



# 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

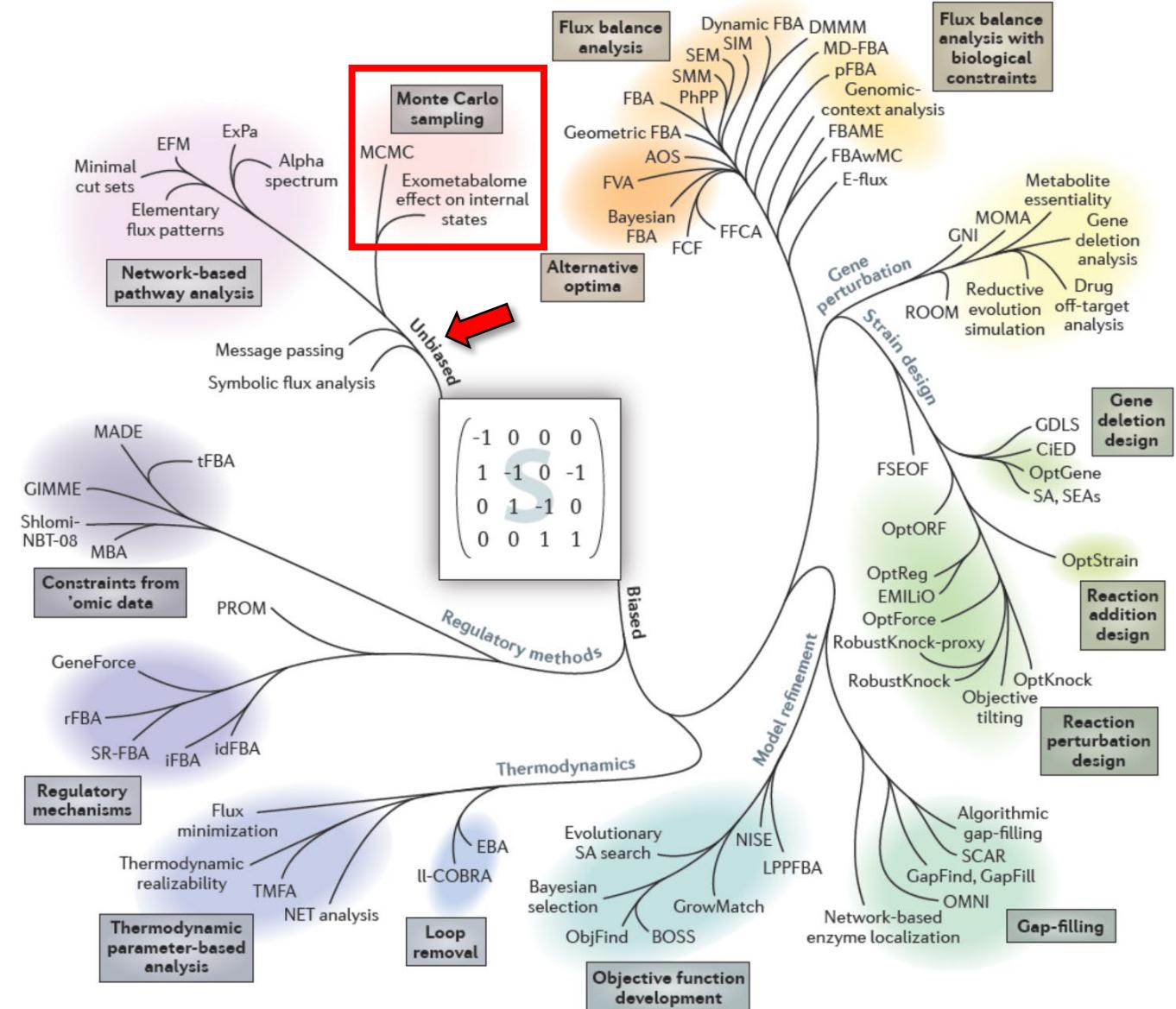


# Lesson Outline

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



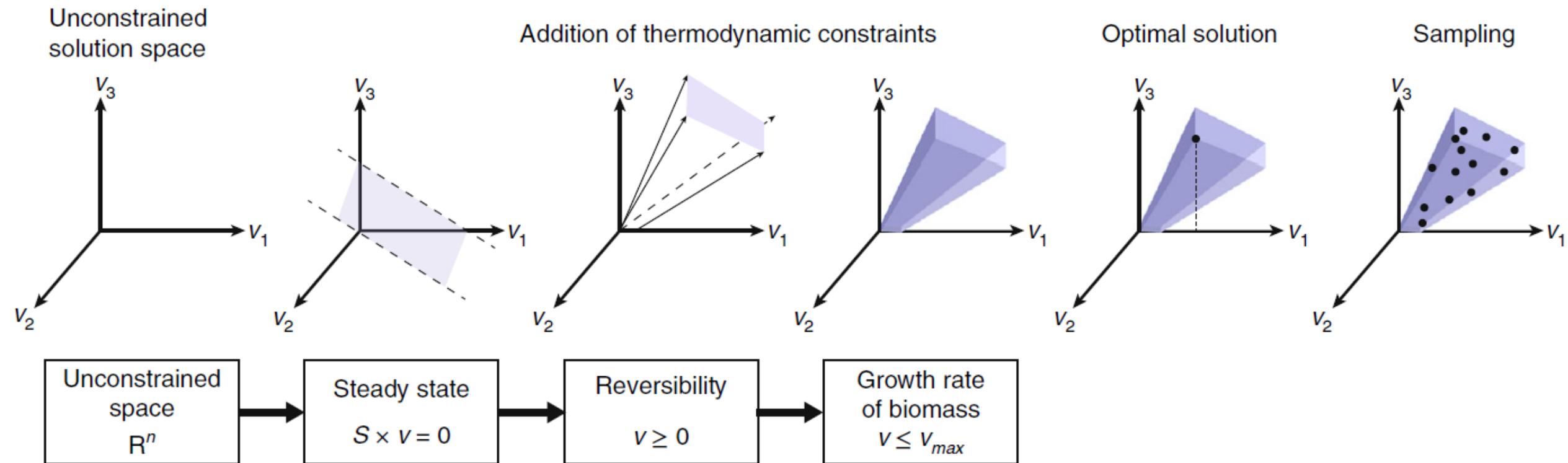
# The 'Phylogeny' of Constraint-based Modeling Methods



Lewis, N. E., H. Nagarajan, et al. (2012). "Constraining the metabolic genotype-phenotype relationship using a phylogeny of in silico methods." *Nature reviews Microbiology* 10(4): 291-305.



# Solution Spaces

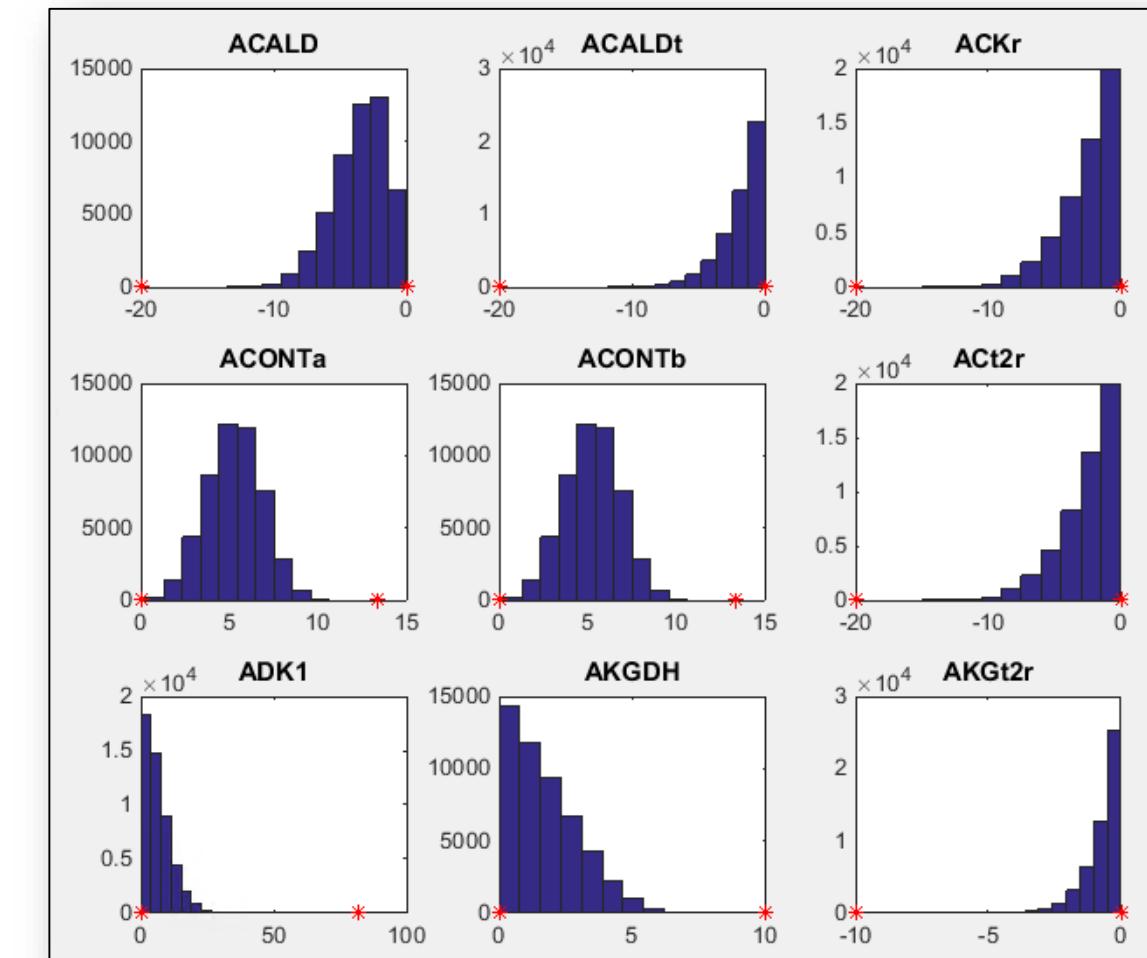


Laurent Heirendt et al, Creation and analysis of biochemical constraint-based models: the COBRA Toolbox v3.0, Nature Protocols, volume 14, pages 639-702, 2019



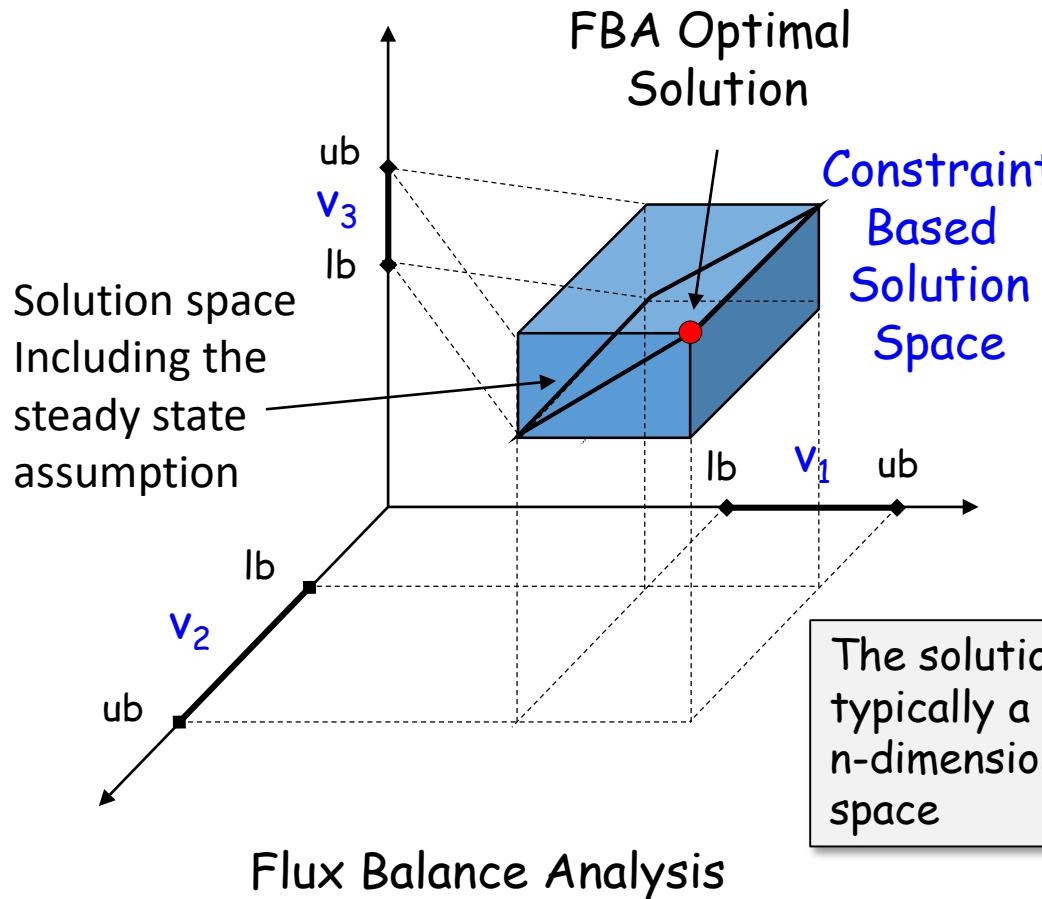
# Uniform Random Sampling

- 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.



