Randomized Sampling and Adaptive Laboratory Evolution
LEARNING OBJECTIVES

Each student should be able to:

• Explain randomized sampling

• Explain the Method of Minimization of Metabolic Adjustment (MOMA)

• Explain adaptive laboratory evolution

• Explain extreme pathways
Lesson Outline

• Randomized Sampling

• Randomized Sampling Examples

• Method of Minimization of Metabolic Adjustment (MOMA)

• Adaptive Laboratory Evolution

• Extreme Pathways
Randomized Sampling

• An alternative approach to characterizing the contents of a networks solution space is uniform random sampling.

• This approach involves obtaining a statistically meaningful number of solutions that have been uniformly distributed through the entire solution space.

• Randomized sampling of candidate network states throughout an entire solution space gives an unbiased assessment of its properties.

• The process of obtaining a uniform set of candidate solutions includes:
  1. Defining the space to be sampled based on the imposed constraints
  2. Randomly sampling it based on uniform statistical criteria
  3. Further segmenting the solution space based on additional post-sampling criteria as necessary.

Solution Space

Flux Balance Analysis

Identifies one solution from all solution space

Randomized Sampling

Multiple solutions representing all solution space

The solution space is typically a polytope in n-dimensional solution space

The "solution space," contains the set of all feasible solutions that satisfy the imposed constraints

Constraint-based Metabolic Reconstructions & Analysis

Lesson: Randomized Sampling & Adaptive Laboratory Evolution
Sampling of the Steady State Solution Space

Monte Carlo Sampling is used to generate a set of uniform flux distributions. The method is based on the Artificially Centered Hit and Run (ACHR) algorithm with slight modifications. Initially a set of non-uniform pseudo-random points, called warm-up points, is generated. In a series of iterations, each point is randomly moved, always remaining within the feasible flux space.

This procedure is achieved by 1) choosing a random direction, 2) computing the limits of how far one can travel in that direction in either positive or negative direction and 3) choosing a new point randomly along this line. After many iterations, the set of points is mixed and approach a uniform sample of the solution space.

Jan Schellenberger, PhD Dissertation, University of California, San Diego, 2010
Hit-and-Run Sampling

• The “hit-and-run” method: Characterize the solution space, that is all the possible flux states of the system using only the constraints imposed by mass conservation and stoichiometry.

• Starting from a random initial point (red) inside the positive flux cone in a randomly chosen direction, the bouncer travels deterministically a distance $d$ between sample points. Each sample point (green), corresponds to a solution vector where the components are the individual fluxes. After every $b^{th}$ bounce off the internal walls of the flux cone, the direction of the bouncer is randomized.

• Implemented in the Cobra Toolbox with createHRWarmup.m which creates the initial point set point for hit-and-run sampling and ACHRSampler.m which performs the hit-and-run sampling.

Mixed Fraction Parameter

- There is no guarantee that these sample points will uniformly cover the entire solution space.
- The mixed fraction (mf) measures the fraction of points that have crossed the median in any direction.
- If the points have been perfectly mixed and there is no dependence between the initial position and final position, then the chance of crossing the partition is exactly 50%.
- Therefore taking an average of all points would result in a mixed fraction of 0.5 when mixing is achieved.
- Initially, before points have moved at all, the mixed fraction is exactly 1 and the mixed fraction would be expected to decrease exponentially towards 0.5.

Several points are moved throughout the space of interest in parallel (a). The mixed fraction is computed as follows: Axes are drawn along all principle directions (b) and a count is tabulated of which points cross which boundaries (c). A ‘0’ indicates that a point crossed a certain axis and a ‘1’ indicates the point is still on the same side.

Jan Schellenberger, PhD Dissertation, University of California, San Diego, 2010
gpSampler: Cobra Toolbox Function

gpSampler Samples an arbitrary linearly constrained space using a fixed number of points that are moved in parallel

\[
\text{[sampleStructOut, mixedFraction]} = \text{gpSampler(sampleStruct, nPoints, bias, maxTime, maxSteps)}
\]

**INPUTS**

sampleStruct Structure describing the space to be sampled and previous point sets

**OPTIONAL INPUTS**

nPoints Number of points used in sampling (default = 2*nRxns or 5000 whichever is greater)

bias In most cases is empty, []

method Biasing distribution: 'uniform', 'normal'

index The reaction indexes which to bias (nBias total)

param nBias x 2 matrix of parameters (for uniform it’s min, max, for normal it’s mu, sigma).

maxTime Maximum time allotted for the sampling in seconds (default 600 s, pass an empty number [] to set maxSteps instead)

maxSteps Maximum number of steps to take (default 1e10). Sampler will run until either maxStep or maxTime is reached.

