20</td>
<td>-20</td>
<td>0</td>
</tr>
<tr>
<td>FBA</td>
<td>-0.8357</td>
<td>-0.83568</td>
<td>0</td>
</tr>
<tr>
<td>FBP</td>
<td>0.835681</td>
<td>0.835714</td>
<td>0</td>
</tr>
<tr>
<td>FORt2</td>
<td>0</td>
<td>0.000132</td>
<td>0.00013187</td>
</tr>
<tr>
<td>FORt1</td>
<td>0</td>
<td>0.000132</td>
<td>0.00013187</td>
</tr>
<tr>
<td>FUM</td>
<td>27.2312</td>
<td>27.2312</td>
<td>0</td>
</tr>
<tr>
<td>G6PDH2r</td>
<td>0</td>
<td>4.40E-05</td>
<td>0</td>
</tr>
<tr>
<td>GAPD</td>
<td>-2.23334</td>
<td>-2.23E+00</td>
<td>0</td>
</tr>
<tr>
<td>GLNS</td>
<td>0.214822</td>
<td>0.214855</td>
<td>0</td>
</tr>
<tr>
<td>GLUDy</td>
<td>-4.36626</td>
<td>-4.36623</td>
<td>0</td>
</tr>
<tr>
<td>GLUN</td>
<td>0</td>
<td>3.30E-05</td>
<td>0</td>
</tr>
<tr>
<td>GLUSy</td>
<td>0</td>
<td>3.30E-05</td>
<td>0</td>
</tr>
<tr>
<td>GLUT2r</td>
<td>-2.75E-06</td>
<td>0.00E+00</td>
<td>0</td>
</tr>
<tr>
<td>GND</td>
<td>0.00E+00</td>
<td>4.40E-05</td>
<td>0</td>
</tr>
<tr>
<td>H2Ot</td>
<td>-30.3675</td>
<td>-3.04E+01</td>
<td>0</td>
</tr>
<tr>
<td>ICDHyr</td>
<td>8.13761</td>
<td>8.13764</td>
<td>0</td>
</tr>
<tr>
<td>ICL</td>
<td>0</td>
<td>3.30E-05</td>
<td>0</td>
</tr>
<tr>
<td>LDH_D</td>
<td>-4.25E-06</td>
<td>0.00E+00</td>
<td>0</td>
</tr>
<tr>
<td>MALs</td>
<td>0.00E+00</td>
<td>3.30E-05</td>
<td>0</td>
</tr>
<tr>
<td>MDH</td>
<td>13.565</td>
<td>2.01E+01</td>
<td>6.49253</td>
</tr>
<tr>
<td>ME1</td>
<td>0</td>
<td>6.4925</td>
<td>6.4925</td>
</tr>
<tr>
<td>ME2</td>
<td>7.17372</td>
<td>13.6663</td>
<td>6.49254</td>
</tr>
<tr>
<td>NADH16</td>
<td>39.3216</td>
<td>39.3216</td>
<td>0</td>
</tr>
</tbody>
</table>

### Additional Output

- **FVA_example1.xlsx**

---

**Constraint-based Metabolic Reconstructions & Analysis**

2019 H. Scott Hinton

Lesson: Flux Variability Analysis & Parsimonious Flux Balance Analysis
Flux Variability Analysis Example

FVA core.xlsx
Flux Variability Analysis Example (Continued)

<table>
<thead>
<tr>
<th>Reaction</th>
<th>Minimum Flux (mmol gDW(^{-1}) hr(^{-1}))</th>
<th>Maximum Flux (mmol gDW(^{-1}) hr(^{-1}))</th>
</tr>
</thead>
<tbody>
<tr>
<td>MDH</td>
<td>13.56</td>
<td>20.06</td>
</tr>
<tr>
<td>ME1</td>
<td>0</td>
<td>6.49</td>
</tr>
<tr>
<td>ME2</td>
<td>7.17</td>
<td>13.67</td>
</tr>
<tr>
<td>NADTRHD</td>
<td>0</td>
<td>6.49</td>
</tr>
<tr>
<td>PPCK</td>
<td>3.93</td>
<td>10.42</td>
</tr>
<tr>
<td>PYK</td>
<td>0</td>
<td>6.49</td>
</tr>
</tbody>
</table>

Variable Reactions For Growth On Succinate
(Same as Alternate Optimal Solutions Example)
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
## Flux Variability Analysis #2