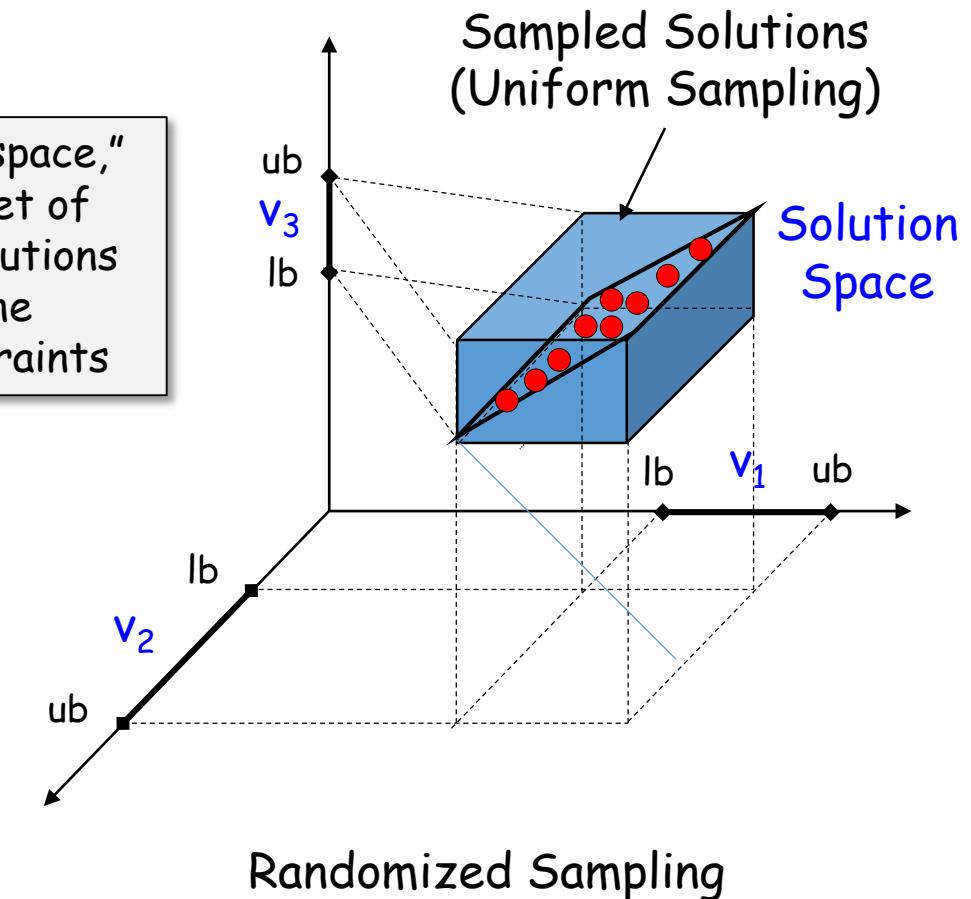


# Solution Space



Identifies one solution from all solution space

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



Multiple solutions representing all solution space

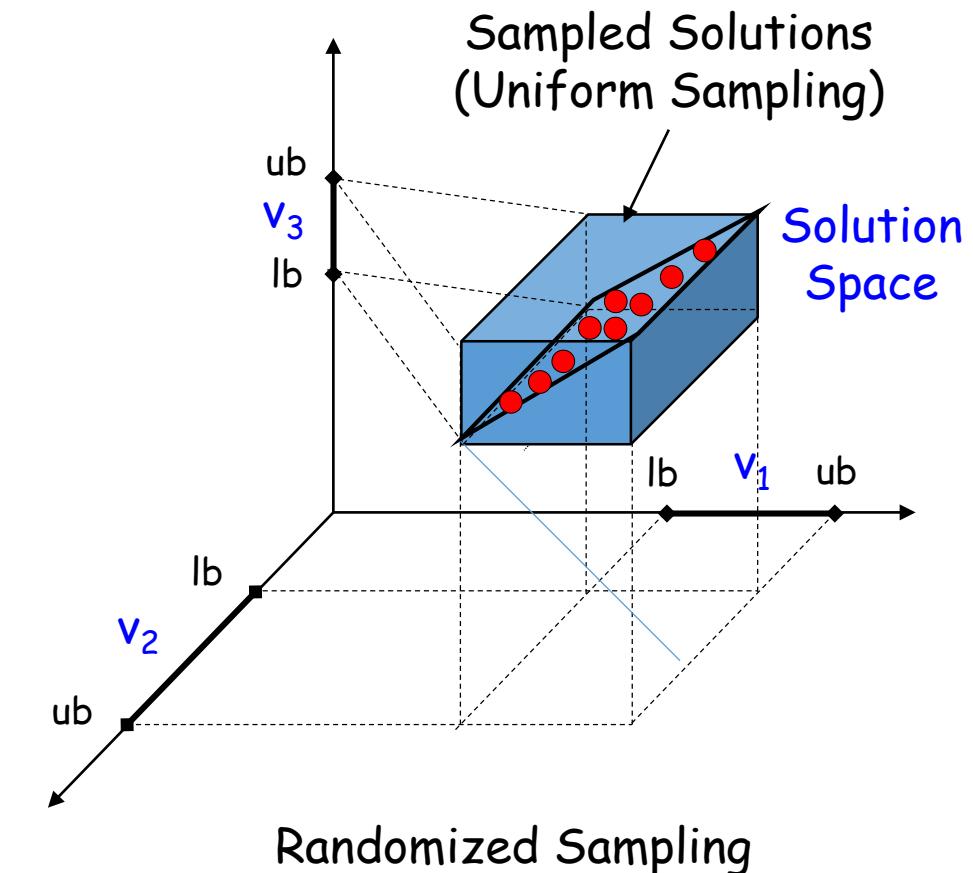


# Sampling of the Steady State Solution Space

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.

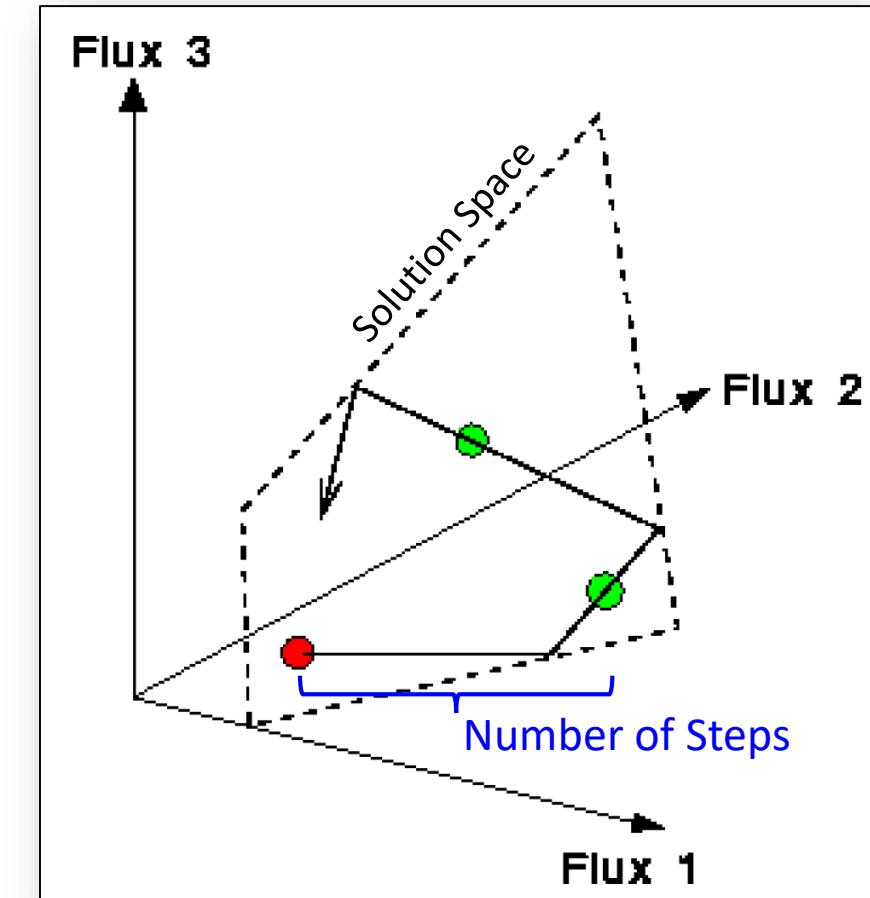
Jan Schellenberger, PhD Dissertation, University of California, San Diego, 2010





# 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.

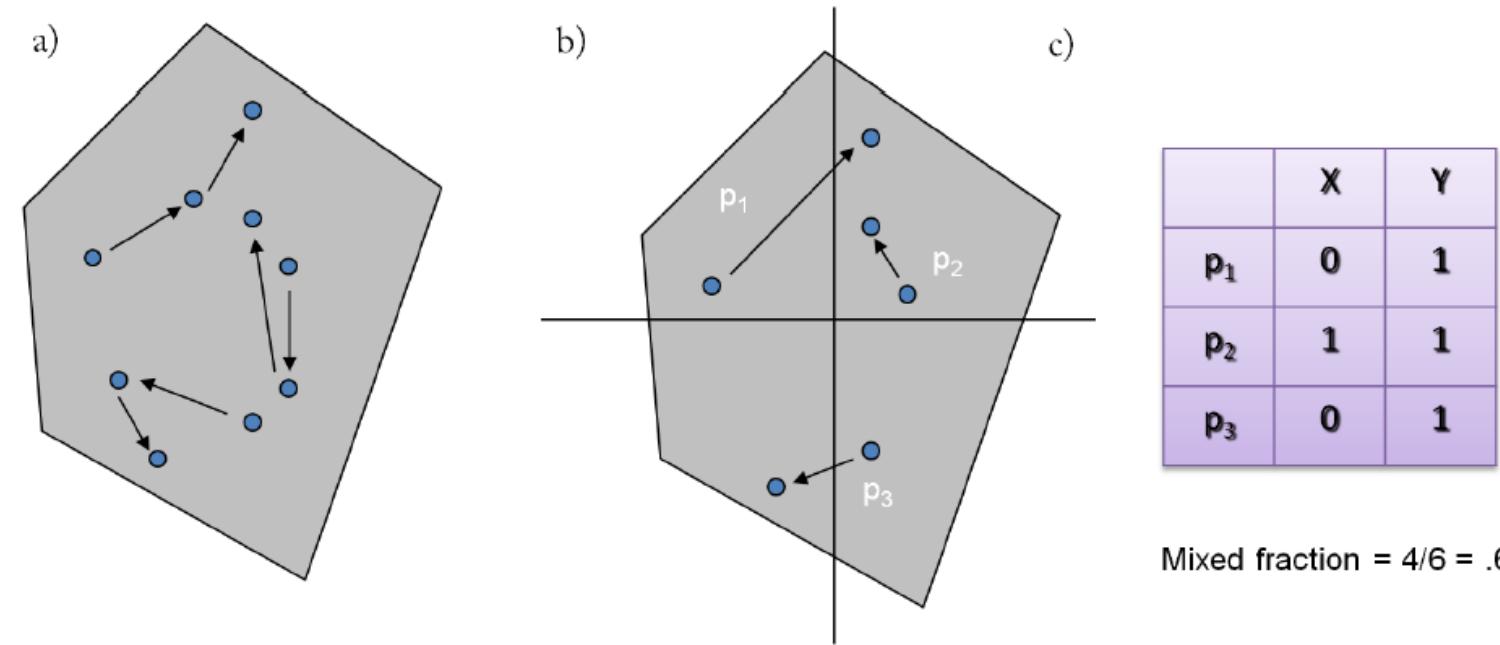


Almaas, E., B. Kovacs, et al. (2004). "Global organization of metabolic fluxes in the bacterium Escherichia coli." Supplementary Material, Nature 427(6977): 839-843.