**OUTPUT**

sampleStructOut The sampling structure with some extra fields.

mixedFract The fraction mixed. A value of 1 means not mixed at all, a value of .5 means completely mixed.
Histograms of Flux Samples

% Sampling_Histogram.m

clear;

% Input the E.coli core model and set constraints
load('ecoli_textbook.mat');
model = changeRxnBounds(model,'EX_glc(e)',-10,'l');
model = changeRxnBounds(model,'EX_o2(e)',-20,'l');
model = changeObjective(model,'Biomass_Ecoli_core_N(w/GAM)_Nmet2');

% Sample model
[sampleStruct,mixedFrac] = gpSampler(model,5000,[],120);

% Determine the minimum and maximum possible fluxes so the sampling results can be plotted for the reactions in the model
[minFlux,maxFlux] = fluxVariability(model,0);
for i = 1 : 95
    subplot(8,12,i)
    hist(sampleStruct.points(i,:),50);
    hold on
    plot([minFlux(i) maxFlux(i)], [0 1],'*r');
    title(model.rxns{i});
end
sampleStruct in Matlab Desktop

Sampling_Histogram.m
% Sampling_Succ_Histogram.m

clear;
load('ecoli_textbook.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');
FBAsolution = optimizeCbModel(model,'max',0,0);

% Sample model
[sampleStruct,mixedFrac] = gpSampler(model,5000,[],120);

% Plot histograms for selected reactions
rxnList = {'FORt2', 'FORti', 'MDH', 'ME1', 'ME2', 'NADTRHD', 'PPCK', 'PYK', 'EX_succ(e)', 'Biomass_Ecoli_core_N(w/GAM)_Nmet2'};

% Include optimal flux values on histograms
figure(1);
for i = 1 : 10
  subplot(2,5,i)
  rxnID = findRxnIDs(model,rxnList(i))
  hist(sampleStruct.points(rxnID,:),50);
  hold on
  plot(FBAsolution.x(rxnID), [0 1], '*.r');
  title(rxnList(i));
end

% Include flux variability analysis mins and maxs
figure(2);
[minFlux,maxFlux]=fluxVariability(model,100,'max',model.rxns,false,0);
for i = 1 : 10
  subplot(2,5,i)
  rxnID = findRxnIDs(model,rxnList(i))
  hist(sampleStruct.points(rxnID,:),50);
  hold on
  plot([minFlux(rxnID) maxFlux(rxnID)], [0 1], '*.r');
  title(rxnList(i));
end
Constraint-based Metabolic Reconstructions & Analysis

Optimal Flux Values Included (*)

Flux Variability Analysis Min and Max Values Included (*)

Sampling_Succ_Histogram.m
Why are the Fluxes for FORti and FORt2 so large?
Comparing Sampling Results to Optimized Results
(Succinate Example from FVA)

```matlab
% Sampling_Succ_Optimal.m

clear;
load('ecoli_textbook.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');

% Find optimal value objective function (growth rate)
FBAsolution = optimizeCbModel(model,'max',0,0); % Force output to be close to optimal value
printFluxVector(model, FBAsolution.x, true);

% Sample model
[sampleStruct,mixedFrac] = gpSampler(model,5000,[],120);

% Plot histograms for selected reactions
rxnList= {'FORt2', 'FORti', 'MDH', 'ME1', 'ME2', 'NADTRHD', 'PPCK', 'PYK', 'EX_succ(e)', 'Biomass_Ecoli_core_N(w/GAM)_Nmet2'};

% Include flux variability analysis mins and maxs on the histogram
[minFlux,maxFlux]=fluxVariability(model,100,'max',model.rxs,false);
for i = 1 : 10
    subplot(2,5,i)
    rxnID= findRxnIDs(model,rxnList(i));
    hist(sampleStruct.points(rxnID,:),50);
    hold on
    plot([minFlux(rxnID) maxFlux(rxnID)], [0 1],'*r');
    title(rxnList(i));
end
```
Comparing Sampling Results to Optimized Results

Legend:
- **FORt2**
- **FORti**
- **MDH**
- **ME1**
- **ME2**
- **NADTRHD**
- **PPCK**
- **PYK**
- **EX_{ucc}(e)**
- **Biomass_{L\,coli\,ore\,(w/GAM)\,met2}**
Comparing Sampling Results to Optimized Results

Allow biomass function to be ≥ 90% maximum growth rate

Note: FVA is based on the optimal 100% of the objective function value

Sampling_Succ_Optimal.m
**plotSampleHist**

plotSampleHist Compare flux histograms for one or more samples for one or more reactions

\[
\text{plotSampleHist}(\text{rxnNames}, \text{samples}, \text{models}, \text{nBins}, \text{perScreen})
\]

**INPUTS**
- `rxnNames` Cell array of reaction abbreviations
- `samples` Cell array containing samples
- `models` Cell array containing model structures or common model structure

**OPTIONAL INPUTS**
- `nBins` Number of bins to be used (Default = round(nSamples/25))
- `perScreen` Number of reactions to show per screen. Either a number or [nY, nX] vector.
  