**findingOptimalSolutionsSuccVisualize.m**

<table>
<thead>
<tr>
<th>Reaction</th>
<th>Upper Bound</th>
<th>Lower Bound</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACONTa</td>
<td>8.13764</td>
<td>FBA</td>
<td>-0.835681</td>
</tr>
<tr>
<td>ACON Tb</td>
<td>8.13764</td>
<td>FBP</td>
<td>0.835681</td>
</tr>
<tr>
<td>AKGDH</td>
<td>7.23122</td>
<td>FUM</td>
<td>27.2312</td>
</tr>
<tr>
<td>ATPM</td>
<td>8.39</td>
<td>GAPD</td>
<td>-2.2333</td>
</tr>
<tr>
<td>ATPS4r</td>
<td>57.7816</td>
<td>GLNS</td>
<td>0.214822</td>
</tr>
<tr>
<td>Biomass</td>
<td>0.840134</td>
<td>GLUDy</td>
<td>-4.36626</td>
</tr>
<tr>
<td>CO2t</td>
<td>-44.2477</td>
<td>H2Ot</td>
<td>-30.3675</td>
</tr>
<tr>
<td>CS</td>
<td>8.13764</td>
<td>ICDHyr</td>
<td>8.13764</td>
</tr>
<tr>
<td>CYTBD</td>
<td>66.5528</td>
<td>MDH</td>
<td>13.565</td>
</tr>
<tr>
<td>ENO</td>
<td>-3.49017</td>
<td>ME1</td>
<td>6.49242</td>
</tr>
<tr>
<td>EX_co2(e)</td>
<td>44.2477</td>
<td>ME2</td>
<td>7.1738</td>
</tr>
<tr>
<td>EX_h2o(e)</td>
<td>30.3675</td>
<td>NADH16</td>
<td>39.3216</td>
</tr>
<tr>
<td>EX_h(e)</td>
<td>-23.1469</td>
<td>NH4t</td>
<td>4.58108</td>
</tr>
<tr>
<td>EX_nh4(e)</td>
<td>-4.58108</td>
<td>O2t</td>
<td>33.2764</td>
</tr>
<tr>
<td>EX_o2(e)</td>
<td>-33.2764</td>
<td>PDH</td>
<td>11.2863</td>
</tr>
<tr>
<td>EX_pi(e)</td>
<td>-3.0906</td>
<td>PGI</td>
<td>-0.172228</td>
</tr>
<tr>
<td>EX_succ(e)</td>
<td>-20</td>
<td>PGK</td>
<td>2.23333</td>
</tr>
</tbody>
</table>

**FVA Upper Bound; Lower Bound.**

**Other FVA Reactions**

- NADTRHD: 0
- PYK: 0
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  SUCt2_x 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  Other FVA Reactions
EX_o2(e)  -33.2764  PG1  -0.172228  NADTRHD 0
EX_pi(e)  -3.0906  PGK  2.23333  ME1r  0
EX_succ(e) -20

FVA Upper Bound; Lower Bound.
Three Alternate Optimal Solutions

All three solution produce the same amount of NADH for the electron transport chain.
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
Knocking out reactions with a potential zero flux state can be used to reduce the number of alternate optimal solutions.

FVA Chart for Ethanol Production

FVA Ethanol Production.xlsx
Flux Variability Map

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

Examples:

- **PGK**: \(3\text{pg} + \text{atp} \rightleftharpoons 13\text{dpg} + \text{adp}\)
- **ENO**: \(2\text{pg} \rightleftharpoons \text{h}2\text{o} + \text{pep}\)
- **PPC**: \(\text{co}2 + \text{h}2\text{o} + \text{pep} \rightarrow \text{h} + \text{oa}a + \text{pi}\)
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.