# 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

```
[sampleStructOut, mixedFraction] = 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
model = readCbModel('ecoli_core_model.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-
% 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);

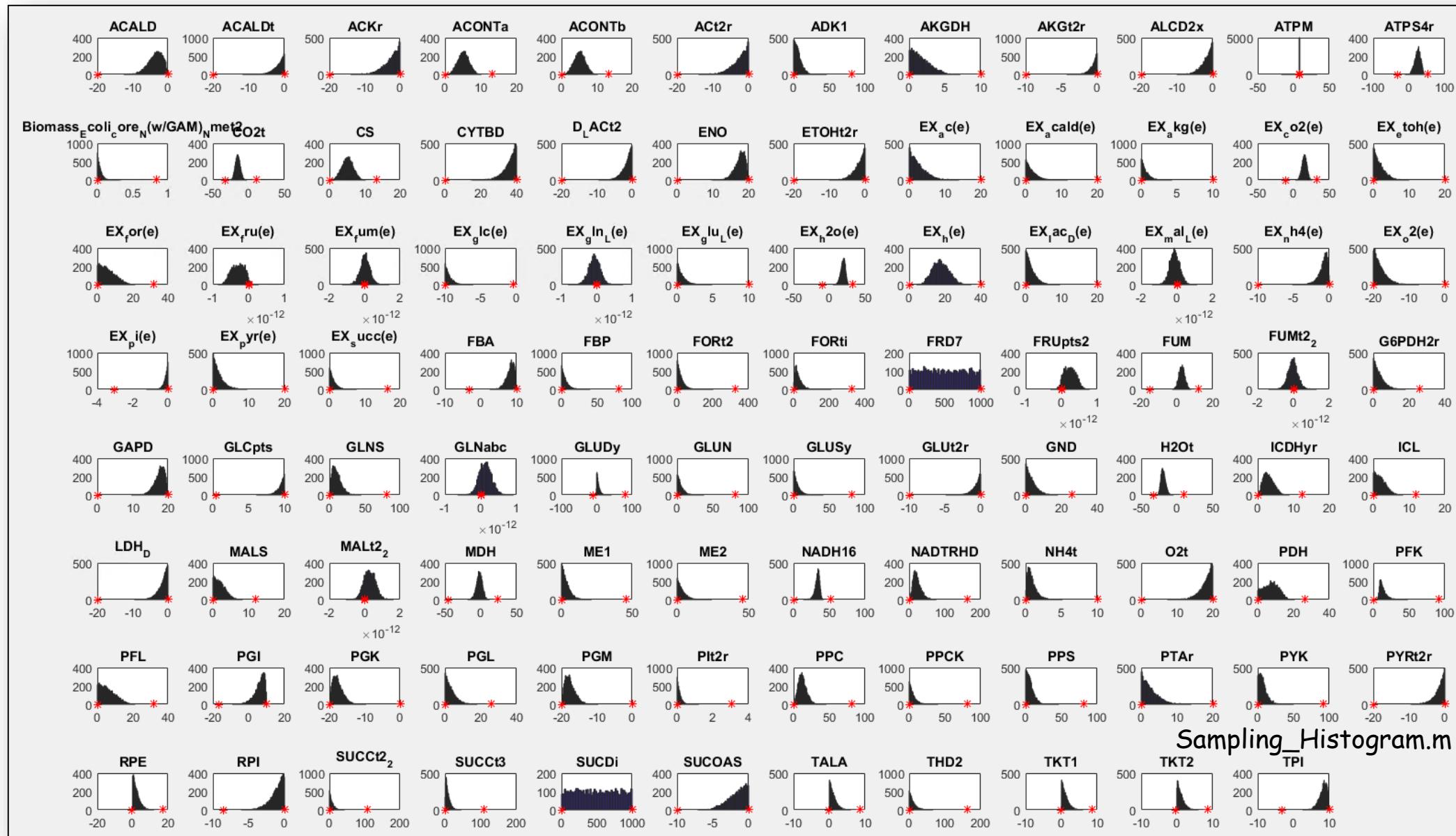
for i = 1 : 95
    subplot(8,12,i)
    hist(sampleStruct.points(i,:),50);
    hold on
    plot([minFlux(i) maxFlux(i)], [0 1],'*r');
    title(model.rxn{1});
end
```



# Constraint-based Metabolic Reconstructions & Analysis

2021 H. Scott Hinton

13





# sampleStruct in Matlab Desktop

Workspace

Name	Value
i	95
maxFlux	95x1 double
minFlux	95x1 double
mixedFrac	0.5178
model	1x1 struct
sampleStruct	1x1 struct

Sampling\_Histogram.m

sampleStruct		sampleStruct.points							
Field	Value	4994	4995	4996	4997	4998	4999	5000	5001
modelVersion	1x1 struct	1	-4.9664	-3.0282	-5.1064	-2.6685	-5.7000	-1.8900	-0.9636
rxns	95x1 cell	2	-2.5519	-2.1618	-2.5663	-2.3402	-1.1459	-1.4780	-0.0908
mets	72x1 cell	3	-3.7079	-0.1409	-1.4245	-2.4573	-2.0253	-0.1233	-3.2611
S	72x95 sparse double	4	5.5826	4.6493	5.9659	7.0335	3.9708	6.0616	5.7890
rev	95x1 double	5	5.5826	4.6493	5.9659	7.0335	3.9708	6.0616	5.7890
c	95x1 double	6	-3.7079	-0.1409	-1.4245	-2.4573	-2.0253	-0.1233	-3.2611
metNames	72x1 cell	7	5.3602	5.2787	20.3048	2.5210	0.7912	8.0200	1.5332
metFormulas	72x1 cell	8	3.0538	1.7533	0.4203	3.8729	1.2061	3.8089	2.5084
lb	95x1 double	9	-0.8316	-1.0255	-0.7326	-0.7649	-0.2090	-1.0182	-0.0678
ub	95x1 double	10	-2.4145	-0.8664	-2.5401	-0.3283	-4.5542	-0.4120	-0.8728
metCharge	72x1 int32	11	8.3900	8.3900	8.3900	8.3900	8.3900	8.3900	8.3900
rules	95x1 cell	12	21.9175	30.7517	37.4636	39.1198	23.0298	25.8820	15.8942
genes	137x1 cell	13	0.0710	0.0959	0.0173	0.0750	0.0498	0.0356	0.0519
rxnGeneMat	95x137 sparse double	14	-14.7393	-17.0052	-18.2290	-15.7290	-17.7635	-16.5790	-12.6169
grRules	95x1 cell	15	5.5826	4.6493	5.9659	7.0335	3.9708	6.0616	5.7890
subSystems	95x1 cell	16	36.4211	37.4837	36.6684	39.5002	35.1127	38.8526	34.7587
confidenceScores	95x1 cell	17	-0.1734	-2.0901	-0.1301	-0.7185	-1.1033	-1.0282	-0.1231
rxnReferences	95x1 cell	18	17.9763	14.0917	17.3390	16.0641	17.0479	15.5371	16.5333
rxnECNumbers	95x1 cell	19	-2.4145	-0.8664	-2.5401	-0.3283	-4.5542	-0.4120	-0.8728
rxnNotes	95x1 cell	20	3.7079	0.1409	1.4245	2.4573	2.0253	0.1233	3.2611
rxnNames	95x1 cell	21	2.5519	2.1618	2.5663	2.3402	1.1459	1.4780	0.0908
metChEBIID	72x1 cell	22	0.8316	1.0255	0.7326	0.7649	0.2090	1.0182	0.0678
metKEGGID	72x1 cell	23	14.7393	17.0052	18.2290	15.7290	17.7635	16.5790	12.6169
metPubChemID	72x1 cell	24	2.4145	0.8664	2.5401	0.3283	4.5542	0.4120	0.8728
metInChIString	72x1 cell	25	10.8175	3.3370	2.6250	8.0081	5.9937	0.9814	5.5340
b	72x1 double	26	-5.2403e-14	-1.6520e-13	-4.1211e-13	-4.2988e-13	-5.2580e-13	-3.6238e-13	-3.0376e-13
description	'coli_textbook'	27	-7.6383e-14	-1.4122e-13	-1.0658e-13	4.9738e-14	3.7659e-13	2.1316e-14	2.9843e-13
A	72x95 sparse double	28	-9.8893	-8.2680	-8.8847	-8.2942	-9.4990	-8.1017	-8.8687
csense	'EEEEEEEEEEEEEEEEE...'								
internal	1x1 struct								
warmupPts	95x5000 double								
steps	700								
points	95x5000 double								



# Succinate FVA Example

```
% Sampling_Succ_Histogram.m
clear;
model = readCbModel('ecoli_core_model.mat');
model = changeRxnBounds(model,'EX_glc(e)',0,'1');
model = changeRxnBounds(model,'EX_o2(e)',-40,'1');
model = changeRxnBounds(model,'EX_succ(e)',-20,'1');
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.rxnIDs,fal-
se,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
```

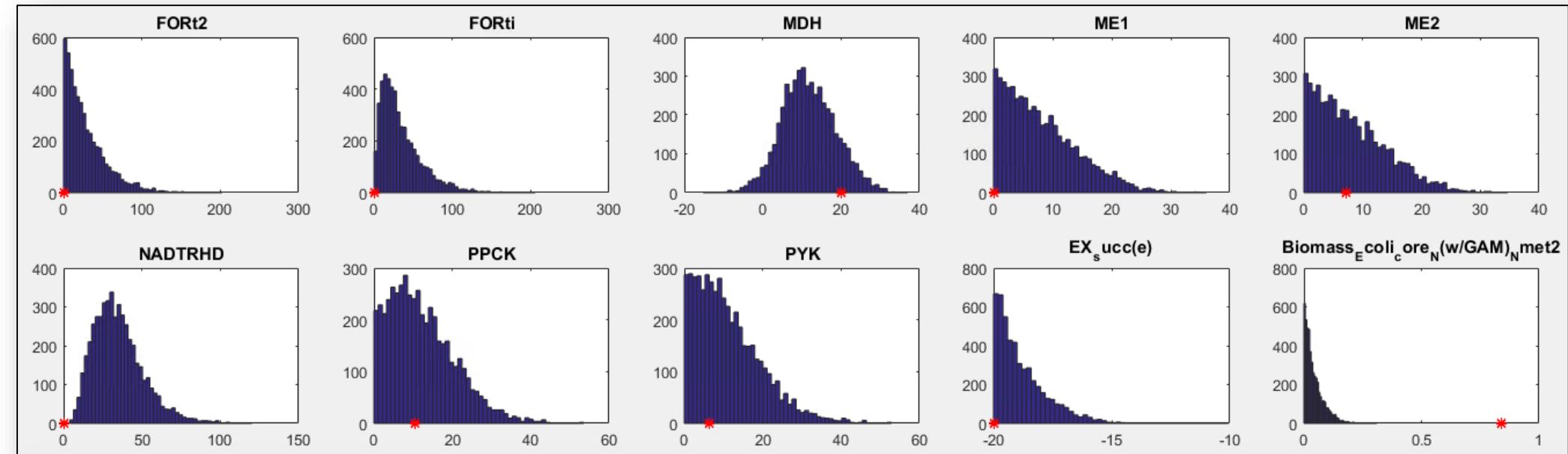


# Constraint-based Metabolic Reconstructions & Analysis

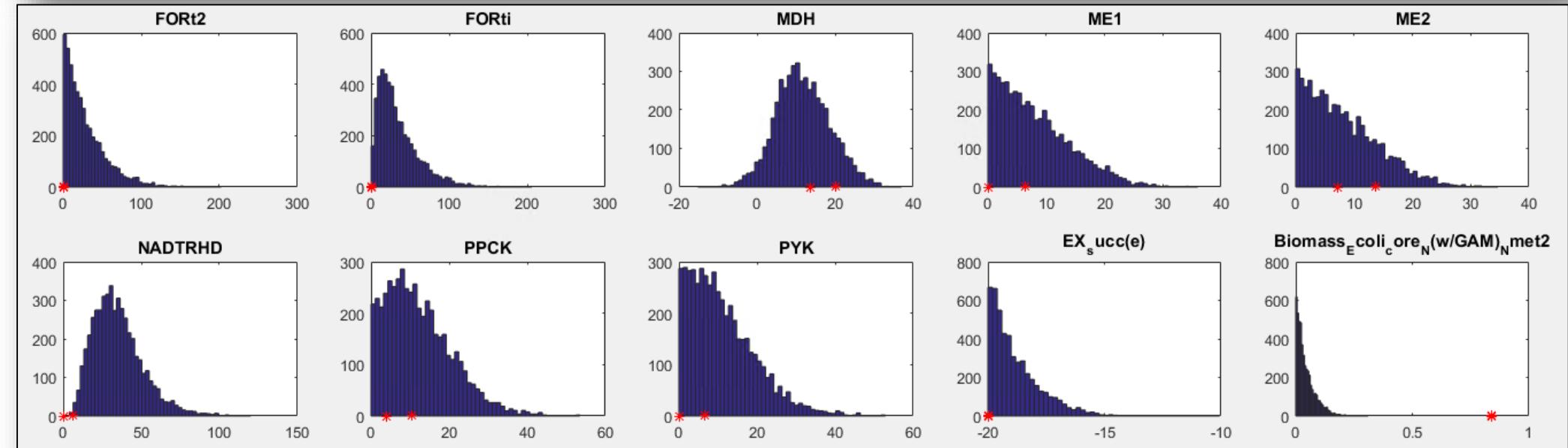
2021 H. Scott Hinton

16

Optimal Flux Values  
Included (\*)