  (press Enter to advance screens)

**CONTROLS**
- To advance to next screen hit enter/return or type f and hit enter/return
- To rewind to previous screen type r or b and hit enter/return
- To quit script type q and hit enter/return

http://opencobra.github.io/cobratoolbox/docs/index.html
Comparing `plotSampleHist` with Hist plots
gpSampler: Comparing Phenotypes

% Sampling_Example_gpSampler.m

clear;

% Input the E.coli core model and set constraints
load('ecoli_textbook.mat');
model_aerobic = model;

% Sampling aerobic model
sampleStruct_aerobic = gpSampler(model_aerobic,2000,[],120);

% Sampling anaerobic model
model_anaerobic = changeRxnBounds(model_aerobic,'EX_o2(e)',0,'1');
sampleStruct_anaerobic = gpSampler(model_anaerobic,2000,[],120);

% Sampling results will be returned in the two structures
% sampleStruct_aerobic and sampleStruct_anaerobic within the field points.

% Visualize sampling results for a set of reactions.
rxnList = {'PGI', 'PFK', 'FBP', 'FBA', 'TPI', 'GAPD', 'PGK', 'PGM', 'ENO', 'PYK'};
plotSampleHist(rxnList, {sampleStruct_aerobic.points, sampleStruct_anaerobic.points }, {model_aerobic, …


The sampled data is only as good as the constraints that define the network.
Scatter Matrix

% Sampling_ScatterMatrix.m

clear;

% Input the E.coli core model and set constraints
load('ecoli_textbook.mat');
model = changeRxnBounds(model,'EX_glc(e)',-10,'l');
model = changeRxnBounds(model,'EX_o2(e)',-20,'l');

% Sample model
sampleStruct = gpSampler(model,5000,[],120);

% Plot scatter matrix
rxnList = {'PGI', 'PFK', 'FBP', 'FBA', 'TPI', 'GAPD', 'PGK', 'PGM', 'ENO', 'PYK'};
sampleScatterMatrix(rxnList,model,sampleStruct.points,250);

A scatter plot allows us to visualize the interaction between two network reactions.
This scatter plot allows us to visualize the interaction between two network reactions. For example, the reactions PGK and PGM are perfectly correlated in the test condition (sample points between the two reactions align on a line). In contrast, the flux through PGK is independent from the flux through the reaction FBP.
Correlated Reaction Sets

Two reactions are part of the same "correlated reaction set" if their fluxes are linearly correlated.

% IdentifyingCorrelSets.m
clear;

% Input the E.coli core model and set constraints
load('ecoli_textbook.mat');
model = changeRxnBounds(model,'EX_glc(e)',-10,'l');
model = changeRxnBounds(model,'EX_o2(e)',-20,'l');
model = changeObjective(model,'Biomass_Ecoli_core_N(w/GAM)_Nmet2');

[sampleStruct,mixedFrac] = gpSampler(model,5000,[],120);
[setsSorted,setNoSorted,setSize] = identifyCorrelSets(model,sampleStruct.points);
setNames = [];
setNumbers = [];
disp('Correlated Reactions Sets')
for i = 1 : length(setsSorted)
    setNames = [setNames; setsSorted(i).names];
    setNumbers = [setNumbers;i*ones(length(setsSorted(i).names),1)];
    disp([i,setsSorted(i).names'])
end

<table>
<thead>
<tr>
<th>Set#</th>
<th>Reactions</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1]</td>
<td>'ACKr'</td>
</tr>
<tr>
<td>[2]</td>
<td>'G6PDH2r'</td>
</tr>
<tr>
<td>[3]</td>
<td>'D_LACT2'</td>
</tr>
<tr>
<td>[4]</td>
<td>'CYTBD'</td>
</tr>
<tr>
<td>[5]</td>
<td>'Biomass'</td>
</tr>
<tr>
<td>[6]</td>
<td>'ALCD2x'</td>
</tr>
<tr>
<td>[7]</td>
<td>'ACONTa'</td>
</tr>
<tr>
<td>[8]</td>
<td>'TALA'</td>
</tr>
<tr>
<td>[9]</td>
<td>'ICL'</td>
</tr>
<tr>
<td>[10]</td>
<td>'GAPD'</td>
</tr>
<tr>
<td>[11]</td>
<td>'FBA'</td>
</tr>
<tr>
<td>[12]</td>
<td>'EX_pyr(e)'</td>
</tr>
<tr>
<td>[13]</td>
<td>'EX_nh4(e)'</td>
</tr>
<tr>
<td>[14]</td>
<td>'EX_h2o(e)'</td>
</tr>
<tr>
<td>[15]</td>
<td>'EX_glu_L(e)'</td>
</tr>
<tr>
<td>[16]</td>
<td>'EX_glc(e)'</td>
</tr>
<tr>
<td>[17]</td>
<td>'EX_for(e)'</td>
</tr>
<tr>
<td>[18]</td>
<td>'ENO'</td>
</tr>
<tr>
<td>[19]</td>
<td>'CO2t'</td>
</tr>
<tr>
<td>[20]</td>
<td>'AKGt2r'</td>
</tr>
<tr>
<td>[21]</td>
<td>'AKGDH'</td>
</tr>
<tr>
<td>[22]</td>
<td>'ADK1'</td>
</tr>
<tr>
<td>[23]</td>
<td>'ACALDt'</td>
</tr>
</tbody>
</table>
### Correlated Reaction Sets Scatter Matrix

When creating a minimal set of candidate reactions for optKnock, GDLS, or optGene, only one reaction from each reaction set needs to be included since the deletion of any reaction from each set would lead to the same result.