% 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)
## FVA Classifications for Succinate Growth

<table>
<thead>
<tr>
<th>Hard-Coupled Reactions</th>
<th>Partially-Coupled Reactions</th>
<th>Not-Coupled Reactions</th>
<th>Zero-flux Reactions</th>
</tr>
</thead>
<tbody>
<tr>
<td>'Biomass_Ecoli_core_N(w/GAM)-Nmet2'</td>
<td>'ACONTa'</td>
<td>'ACALD'</td>
<td>'EX_fru(e)'</td>
</tr>
<tr>
<td>'ACONTb'</td>
<td>'ACALDt'</td>
<td>'EX_pyr(e)'</td>
<td>'EX_fum(e)'</td>
</tr>
<tr>
<td>'ATPS4r'</td>
<td>'ACKr'</td>
<td>'EX_succ(e)'</td>
<td>'EX_glc(e)'</td>
</tr>
<tr>
<td>'Biomass'</td>
<td>'Act2r'</td>
<td>'FBA'</td>
<td>'EX_gln_L(e)'</td>
</tr>
<tr>
<td>'CS'</td>
<td>'ADK1'</td>
<td>'FORt2'</td>
<td>'EX_mal_L(e)'</td>
</tr>
<tr>
<td>'CYTBD'</td>
<td>'AKGDH'</td>
<td>'FORt1'</td>
<td>'FRD7'</td>
</tr>
<tr>
<td>'EX_co2(e)'</td>
<td>'AKGt2r'</td>
<td>'G6PDH2r'</td>
<td>'FRUpts2'</td>
</tr>
<tr>
<td>'EX_h2o(e)'</td>
<td>'ALCD2x'</td>
<td>'PYK'</td>
<td>'FUMt2_2'</td>
</tr>
<tr>
<td>'FBP'</td>
<td>'CO2t'</td>
<td>'PYRt2r'</td>
<td>'GLCpts'</td>
</tr>
<tr>
<td>'FUM'</td>
<td>'D_LACt2'</td>
<td>'RPE'</td>
<td>'GLNabc'</td>
</tr>
<tr>
<td>'GLNS'</td>
<td>'ENO'</td>
<td>'RPI'</td>
<td>'MALt2_2'</td>
</tr>
<tr>
<td>'ICDHyr'</td>
<td>'ETOHt2r'</td>
<td>'SUCc13'</td>
<td></td>
</tr>
<tr>
<td>'MDH'</td>
<td>'EX_ac(e)'</td>
<td>'SUCOAS'</td>
<td></td>
</tr>
<tr>
<td>'NADH16'</td>
<td>'EX_acald(e)'</td>
<td>'TALA'</td>
<td></td>
</tr>
<tr>
<td>'NH4t'</td>
<td>'EX_akg(e)'</td>
<td>'THD2'</td>
<td></td>
</tr>
<tr>
<td>'O2t'</td>
<td>'EX_eth(e)'</td>
<td>'TKT1'</td>
<td></td>
</tr>
<tr>
<td>'PDH'</td>
<td>'EX_for(e)'</td>
<td>'TKT2'</td>
<td></td>
</tr>
<tr>
<td>'PGK'</td>
<td>'EX_gl_L(e)'</td>
<td>'TPI'</td>
<td></td>
</tr>
<tr>
<td>'PGM'</td>
<td>'EX_h(e)'</td>
<td></td>
<td></td>
</tr>
<tr>
<td>'PIt2r'</td>
<td>'EX_lac_D(e)'</td>
<td></td>
<td></td>
</tr>
<tr>
<td>'SUCc2_2'</td>
<td>'EX_lac_H(e)'</td>
<td></td>
<td></td>
</tr>
<tr>
<td>'SUCD1'</td>
<td>'EX_nh4(e)'</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>'EX_o2(e)'</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>'EX_pi(e)'</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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
  - \([\text{grRatio}, \text{grRateKO}, \text{grRateWT}, \text{hasEffect}, \text{delRxs}, \text{hasEffect}] = \text{singleRxnDeletion}(\text{model})\]
  - \([\text{grRatio}, \text{grRateKO}, \text{grRateWT}, \text{delRxs}, \text{hasEffect}] = \text{singleGeneDeletion}(\text{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, metabolic genes necessary for \textit{in silico} growth in the given media:

\begin{verbatim}
% 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)
\end{verbatim}