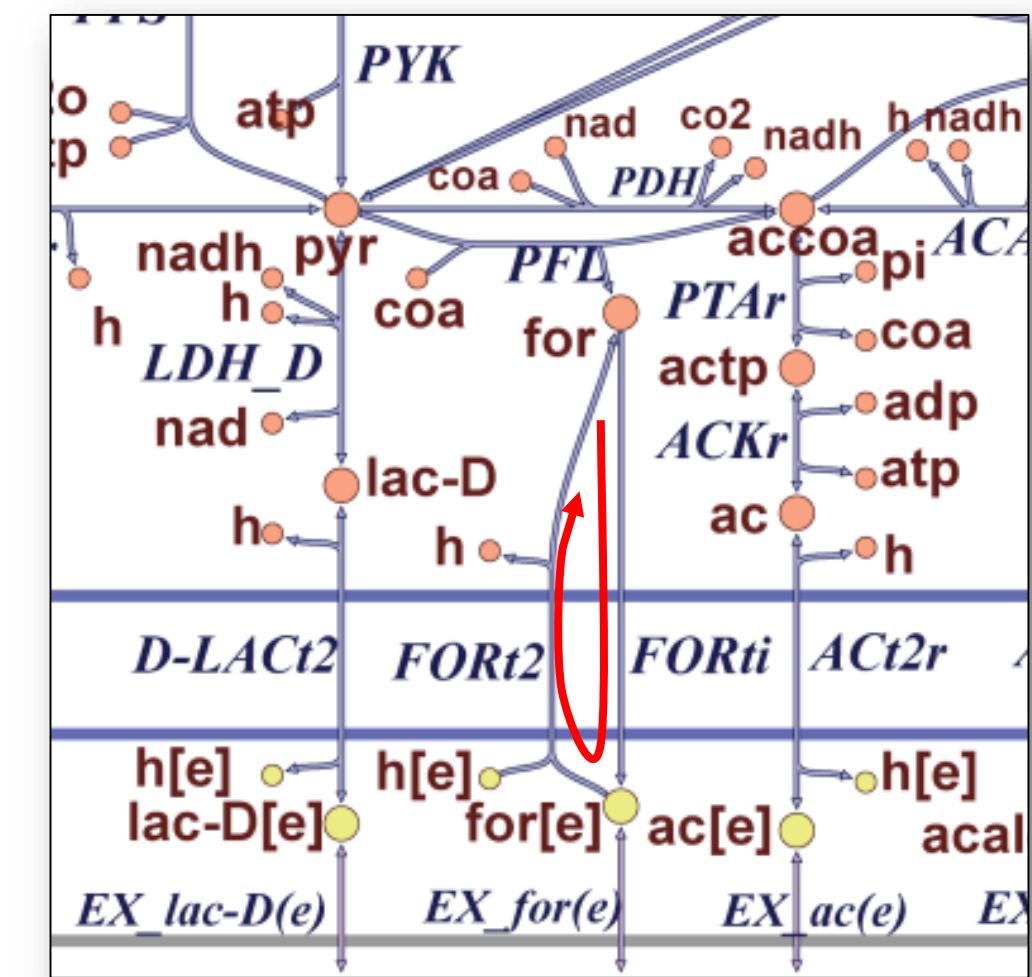
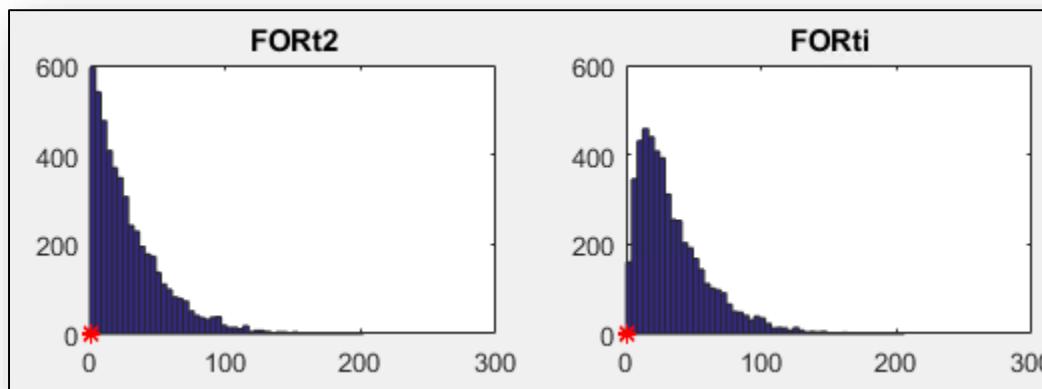
Flux Variability  
Analysis Min and Max  
Values Included (\*)



Sampling\_Succ\_Histogram.m



# Why are the Fluxes for FORt2 and FORti so large?





# Comparing Sampling Results to Optimized Results

(Succinate Example from FVA)

```
% Sampling_Succ_Optimal.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');

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

% Force output to be close to optimal value
model = changeRxnBounds(model,'Biomass_Ecoli_core_N(w/GAM)-Nmet2',
FBAsolution.f,'b');
% model =
changeRxnBounds(model,'Biomass_Ecoli_core_N(w/GAM)_Nmet2',
0.9*FBAsolution.f,'l');
```

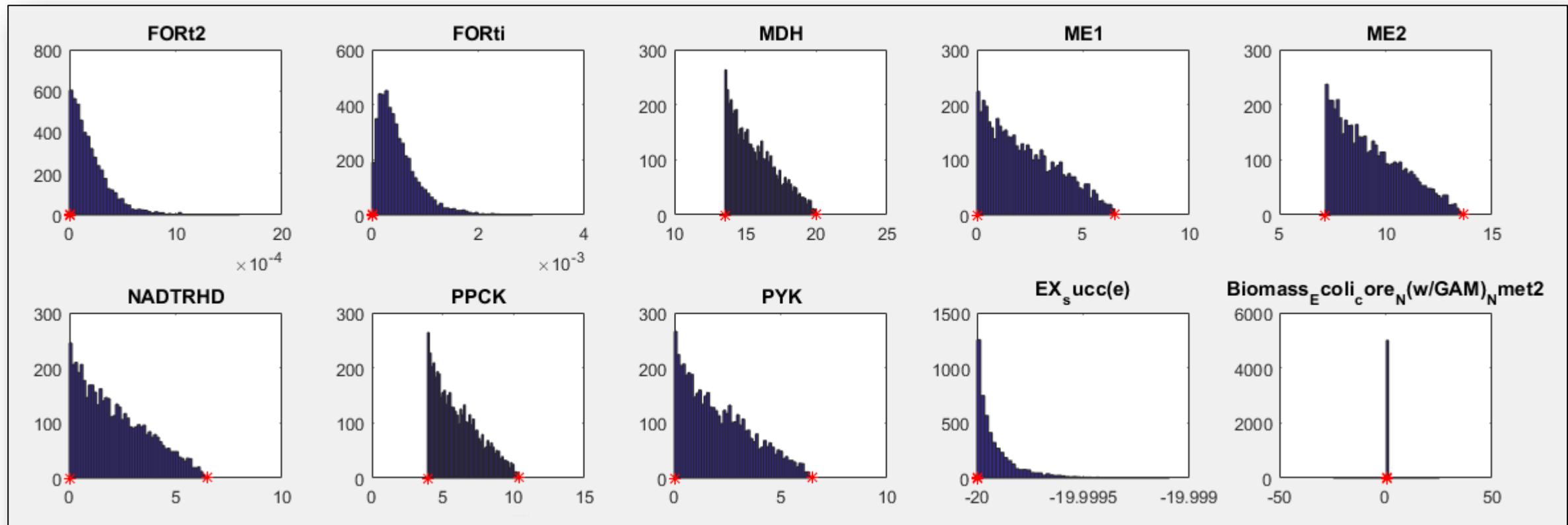
```
% Sample model
[sampleStruct,mixedFrac] = gpSampler(model,5000,[],120);

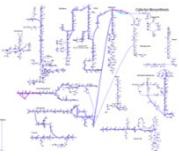
% Plot histograms for selected reactions
rxnList = {'FORT2', 'FORTi', 'MDH', 'ME1', 'ME2', 'NADTRHD',
'PPCK', 'PYK', 'EX_succ(e)', 'Biomass_Ecoli_core_N(w/GAM)-Nmet2'};

% Include flux variability analysis mins and maxs on the histogram
[minFlux,maxFlux]=fluxVariability(model,100,'max',model.rxnns,false,
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





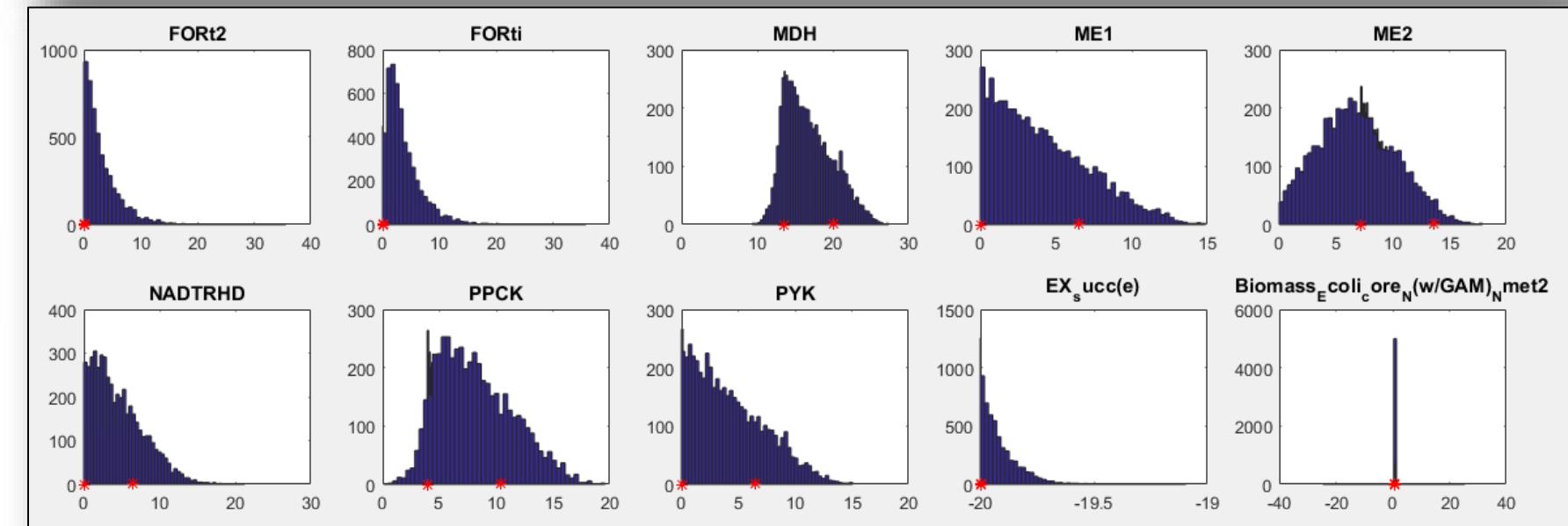
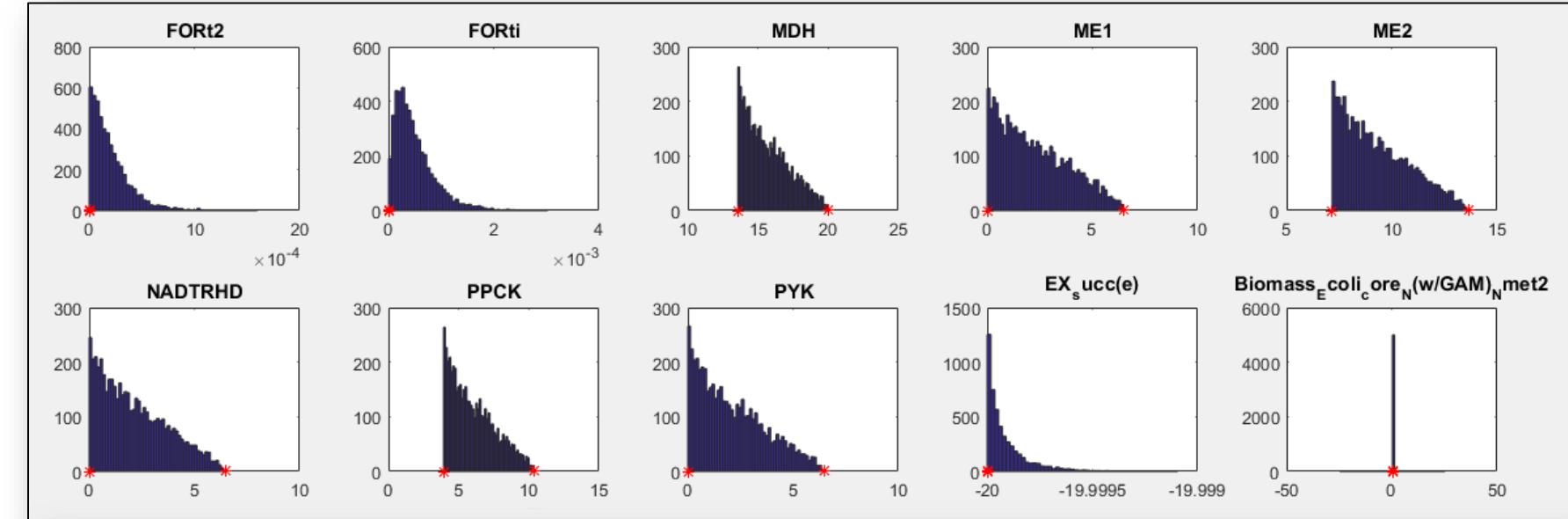
Force biomass function to be maximum growth rate

## Comparing Sampling Results to Optimized Results

Allow biomass function to be  $\geq 90\%$  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