![Correlated Reaction Sets Scatter Matrix](CorrelSets_ScatterMatrix.m)
### Mapped Correlated Reaction Sets

<table>
<thead>
<tr>
<th>Set#</th>
<th>Reactions</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1]</td>
<td>'ACKr' 'ACT2r' 'EX_ac(e)' 'PTAr'</td>
</tr>
<tr>
<td>[2]</td>
<td>'G6PD2H2r' 'GND' 'PGL'</td>
</tr>
<tr>
<td>[3]</td>
<td>'D_LACT2' 'EX_lac_(D)(e)' 'LDH_D'</td>
</tr>
<tr>
<td>[4]</td>
<td>'CYTBD' 'EX_o2(e)' 'O2t'</td>
</tr>
<tr>
<td>[5]</td>
<td>'Biomass' 'EX_pi(e)' 'PTIt2r'</td>
</tr>
<tr>
<td>[6]</td>
<td>'ALCD2x' 'ETOHT2r' 'EX_etoh(e)'</td>
</tr>
<tr>
<td>[7]</td>
<td>'ACONTa' 'ACONTb' 'CS'</td>
</tr>
<tr>
<td>[8]</td>
<td>'TALA' 'TKT1'</td>
</tr>
<tr>
<td>[9]</td>
<td>'ICL' 'MALS'</td>
</tr>
<tr>
<td>[10]</td>
<td>'GAPD' 'PGK'</td>
</tr>
<tr>
<td>[11]</td>
<td>'FBA' 'TPI'</td>
</tr>
<tr>
<td>[12]</td>
<td>'EX_pyr(e)' 'PYRt2r'</td>
</tr>
<tr>
<td>[13]</td>
<td>'EX_nh4(e)' 'NH4t'</td>
</tr>
<tr>
<td>[14]</td>
<td>'EX_h2o(e)' 'H2Ot'</td>
</tr>
<tr>
<td>[15]</td>
<td>'EX_glu_L(e)' 'GLUt2r'</td>
</tr>
<tr>
<td>[16]</td>
<td>'EX_glc(e)' 'GLCpts'</td>
</tr>
<tr>
<td>[17]</td>
<td>'EX_for(e)' 'PFL'</td>
</tr>
<tr>
<td>[18]</td>
<td>'ENO' 'PGM'</td>
</tr>
<tr>
<td>[19]</td>
<td>'CO2t' 'EX_co2(e)'</td>
</tr>
<tr>
<td>[20]</td>
<td>'AKGt2r' 'EX_akg(e)'</td>
</tr>
<tr>
<td>[21]</td>
<td>'AKGDH' 'SUCOAS'</td>
</tr>
<tr>
<td>[22]</td>
<td>'ADK1' 'PPS'</td>
</tr>
<tr>
<td>[23]</td>
<td>'ACALDt' 'EX_acald(e)'</td>
</tr>
</tbody>
</table>

**IdentifyingCorrelSets.m**

- `EX_glc(e)=-10`
- `EX_o2(e)=-20`
Review Questions

- What are correlated reaction sets?
- What is the solution space?
- What determines the solution space that is used in randomized sampling?
- Is randomized sampling classified as biased or unbiased assessment?
- What is hit-and-run sampling?
- What is the mixed fraction parameter?
- Under what name are the sample points listed in the sampleStruct?
- What role does the objective function play in randomized sampling?
- What Cobra function allows the graphical comparison of different sampled solutions?
- What Cobra function provides a graphical representation of the correlations between reactions?
- What Cobra function can be used for randomized sampling?
- What role do reaction constraints play in the accuracy of the data generated by randomized sampling?
Lesson Outline

• Randomized Sampling

• Randomized Sampling Examples
  • Method of Minimization of Metabolic Adjustment (MOMA)
  • Adaptive Laboratory Evolution
  • Extreme Pathways
Ethanol Production

% EthanolProduction_Sampling.m
clear;
% Input the E.coli core model
load('ecoli_textbook.mat');
% Set carbon source and oxygen uptake rates for wild type model
model = changeRxnBounds(model,'EX_glc(e)',-5,'l');
model = changeRxnBounds(model,'EX_o2(e)',-20,'l');
model = changeObjective(model,'Biomass_Ecoli_core_N(w/GAM)_Nmet2');
FBAsolution = optimizeCbModel(model,'max',0,0);
model_WT = model;