\begin{itemize}
  \item \textbf{Aerobic}
    \begin{itemize}
      \item 'ACONTa'
      \item 'ACONTb'
      \item Biomass
      \item 'CS'
      \item 'ENO'
      \item 'EX_glc(e)'
      \item 'EX_h(e)'
      \item 'EX_nh4(e)'
      \item 'EX_pi(e)'
      \item 'GAPD'
      \item 'GLCPts'
      \item 'GLNS'
      \item 'ICDHyr'
      \item 'NH4t'
      \item 'PFK'
      \item 'PGK'
      \item 'PGMT2r'
      \item 'PIt2r'
      \item 'PPC'
      \item 'RPI'
    \end{itemize}
  \item \textbf{Anaerobic}
    \begin{itemize}
      \item 'ACONTa'
      \item 'ACONTb'
      \item Biomass
      \item 'CS'
      \item 'ENO'
      \item 'EX_glc(e)'
      \item 'EX_h(e)'
      \item 'EX_nh4(e)'
      \item 'EX_pi(e)'
      \item 'GAPD'
      \item 'GLCPts'
      \item 'GLNS'
      \item 'ICDHyr'
      \item 'NH4t'
      \item 'PFK'
      \item 'PGK'
      \item 'PGM'
      \item 'PIt2r'
      \item 'PPP'
      \item 'RPI'
    \end{itemize}
\end{itemize}

\textbf{BioMass}

\textbf{RxnRatio} = Computed growth rate ratio between deletion strain and wild type
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);
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.

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.

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
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).

---

<table>
<thead>
<tr>
<th>Essential</th>
<th>pFBA Optima</th>
<th>Enzymatically Less Efficient</th>
<th>Metabolically Less Efficient</th>
<th>pFBA No-flux</th>
<th>Blocked</th>
</tr>
</thead>
<tbody>
<tr>
<td>'ACONTa'</td>
<td>'GLNS'</td>
<td>'ACALD'</td>
<td>'FRD7'</td>
<td>'ACALD1'</td>
<td>'FORT2'</td>
</tr>
<tr>
<td>'ACONTb'</td>
<td>'ICDHyr'</td>
<td>'ACKr'</td>
<td>'SUCDI'</td>
<td>'ADK1'</td>
<td>'FUM'</td>
</tr>
<tr>
<td>Biomass</td>
<td>'NH4t'</td>
<td>'AC12r'</td>
<td>'GLUDY'</td>
<td>'AKGDH'</td>
<td>'G6PDH2r'</td>
</tr>
<tr>
<td>'CS'</td>
<td>'PFK'</td>
<td>'ALCD2x'</td>
<td>'H2O2t'</td>
<td>'AKG12r'</td>
<td>'GLUN'</td>
</tr>
<tr>
<td>'ENO'</td>
<td>'PGI'</td>
<td>'ATPM'</td>
<td>'PEL'</td>
<td>'D_LAC12'</td>
<td>'GLUSy'</td>
</tr>
<tr>
<td>'EX_gl(e)'</td>
<td>'PGK'</td>
<td>'ATPS4r'</td>
<td>'PTAr'</td>
<td>'EX_acald(e)'</td>
<td>'GLU12r'</td>
</tr>
<tr>
<td>'EX_h(e)'</td>
<td>'PGM'</td>
<td>'CO2t'</td>
<td>'PYK'</td>
<td>'EX_ ald(e)'</td>
<td>'GND'</td>
</tr>
<tr>
<td>'EX_nh4(e)'</td>
<td>'Pit2r'</td>
<td>'ETOH2r'</td>
<td>'RPE'</td>
<td>'EX_glu_L(e)'</td>
<td>'ICL'</td>
</tr>
<tr>
<td>'EX_pi(e)'</td>
<td>'PPC'</td>
<td>'EX_ac(e)'</td>
<td>'TALa'</td>
<td>'EX_lac_D(e)'</td>
<td>'LDH_D'</td>
</tr>
<tr>
<td>'FBA'</td>
<td>'RPI'</td>
<td>'EX_co2(e)'</td>
<td>'THD2'</td>
<td>'EX_py(e)'</td>
<td>'MALS'</td>
</tr>
<tr>
<td>'GAPD'</td>
<td>'TPi'</td>
<td>'EX_etoh(e)'</td>
<td>'TKT1'</td>
<td>'EX_suc(e)'</td>
<td>'MDH'</td>
</tr>
<tr>
<td>'GLCpts'</td>
<td>'EX_for(e)'</td>
<td>'TKT2'</td>
<td></td>
<td>'FBP'</td>
<td>'ME1'</td>
</tr>
</tbody>
</table>

---

2019 H. Scott Hinton

Lesson: Flux Variability Analysis & Parsimonious Flux Balance Analysis
## Parsimonious FBA Data