```
plotSampleHist(rxnNames,samples,models,nBins,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))

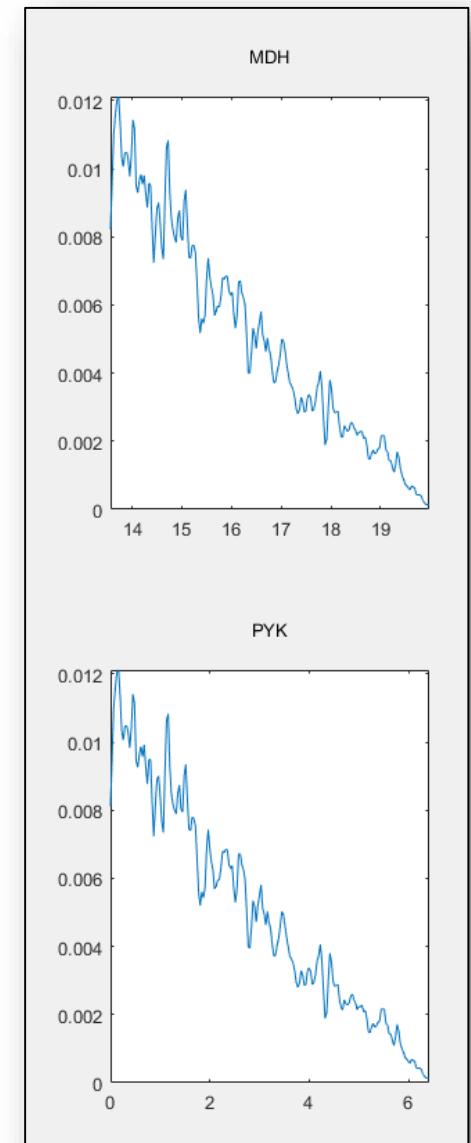
perScreen Number of reactions to show per screen. Either a number or [nY, nX] vector.  
(press Enter to advance screens)

## 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

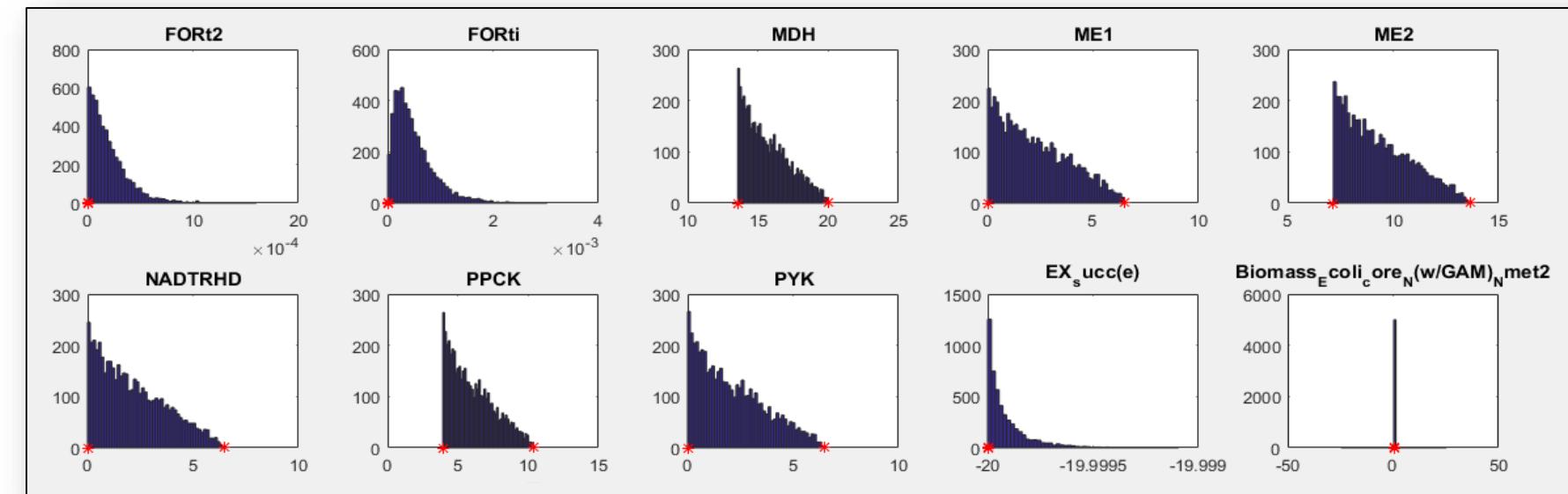
To quit script type q and hit enter/return



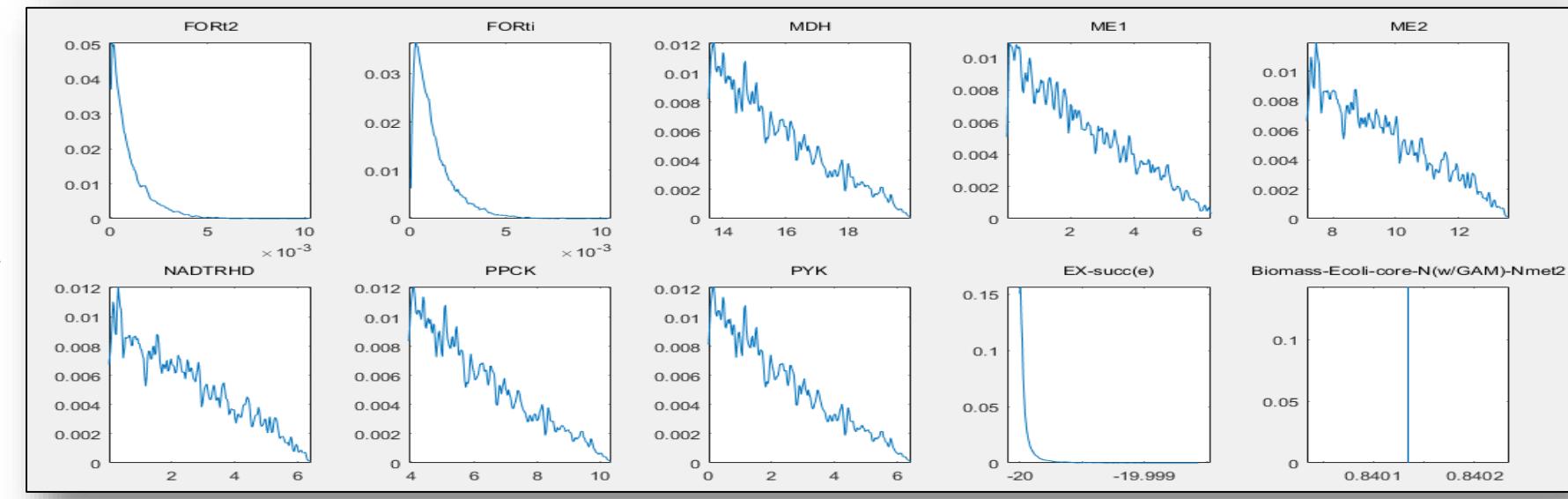


Comparing  
plotSampleHist  
with Hist plots

hist plot



plotSamplehist plot





# gpSampler: Comparing Phenotypes

```
% Sampling_Example_gpSampler.m
```

```
clear;
```

```
% Input the E.coli core model and set constraints
```

```
model = readCbModel('ecoli_core_model.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);
```

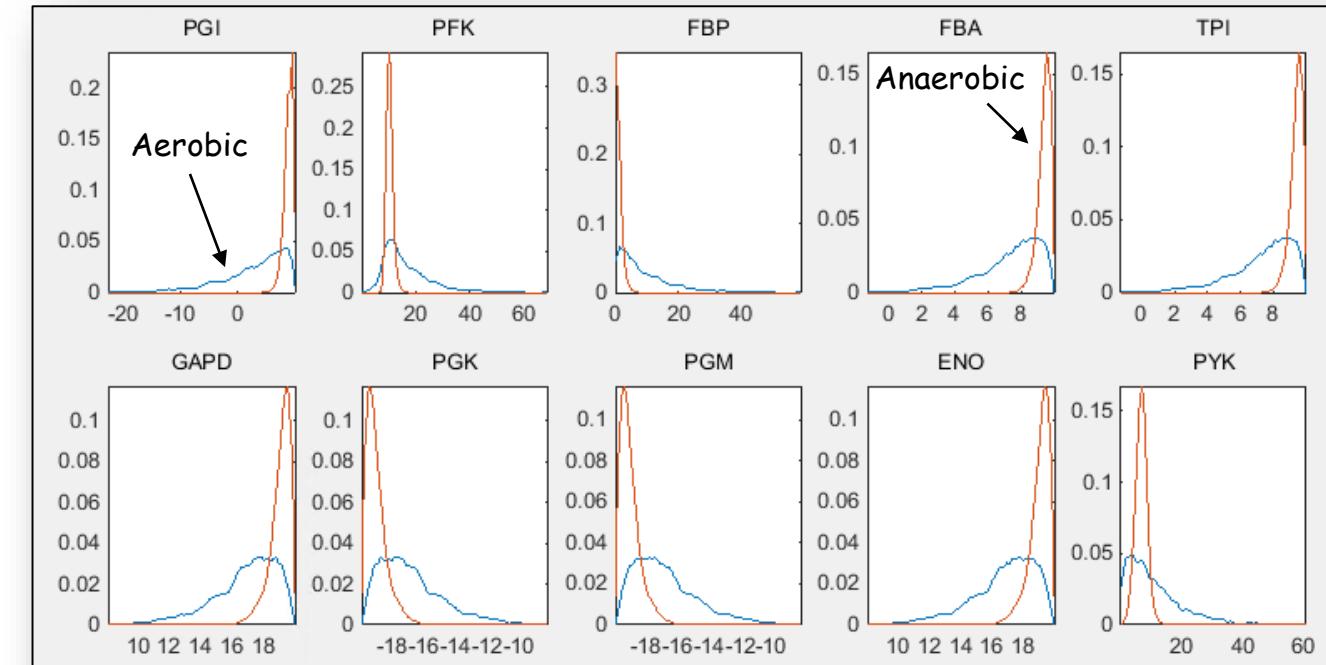
```
% Sampling results will be returned in the two structures
```

```
% sampleStruct_aerobic and sampleStruct_anaerobic within the field points.
```

```
% Visualize sampling results for a set of reactions.
```

```
rxnList = {'PGI', 'PFK', 'FBP', 'TPI', 'GAPD', 'PGK', 'PGM', 'ENO', 'PYK'};
```

```
plotSampleHist(rxnList, {sampleStruct_aerobic.points, sampleStruct_anaerobic.points}, {model_aerobic, ...
    model_anaerobic},[],[2,5]);
```