% Knockout reactions for mutant model
model = changeRxnBounds(model,'NADH16',0,'b');
model = changeRxnBounds(model,'PTAr',0,'b');
model = changeRxnBounds(model,'TKT2',0,'b');
Mutantsolution = optimizeCbModel(model,'max',0,0);
model_Mutant = model;

sampleStruct_WT = gpSampler(model_WT,2000,[],120);
% Simulation time is ~120 s.
sampleStruct_Mutant = gpSampler(model_Mutant,2000,[],120);
% Simulation time is ~120 s. Sampling results will be returned in the
% two structures sampleStruct_WT and sampleStruct_Mutant within the
% field points.
% Visualize sampling results for a set of reactions.
rxnList = {'EX_glc(e)', 'EX_o2(e)', 'EX_etoh(e)', 'EX_lac_D(e)',
            'EX_for(e)', 'ATPS4r', 'CYTBD', 'GND', 'ICDHyr',
            'Biomass_Ecoli_core_N(w/GAM)_Nmet2'};
plotSampleHist(rxnList, {sampleStruct_WT.points,
                        sampleStruct_Mutant.points }, {model_WT, model_Mutant},[],[2,5]);

EthanolProduction_Sampling.m
Ethanol Production Comparative Example

- Note that the solution space has been reduced.
- Glucose flux distribution is narrower and closer to the maximum uptake.
- Oxygen flux distribution is narrower and closer to zero.
- Ethanol flux distribution is narrower and centered at a higher secretion rate.
- Formate flux distribution is narrower and centered at a higher secretion rate.
- ATP54r flux distribution (ATP production by oxidative phosphorylation) is narrower and closer to zero.
- CYTDB flux distribution (electron transport chain for oxidative phosphorylation) is narrower and closer to zero.
- The biomass production is narrower and closer to zero.

Blue = Wild Type; Red = Mutant (Knockouts = {NADH16, PTAr, TKT2})
Probability Flux Distributions for Human Red Blood Cells

- The red blood cell model with imposed maximum and minimum constraints on each flux was sampled using the *in silico* algorithm.

- The histograms next to each reaction represent the distribution of solutions with respect to each reaction flux. The vertical shaded line on each plot indicates where the zero flux line is.

- Due to the convexity of the solution space, no distribution can have more than one peak.

- The flux distribution shape gives information about the sensitivity of the solution space to each constraint.

- If a flux distribution has a right peak, decreasing a maximum constraint will eliminate many solutions from the valid space.

- Reactions that are part of the same pathway with no intermediate branch points (PGM, EM, PK) all have the same flux distributions.

- Distributions shown are based on 500,000 uniformly distributed points in the steady-state flux space.

Metabolic Network States Under Normal And Disease Conditions

By applying constraints (Vmin, Vmax) to reaction, uptake, and secretion rates based on experimental data, the range of allowable steady states of the metabolic network consistent with these experimental data can be generated.

Lesson Outline

• Randomized Sampling

• Randomized Sampling Examples

• Method of Minimization of Metabolic Adjustment (MOMA)

• Adaptive Laboratory Evolution

• Extreme Pathways
Do Cells Really Operate at the Calculated Optimal State Of Bioproduct Production?

• It has been assumed that the mutant bacteria display an optimal metabolic state.

• Unfortunately, mutants generated artificially in the laboratory are generally not subjected to the same evolutionary pressure that shaped the wild type. Thus, a mutant is likely to initially display a suboptimal flux distribution that is somehow intermediate between the wild-type optimum and the mutant optimum.

• The method of minimization of metabolic adjustment (MOMA) has been developed, which is based on the same stoichiometric constraints as FBA, but relaxes the assumption of optimal growth flux for gene/reaction deletions.

• MOMA provides a mathematically tractable approximation for this intermediate suboptimal state, based on the conjecture that the mutant remains initially as close as possible.

**Method of Minimization of Metabolic Adjustment (MOMA)**


- Uses the same steady state flux cone as FBA.
- Relaxes the assumption of maximal optimal growth.
- MOMA searches the flux distribution in the “mutant flux space” which is closest to the optimal flux distribution in the “wild-type flux space.”
- Typically returns suboptimal flux distribution between wild type optimum and mutant optimum.

% EthanolProduction_GDLS_Mutants_MOMA.m

clear;

% Load the E.coli core model
load('ecoli_textbook.mat');

% Set carbon source and oxygen uptake rates for wild type model
model = changeRxnBounds(model,'EX_glc(e)',-5,'l');
model = changeRxnBounds(model,'EX_o2(e)',-20,'l'); % Aerobic ethanol production
model = changeObjective(model,'Biomass_Ecoli_core_N(w/GAM)_Nmet2');
FBAsolution = optimizeCbModel(model,'max',0,0);
modelWT = model;

% Knockout reactions for the mutant model
model = changeRxnBounds(model,'NADH16',0,'b');
model = changeRxnBounds(model,'PTAr',0,'b');
model = changeRxnBounds(model,'TKT2',0,'b');
Mutantsolution = optimizeCbModel(model,'max',0,0);
modelMutant = model;

% MOMA calculation
[solutionDel,solutionWT,totalFluxDiff,solStatus] = MOMA(modelWT,modelMutant,'max',false)
printFluxVector(model, [FBAsolution.x,Mutantsolution.x solutionDel.x], true)
# Flux Differences: Wild Type, GDLS, & MOMA