<table>
<thead>
<tr>
<th>Essential</th>
<th>pFBA Optima</th>
<th>Enzymatically Less Efficient</th>
<th>Metabolically Less Efficient</th>
<th>pFBA No-flux</th>
<th>Blocked</th>
</tr>
</thead>
<tbody>
<tr>
<td>'ACONTa'</td>
<td>'ACALD'</td>
<td>'FRD7'</td>
<td>'ACALD1'</td>
<td>'ME2'</td>
<td>'CYTBD'</td>
</tr>
<tr>
<td>'ACONTb'</td>
<td>'ACKr'</td>
<td>'SUCDi'</td>
<td>'ADK1'</td>
<td>'NADH16'</td>
<td>'EX_fru(e)'</td>
</tr>
<tr>
<td>Biomass</td>
<td>'AC12r'</td>
<td>'AKGDH'</td>
<td>'NADTRHD'</td>
<td>'EX_fum(e)'</td>
<td></td>
</tr>
<tr>
<td>'CS'</td>
<td>'ALCD2x'</td>
<td>'AKG12r'</td>
<td>'PDH'</td>
<td>'EX_gln_L(e)'</td>
<td></td>
</tr>
<tr>
<td>'ENO'</td>
<td>'ATPM'</td>
<td>'D_LAc12'</td>
<td>'PGL'</td>
<td>'EX_mat_L(e)'</td>
<td></td>
</tr>
<tr>
<td>'EX_glc(e)'</td>
<td>'ATPS4r'</td>
<td>'EX_acald(e)'</td>
<td>'PPCK'</td>
<td>'EX_o2(e)'</td>
<td></td>
</tr>
<tr>
<td>'EX_h(e)'</td>
<td>'CO2t'</td>
<td>'EX_akg(e)'</td>
<td>'PPS'</td>
<td>'FRUpts2'</td>
<td></td>
</tr>
<tr>
<td>'EX_nh4(e)'</td>
<td>'ETOH12r'</td>
<td>'EX_glu_L(e)'</td>
<td>'PYRt2r'</td>
<td>'FUMt2_2'</td>
<td></td>
</tr>
<tr>
<td>'EX_pi(e)'</td>
<td>'EX_ac(e)'</td>
<td>'EX_lac_D(e)'</td>
<td>'SUCCI2_2'</td>
<td>'GLNabc'</td>
<td></td>
</tr>
<tr>
<td>'FBA'</td>
<td>'EX_co2(e)'</td>
<td>'EX_pyr(e)'</td>
<td>'SUCCl3'</td>
<td>'MALt2_2'</td>
<td></td>
</tr>
<tr>
<td>'GAPD'</td>
<td>'EX_etoh(e)'</td>
<td>'EX_succ(e)'</td>
<td>'SUCOAS'</td>
<td>'O2t'</td>
<td></td>
</tr>
<tr>
<td>'GLCpts'</td>
<td>'EX_for(e)'</td>
<td>'FBP'</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>'GLNS'</td>
<td>'EX_h2o(e)'</td>
<td>'FORt2'</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ICDHyr</td>
<td>'FORti'</td>
<td>'FUM'</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>'NH4t'</td>
<td>'GLUDy'</td>
<td>'G6PDH2r'</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>'PFK'</td>
<td>'H2O1'</td>
<td>'GLUN'</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>'PG1'</td>
<td>'PFL'</td>
<td>'GLUSy'</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>'PGK'</td>
<td>'PTAr'</td>
<td>'GLU12r'</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>'PGM'</td>
<td>'PYK'</td>
<td>'GND'</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>'PI12r'</td>
<td>'RPE'</td>
<td>'ICL'</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>'PPI'</td>
<td>'TALA'</td>
<td>'LDH_D'</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>'RPI'</td>
<td>'THD2'</td>
<td>'MALs'</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>'TPI'</td>
<td>'TKT1'</td>
<td>'MDH'</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>'TKT2'</td>
<td></td>
<td>'ME1'</td>
<td></td>
</tr>
</tbody>
</table>

---

**Diagram: FBA_Ecoli_Core.m (Anaerobic)**

- **PPP**
- **Glyc**
- **OxP**
- **TCA**
- **Ana**
- **Ferm**

---

**Utah State University**
**BIE 5500/6500**
Lesson: Flux Variability Analysis & Parsimonious Flux Balance Analysis
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.

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);

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

% 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)