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

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



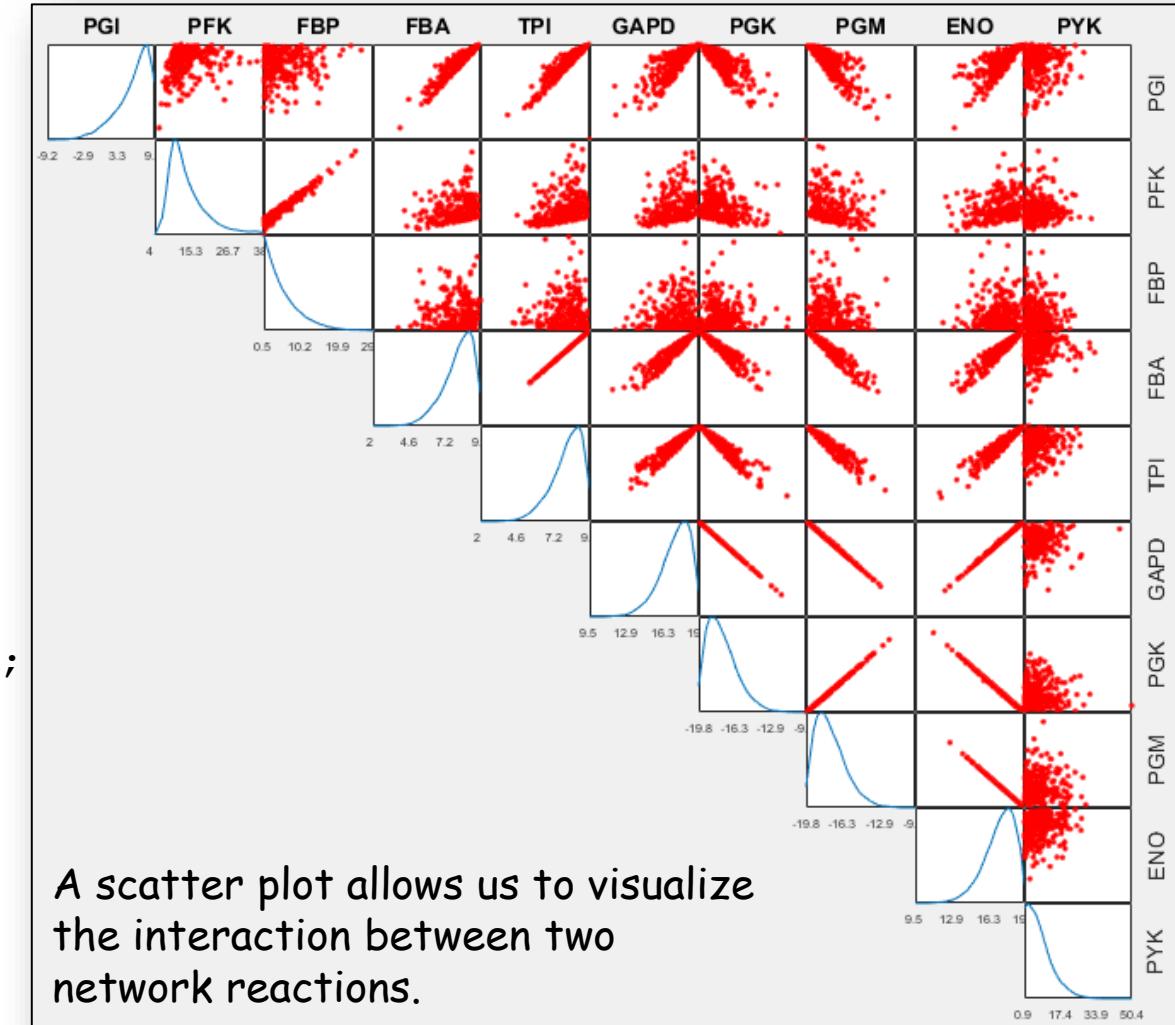
# Scatter Matrix

```
% Sampling_ScatterMatrix.m
clear;

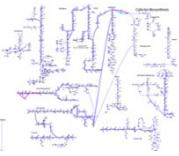
% Input the E.coli core model and set constraints
model = readCbModel('ecoli_core_model.mat');
model = changeRxnBounds(model,'EX_glc(e)',-10,'l');
model = changeRxnBounds(model,'EX_o2(e)',-20,'l');

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

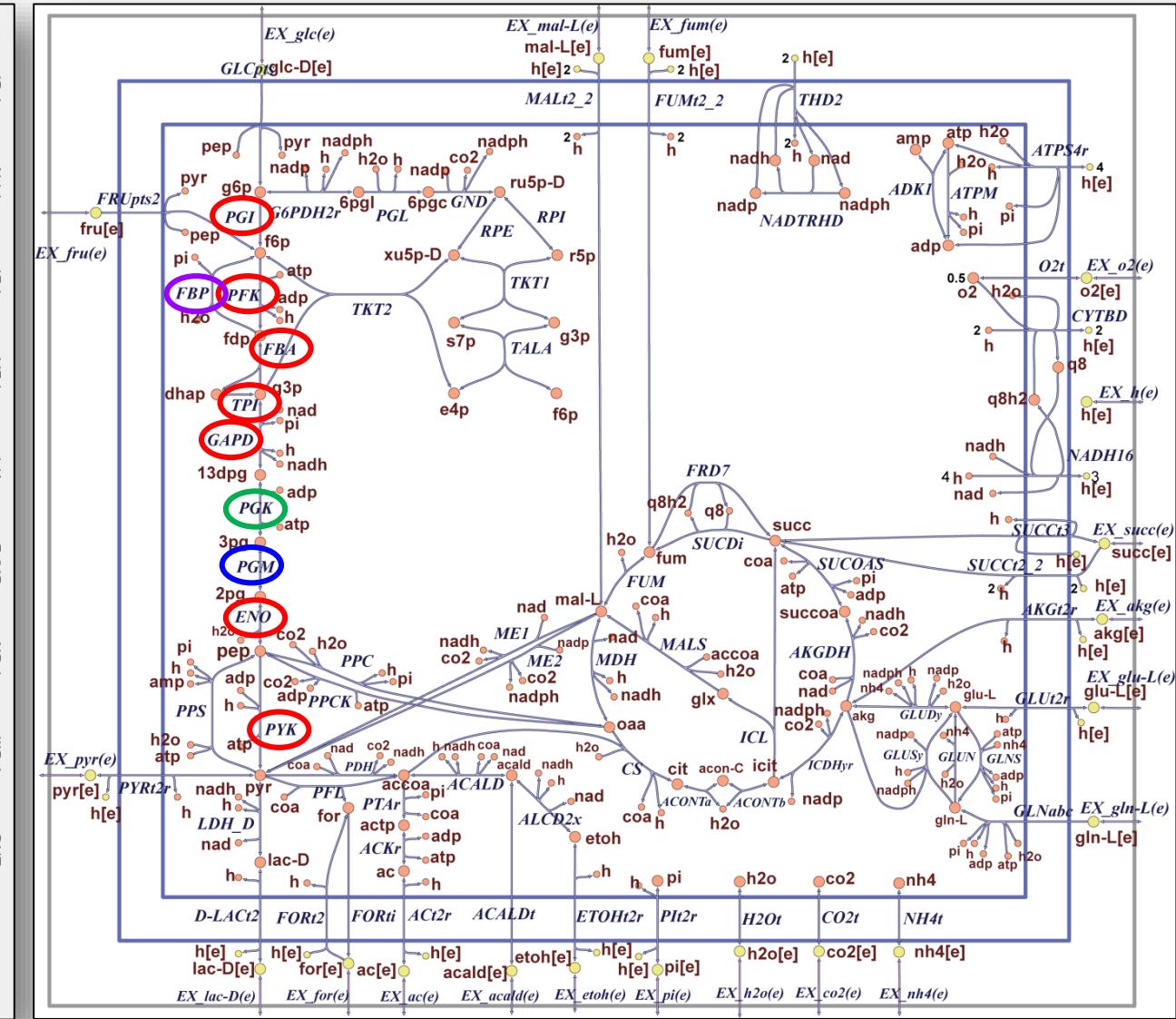
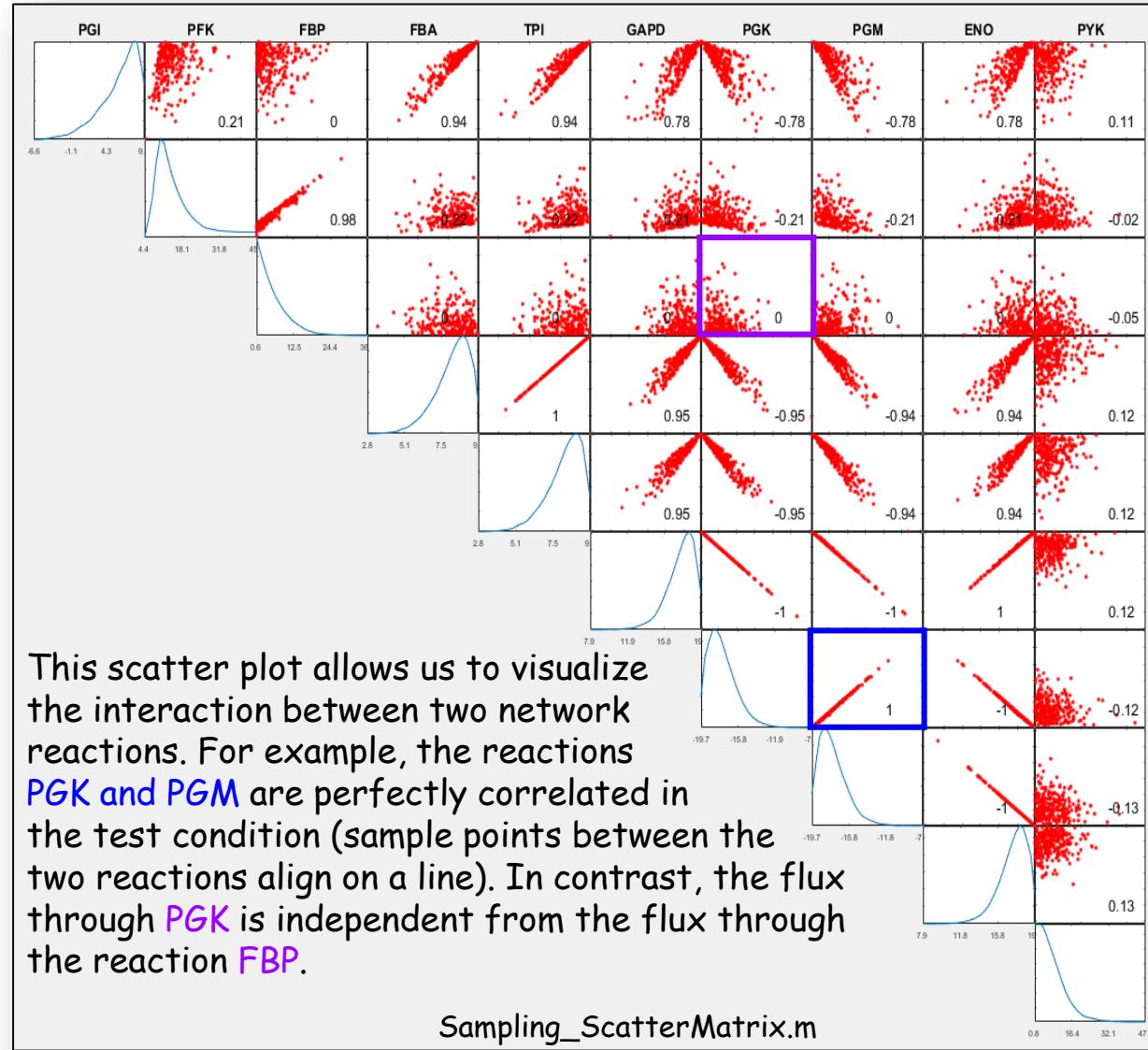
% Plot scatter matrix
rxnList = {'PGI', 'PFK', 'FBP', 'FBA', 'TPI', 'GAPD', 'PGK', 'PGM', 'ENO', 'PYK'};
sampleScatterMatrix(rxnList,model,sampleStruct.points,250);
```



A scatter plot allows us to visualize the interaction between two network reactions.