<table>
<thead>
<tr>
<th>Reaction</th>
<th>WT</th>
<th>GDSL</th>
<th>MOMA</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACALD</td>
<td>2.27E-13</td>
<td>-7.38652</td>
<td>-5.38586</td>
</tr>
<tr>
<td>ACONTa</td>
<td>3.44346</td>
<td>1.87357</td>
<td>0.010514</td>
</tr>
<tr>
<td>ACON Tb</td>
<td>3.44346</td>
<td>1.87357</td>
<td>0.010514</td>
</tr>
<tr>
<td>AKGDH</td>
<td>2.99507</td>
<td>1.81911</td>
<td>0</td>
</tr>
<tr>
<td>ALCD2x</td>
<td>0</td>
<td>-7.38652</td>
<td>-5.38586</td>
</tr>
<tr>
<td>ATPM</td>
<td>8.39</td>
<td>8.39</td>
<td>8.39</td>
</tr>
<tr>
<td>ATPS4r</td>
<td>24.8712</td>
<td>0.094332</td>
<td>1.27156</td>
</tr>
<tr>
<td>Biomass</td>
<td>0.415598</td>
<td>0.050473</td>
<td>0.009745</td>
</tr>
<tr>
<td>CO2t</td>
<td>-12.314</td>
<td>-3.6298</td>
<td>-5.36298</td>
</tr>
<tr>
<td>CS</td>
<td>3.44346</td>
<td>1.87357</td>
<td>0.010514</td>
</tr>
<tr>
<td>CYTBD</td>
<td>23.6671</td>
<td>1.81911</td>
<td>0</td>
</tr>
<tr>
<td>D_LACt2</td>
<td>0</td>
<td>0</td>
<td>-2.39504</td>
</tr>
<tr>
<td>ENO</td>
<td>7.58501</td>
<td>9.76307</td>
<td>7.88854</td>
</tr>
<tr>
<td>ETOHtr</td>
<td>0</td>
<td>-7.38652</td>
<td>-5.38586</td>
</tr>
<tr>
<td>EX_co2(e)</td>
<td>12.314</td>
<td>3.6298</td>
<td>5.36298</td>
</tr>
<tr>
<td>EX_etho(e)</td>
<td>0</td>
<td>7.38652</td>
<td>5.38586</td>
</tr>
<tr>
<td>EX_for(e)</td>
<td>0</td>
<td>9.44925</td>
<td>0.068288</td>
</tr>
<tr>
<td>EX_glc(e)</td>
<td>-5</td>
<td>-5</td>
<td>-3.96714</td>
</tr>
<tr>
<td>EX_h2o(e)</td>
<td>15.3414</td>
<td>-3.38905</td>
<td>0.048112</td>
</tr>
<tr>
<td>EX_h(e)</td>
<td>8.33689</td>
<td>10.4617</td>
<td>2.65882</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Reaction</th>
<th>WT</th>
<th>GDSL</th>
<th>MOMA</th>
</tr>
</thead>
<tbody>
<tr>
<td>EX_lac_D(e)</td>
<td>0</td>
<td>0</td>
<td>2.39504</td>
</tr>
<tr>
<td>EX_nh4(e)</td>
<td>-2.26617</td>
<td>-0.27522</td>
<td>-0.05314</td>
</tr>
<tr>
<td>EX_o2(e)</td>
<td>-11.8336</td>
<td>-0.90956</td>
<td>0</td>
</tr>
<tr>
<td>EX_pi(e)</td>
<td>-1.52886</td>
<td>-0.18568</td>
<td>-0.03585</td>
</tr>
<tr>
<td>FBA</td>
<td>3.89814</td>
<td>4.92254</td>
<td>3.95219</td>
</tr>
<tr>
<td>FORti</td>
<td>0</td>
<td>9.44925</td>
<td>0.068288</td>
</tr>
<tr>
<td>FRD7</td>
<td>0</td>
<td>0</td>
<td>1.49752</td>
</tr>
<tr>
<td>FUM</td>
<td>2.99507</td>
<td>1.81911</td>
<td>0</td>
</tr>
<tr>
<td>G6PDH2r</td>
<td>2.06541</td>
<td>0.081752</td>
<td>0.015785</td>
</tr>
<tr>
<td>GAPD</td>
<td>8.20675</td>
<td>9.83858</td>
<td>7.90311</td>
</tr>
<tr>
<td>GLCpts</td>
<td>5</td>
<td>5</td>
<td>3.96714</td>
</tr>
<tr>
<td>GLNS</td>
<td>0.106268</td>
<td>0.012906</td>
<td>0.002492</td>
</tr>
<tr>
<td>GLUDy</td>
<td>-2.1599</td>
<td>-0.26232</td>
<td>-0.05065</td>
</tr>
<tr>
<td>GND</td>
<td>2.06541</td>
<td>0.081752</td>
<td>0.015785</td>
</tr>
<tr>
<td>H2Ot</td>
<td>-15.3414</td>
<td>3.38905</td>
<td>-0.04811</td>
</tr>
<tr>
<td>ICDHyr</td>
<td>3.44346</td>
<td>1.87357</td>
<td>0.010514</td>
</tr>
<tr>
<td>LDH_D</td>
<td>0</td>
<td>0</td>
<td>-2.39504</td>
</tr>
<tr>
<td>MDH</td>
<td>2.99507</td>
<td>1.81911</td>
<td>-0.13553</td>
</tr>
<tr>
<td>ME2</td>
<td>0</td>
<td>0</td>
<td>0.135527</td>
</tr>
<tr>
<td>NADH16</td>
<td>20.672</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

The file `EthanolProduction_GDLS_Mutants_MOMA.m` contains the data for the ethanol production model.
Flux Maps: Wild Type, GDLS, & MOMA

Wild Type

GDLS-Mutant

MOMA Result

EthanolProduction_WT.m

EthanolProduction_GDLS_Mutant.m

EthanolProduction_GDLS_Mutants_MOMA.m
Production Envelope - Ethanol Production Mutant

Solution Space

FBA Operating Point
(0.050473, 7.3865)

MOMA Operating Point
(0.009745, 5.38586)

Wild Type Production Envelope

Mutant Production Envelope

EthanolProduction_WT.m

EthanolProduction_ProductionEnvelope_GDLS_Mutants.m
Review Questions

1. After a gene has been knockout or a new gene has been added to a host cell does the maximum theoretical performance typically match the laboratory results?

2. What Cobra function can be used to approximate the intermediate suboptimal state of the modified host cell?

3. What is a wild-type cell/model? How does it differ from the mutant cell/model?

4. Are the optimized flux values similar to the MOMA flux results?
Lesson Outline

• Randomized Sampling

• Randomized Sampling Examples

• Method of Minimization of Metabolic Adjustment (MOMA)

→ • Adaptive Laboratory Evolution

• Extreme Pathways
Line of Optimality

- The line of optimality (LO) is defined as a line representing the optimal relation between the two metabolic fluxes used to create a phenotype phase plane.

- The line of optimality is determined by specifying an uptake rate of the substrate along the x-axis and then allowing any value for the flux along the y-axis. Linear Programming can then be used to calculate the optimal value of the objective as a function of the y-axis flux. Once the objective is determined, the corresponding flux value for the y-axis is used to plot the line of optimality (LO).

- The LO defines the optimal utilization of the metabolic pathways without limitations on the availability of the substrates.
Desired Cell Evolution

![Graph showing Desired Cell Evolution](image)

- (0.009745, 5.38586)
- (0.050473, 7.3865)
Adaptive Laboratory Evolution

Adaptive Laboratory Evolution Methods

Adaptive Laboratory Evolution

- Cells will modify their genotype to optimize their fitness in the given environment.
- FBA results can be presented on phenotypic phase-plane (PPP) plots of model-predicted optimal biomass production (growth rate) versus carbon source uptake rate (SUR) and oxygen uptake rate (OUR).
- The line of optimality (LO) describes the most efficient ratio of SUR and OUR for biomass synthesis.
- Under several conditions, the experimentally measured *E.coli* phenotype corresponds to the LO of the PPP.
- When *E.coli* growth is not consistent with the LO, populations migrate toward the LO through adaptive evolution.
- Adaptive evolution outcomes have shown that evolved strains exhibit a general pattern of increased expression of genes and proteins associated with the optimal flux distribution, and decreased expression of genes and proteins associated with unused pathways.
- The frequent mutation of transcriptional regulators is consistent with recent evidence showing that regulatory networks evolve faster than other networks, such as genetic networks, protein interaction networks, and metabolic networks.

Adaptive Laboratory Evolution

**Growth rate of E. coli K-12 on Glucose, Malate, Succinate, & Acetate**

- Growth rate (exponential phase) during adaptive evolution on glucose, malate, succinate and acetate.

- The increases in growth rate over time were as follows:
  - glucose (18%),
  - malate (21%),
  - succinate (17%) and
  - acetate (20%).

- The number of generations for each adaptive evolution was: glucose (500), malate (500), succinate (1,000) and acetate (700).

Growth of *E. coli* K-12 on Malate

Blue dots are starting and ending points

Growth of *E. coli* K-12 on Glucose

Growth of *E. coli* K-12 on Glycerol

**Graphs:**

- **a:** Growth rate (h⁻¹) vs. Time (d) for Days 0–40, showing curves for E1, E2, E3.
  - **37°C**
  - **30°C**
  - E1, E2 evolving
  - E1, E2 stable

- **b:** Day 0 graph with growth rate vs. GI-UR, showing points for different conditions.
- **c:** Day 1–40 graph, similar to b with evolving and stable conditions.
- **d:** Day 40 graph, similar to b.
- **e:** Day 60 graph, similar to b.

*(Ibarra, 2002)*
Review Questions

1. Why don’t the first generation of transformed cells normally achieve the optimal performance predicted in the FBA models?