# Scatter Matrix Results





# 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
model = readCbModel('ecoli_core_model.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');

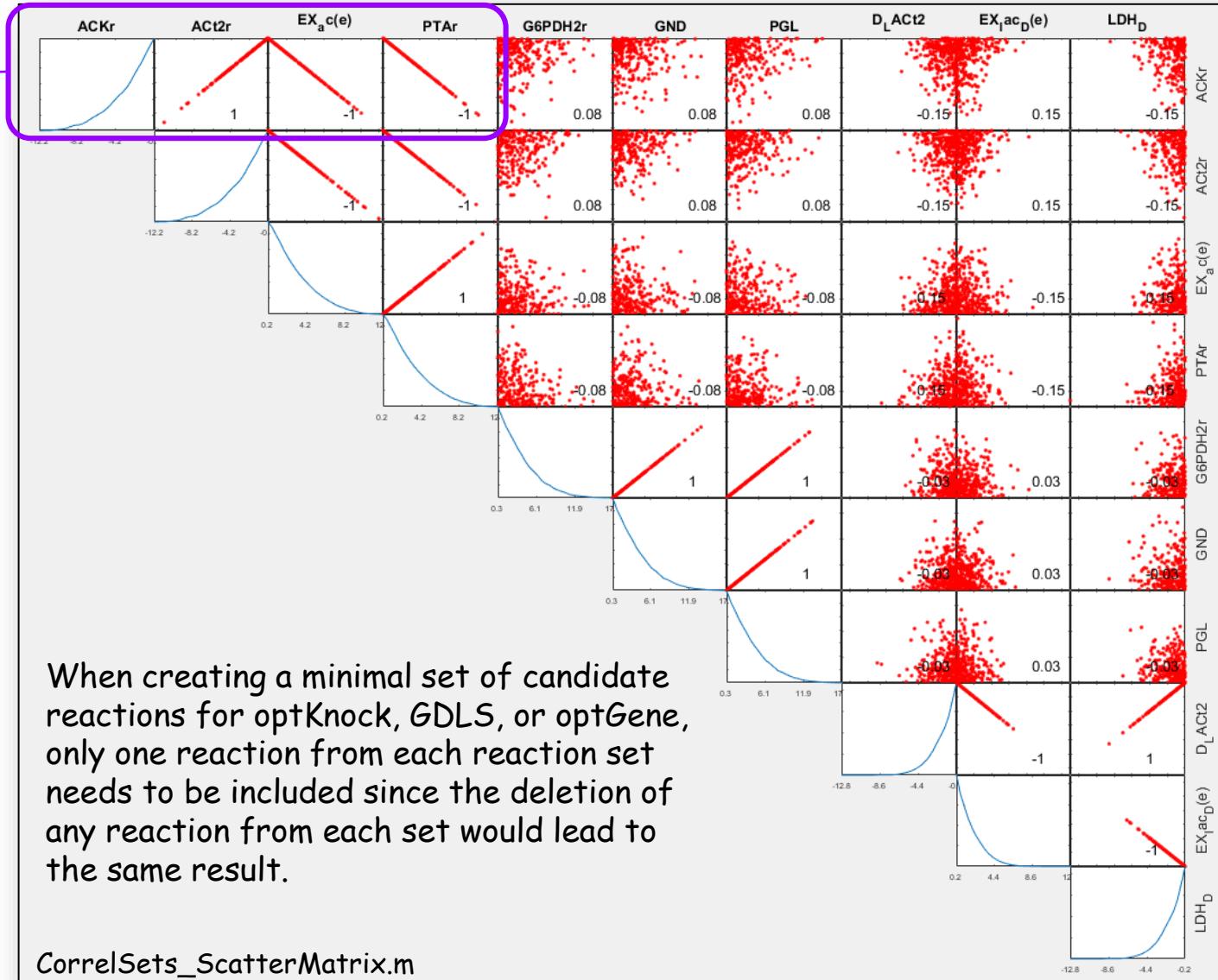
[sampleStruct,mixedFrac] = gpSampler(model,5000,[],120);
[setsSorted,setNoSorted,setSize] = identifyCorrelSets(model,sampleStruct.points);
setNames = [];
setNumbers = [];
disp('Correlated Reations Sets')
for i = 1 : length(setsSorted)
    setNames = [setNames; setsSorted{i}.names];
    setNumbers = [setNumbers;i*ones(length(setsSorted{i}.names),1)];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

Set#	Reactions			
[1]	'ACKr'	'ACT2r'	'EX_ac(e)'	'PTAr'
[2]	'G6PDH2r'	'GND'	'PGL'	
[3]	'D_LACT2'	'EX_lac_D(e)'	'LDH_D'	
[4]	'CYTBD'	'EX_o2(e)'	'O2t'	
[5]	'Biomass'	'EX_pi(e)'	'PIt2r'	
[6]	'ALCD2x'	'ETOHt2r'	'EX_etoh(e)'	
[7]	'ACONTa'	'ACONTb'	'CS'	
[8]	'TALA'	'TKT1'		
[9]	'ICL'	'MALS'		
[10]	'GAPD'	'PGK'		
[11]	'FBA'	'TPI'		
[12]	'EX_pyr(e)'	'PYRt2r'		
[13]	'EX_nh4(e)'	'NH4t'		
[14]	'EX_h2o(e)'	'H2Ot'		
[15]	'EX_glu_L(e)'	'GLUt2r'		
[16]	'EX_glc(e)'	'GLCpts'		
[17]	'EX_for(e)'	'PFL'		
[18]	'ENO'	'PGM'		
[19]	'CO2t'	'EX_co2(e)'		
[20]	'AKGt2r'	'EX_akg(e)'		
[21]	'AKGDH'	'SUCOAS'		
[22]	'ADK1'	'PPS'		
[23]	'ACALDt'	'EX_acald(e)'		



## Correlated Reaction Sets Scatter Matrix

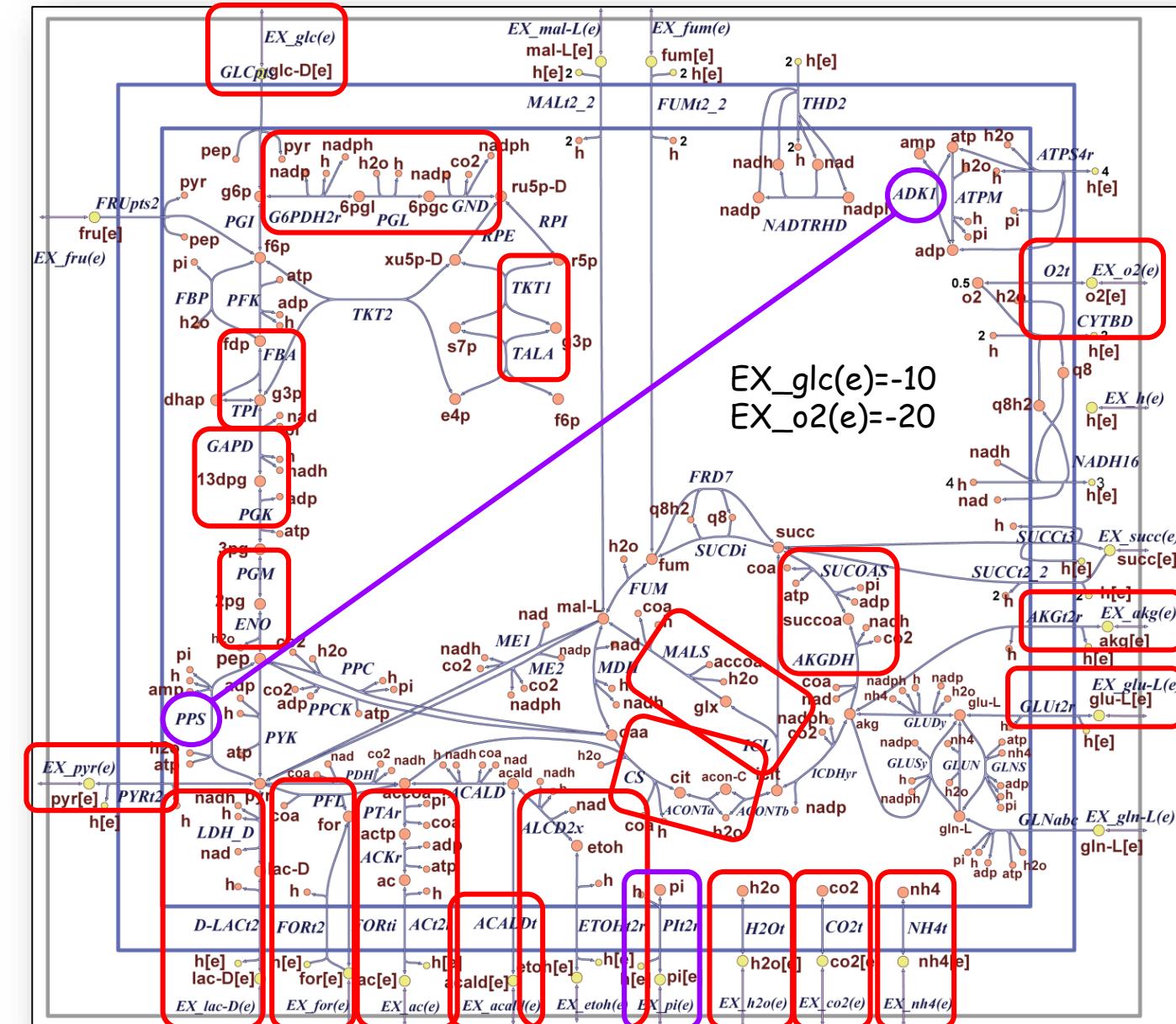
Set#	Reactions			
[1]	'ACKr'	'ACT2r'	'EX_ac(e)'	'PTAr'
[2]	'G6PDH2r'	'GND'	'PGL'	
[3]	'D_LACT2'	'EX_lac_D(e)'	'LDH_D'	
[4]	'CYTBD'	'EX_o2(e)'	'O2t'	
[5]	'Biomass'	'EX_pi(e)'	'PIt2r'	
[6]	'ALCD2x'	'ETOHt2r'	'EX_etooh(e)'	
[7]	'ACONTa'	'ACONTb'	'CS'	
[8]	'TALA'	'TKT1'		
[9]	'ICL'	'MALS'		
[10]	'GAPD'	'PGK'		
[11]	'FBA'	'TPI'		
[12]	'EX_pyr(e)'	'PYRt2r'		
[13]	'EX_nh4(e)'	'NH4t'		
[14]	'EX_h2o(e)'	'H2Ot'		
[15]	'EX_glu_L(e)'	'GLUt2r'		
[16]	'EX_glc(e)'	'GLCpts'		
[17]	'EX_for(e)'	'PFL'		
[18]	'ENO'	'PGM'		
[19]	'CO2t'	'EX_co2(e)'		
[20]	'AKGt2r'	'EX_akg(e)'		
[21]	'AKGDH'	'SUCOAS'		
[22]	'ADK1'	'PPS'		
[23]	'ACALDt'	'EX_acald(e)'		



# Mapped Correlated Reaction Sets

Set#	Reactions			
[1]	'ACKr'	'ACT2r'	'EX_ac(e)'	'PTA'
[2]	'G6PDH2r'	'GND'	'PGL'	
[3]	'D_LACT2'	'EX_lac_D(e)'	'LDH_D'	
[4]	'CYTBD'	'EX_o2(e)'	'O2t'	
[5]	'Biomass'	'EX_pi(e)'	'PIt2r'	
[6]	'ALCD2x'	'ETOHT2r'	'EX_etoh(e)'	
[7]	'ACONTa'	'ACONTb'	'CS'	
[8]	'TALA'	'TKT1'		
[9]	'ICL'	'MALS'		
[10]	'GAPD'	'PGK'		
[11]	'FBA'	'TPI'		
[12]	'EX_pyr(e)'	'PYRt2r'		
[13]	'EX_nh4(e)'	'NH4t'		
[14]	'EX_h2o(e)'	'H2Ot'		
[15]	'EX_glu_L(e)'	'GLUT2r'		
[16]	'EX_glc(e)'	'GLCpts'		
[17]	'EX_for(e)'	'PFL'		
[18]	'ENO'	'PGM'		
[19]	'CO2t'	'EX_co2(e)'		
[20]	'AKGt2r'	'EX_akg(e)'		
[21]	'AKGDH'	'SUCOAS'		
[22]	'ADK1'	'PPS'		
[23]	'ACALDt'	'EX_acald(e)'		