2. How can cells evolve after 100’s of generations to operate on the line of optimality?

3. What is the relationship between MOMA and adaptive laboratory evolution?

4. What are the number of generations required for adaptive laboratory evolution?

Lesson Outline

• Randomized Sampling

• Randomized Sampling Examples

• Method of Minimization of Metabolic Adjustment (MOMA)

• Adaptive Laboratory Evolution

• Extreme Pathways
Genome annotations, biochemical experiments and cell physiology data provide information to describe all reactions within a system.

- A reaction network is created from diverse data sets by defining all the different reactions in an organism.
- Data is used to create a stoichiometric matrix that relates all of the reactions within a network to all of the participating metabolites in a given organism.
- Extreme pathways are a unique set of convex basis vectors that correspond to the edges of a polytope and to flux pathways.
- A linear combination of non-negative convex basis vectors can represent all possible flux distributions that lie within the 'cone' circumscribed by the extreme pathways.

Lesson Outline

• Randomized Sampling

• Randomized Sampling Examples

• Method of Minimization of Metabolic Adjustment (MOMA)

• Adaptive Laboratory Evolution

• Extreme Pathways
New Cobra Toolbox Functions

% Cobra Sampling Function
[sampleStructOut, mixedFraction] = gpSampler(sampleStruct, nPoints, bias, maxTime, maxSteps)

% Plotting histogram (Matlab function)
hist(sampleStruct.points(rxnID,:),50);

% Compare flux histograms for one or more samples for one or more reactions
plotSampleHist(rxnNames,samples,models,nBins,perScreen)

% Plot scatter matrix
sampleScatterMatrix(rxnList,model,sampleStruct.points,250);

% Identify correlated reaction sets
[setsSorted,setNoSorted,setSize] = identifyCorrelSets(model,sampleStruct.points);

% MOMA
[solutionDel,solutionWT,totalFluxDiff,solStatus] = MOMA(modelWT,modelMutant,'max',false)
Extra Slides
Scatter Matrix Results

Sampling_ScatterMatrix.m
Scatter Matrix Results

Sampling_ScatterMatrix.m

- Scatter Matrix Results
- Sampling_ScatterMatrix.m
- Constraint-based Metabolic Reconstructions & Analysis
- g6p <-> f6p
- fdp + h2o --> f6p + pi
- atp + f6p --> adp + fdp + h
- fdp <=> dhap + g3p
- dhap <=> g3p
- g3p + nad + pi <=> 13dpg + h + nadh
- 3pg + atp <=> 13dpg + adp
- 2pg <=> 3pg
- 2pg <=> h2o + pep
- adp + h + pep --> atp + pyr
Network-based Pathways of a Sample Reaction Network

- The reaction network is represented by a stoichiometric matrix (S) with rows representing the participation of metabolites in reactions and columns as the stoichiometric coefficients for the individual reactions.
- Matrix is analyzed with pathway analysis (elementary modes or extreme pathways).
- These pathways can be represented in a matrix (P) where the rows represent the fluxes through corresponding reactions and the columns are the resultant pathways.
- Pathways can be illustrated for simple networks (P1, P2, P3).

The simple reaction network can be decomposed into three extreme pathways (ExPa) and four elementary modes (EIMo).

The extreme pathways and elementary modes satisfy the two shared requirements: each set of pathways is non-decomposable and each set of pathways is unique.

The extreme pathways are systemically independent; note that EIMo 1 (ExPa 1) and EIMo 2 (ExPa 2) can be combined to give EIMo 4.

Ethanol Production

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

% Set carbon source and oxygen uptake rates for wild type model
model = changeRxnBounds(model, 'EX_glc(e)', -5, 'l');
model = changeRxnBounds(model, 'EX_o2(e)', -20, 'l');
model = changeObjective(model, 'Biomass_Ecoli_core_N(w/GAM)_Nmet2');
FBAsolution = optimizeCbModel(model, 'max', 0, 0);
modelWT = model;

% Knockout reactions for mutant model
model = changeRxnBounds(model, 'NADH16', 0, 'b');
model = changeRxnBounds(model, 'PTAr', 0, 'b');
model = changeRxnBounds(model, 'TKT2', 0, 'b');
Mutantsolution = optimizeCbModel(model, 'max', 0, 0);
modelMutant = model;

% Sample model
[sampleStruct, mixedFrac] = gpSampler(model, 5000, [], 120);

% model = modelWT;
% model = modelMutant;

EthanolProduction_GDLS_MOMA_Sampling.m
Ethanol Production Example

\[ \text{EX}_{\text{glc}} > -5; \ \text{EX}_{\text{o2}} > -20; \ \text{Knockouts} = \{\text{NADH16, PTAr, TKT2}\}; \ \text{Points} = 5000, \ \text{MaxTime} = 120 \]
Ethanol Production Example

Objective Function set to the Maximum Value used in FVA

EX_glc(e) > -5; EX_o2(e) > -20; Knockouts = {NADH16, PTAr, TKT2}; Points = 5000, MaxTime = 120
Using MOMA results as Lower Constraint for Ethanol Production

EthanolProduction_GDLS_MOMA_Sampling.m

EthanolProduction_ProdEnvelope_GDLS_Mutants.m