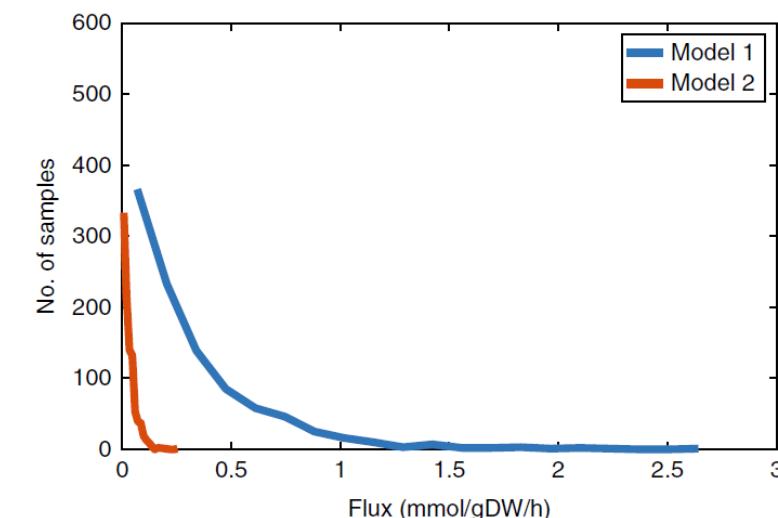
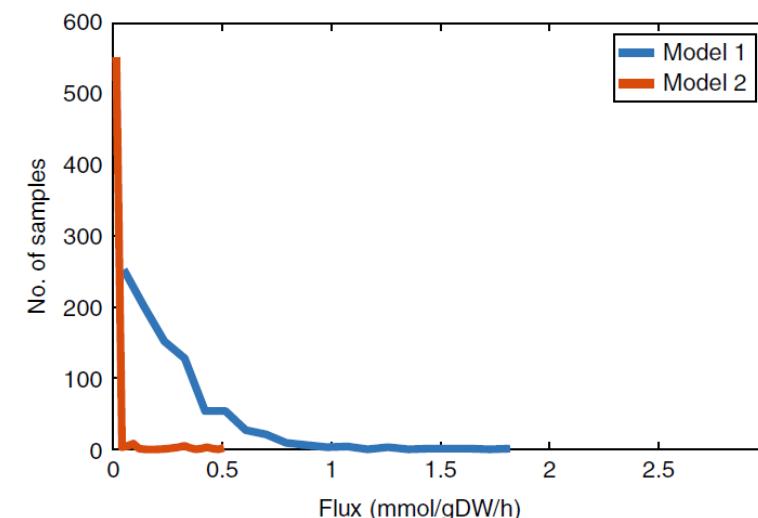
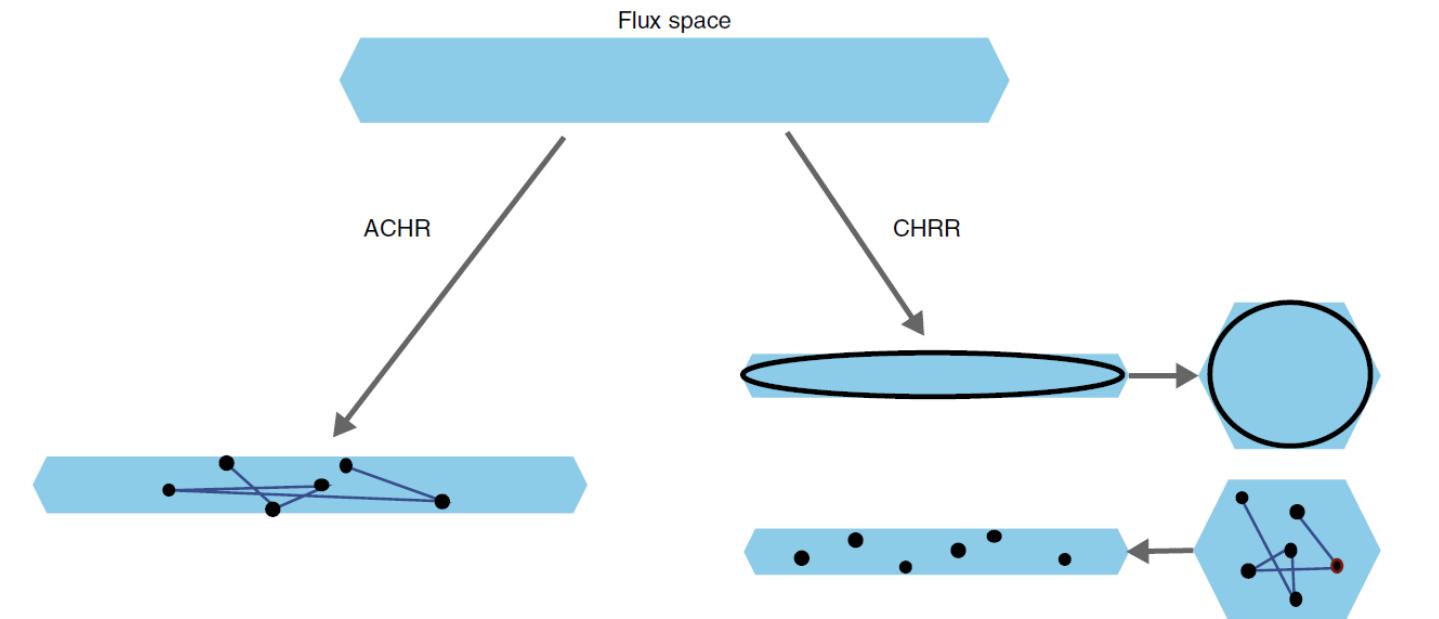
## IdentifyingCorrelSets.m





# ACHR vs. CHRR Algorithm

- Solution spaces from steady-state fluxes are often anisotropic; long in some directions and short in others. This impedes the ability of any sampling algorithm taking a random direction to evenly explore the full feasible set (artificial centering hit-and-run (ACHR) algorithm).
- The CHRR (coordinate hit-and-run with rounding) algorithm first rounds the solution space based on the maximum-volume ellipsoid. Then, the rounded solution space is uniformly sampled using a provably efficient coordinate hit-and-run random walk.
- Finally, the samples are projected back onto the anisotropic feasible set. This leads to a more distributed uniform sampling, so that the converged sampling distributions for the selected reactions become smoother.



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



`sampleCbModel(model, sampleFile, samplerName, options, modelSampling)` - Samples the solution-space of a constraint-based model

#### Usage

```
[modelSampling, samples] = sampleCbModel(model, sampleFile, samplerName, options, modelSampling)
```

#### Input

- `model` - COBRA model structure with fields \* `.S` - Stoichiometric matrix \* `.b` - Right hand side vector \* `.lb` - Lower bounds \* `.ub` - Upper bounds

#### Optional inputs

- `sampleFile` - File names for sampling output files (only implemented for ACHR)
- `samplerName` - {('CHRR'), 'ACHR'} Name of the sampler to be used to sample the solution.
- `options` - Options for sampling and pre/postprocessing (default values in parenthesis).
  - ✓ `.nStepsPerPoint` - Number of sampler steps per point saved (200)
  - ✓ `.nPoin`tsReturned - Number of points loaded for analysis (2000)
  - ✓ `.nWarmupPoints` - Number of warmup points (5000). ACHR only.
  - ✓ `.nFiles` - Number of output files (10). ACHR only.
  - ✓ `.nPoin`tsPerFile - Number of points per file (1000). ACHR only.
  - ✓ `.nFilesSkipped` - Number of output files skipped when loading points to avoid potentially biased initial samples (2) loops (true). ACHR only.
  - ✓ `.maxTime` - Maximum time limit (Default = 36000 s). ACHR only.
  - ✓ `.toRound` - Option to round the model before sampling (true). CHRR only.
  - ✓ `.lambda` - the bias vector for exponential sampling. CHRR\_EXP only.
- `modelSampling` - From a previous round of sampling the same model. Input to avoid repeated preprocessing.

#### Outputs

- `modelSampling` - Cleaned up model used in sampling
- `samples` -  $n \times \text{numSamples}$  matrix of flux vectors

## "sampleCbModel": Sampling Function

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



# sampleCbModel Example

```
% Sampling_sampleCbModel_Example_gpSampler.m

clear;

% Input the E.coli core model and set constraints

model = readCbModel('ecoli_core_model.mat');

model = changeRxnBounds(model, 'EX_glc(e)', -10, '1');

biomassRxnAbbr = 'Biomass_Ecoli_core_N(w/GAM)-Nmet2';

ibm = find(ismember(model.rxn, biomassRxnAbbr));

model.lb(ibm)=0.05;

model.c(:)=0; % Remove biomass as objective function

model_aerobic = changeRxnBounds(model, 'EX_o2(e)', -20, '1');

model_anaerobic = changeRxnBounds(model_aerobic, 'EX_o2(e)', 0, '1');

options.toRound = 1; % sampleCbModel options

options.nFiles = 50;

options.nPointsReturned = 5000;

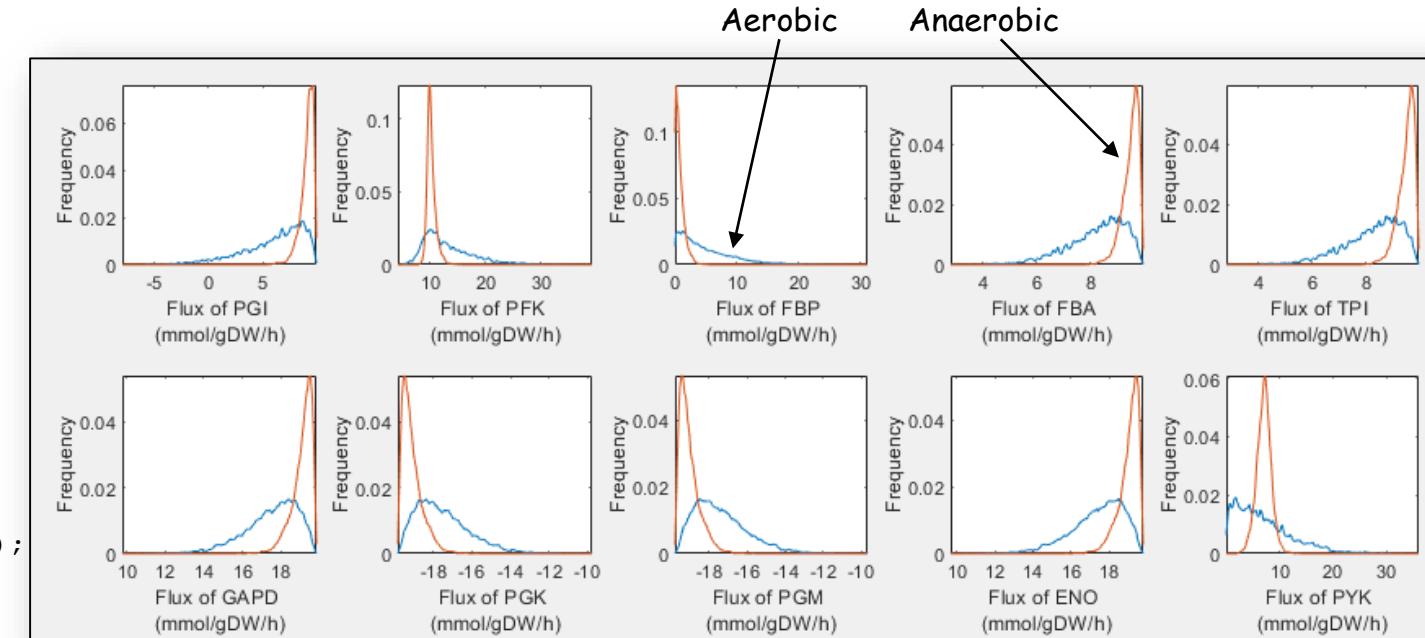
[aerobic_modelSampling,aerobic_samples] = sampleCbModel(model_aerobic,[],[],options);

[anaerobic_modelSampling,anaerobic_samples] = sampleCbModel(model_anaerobic,[],[],options);

% Visualize sampling results for a set of reactions.

rxnList = {'PGI', 'PFK', 'FBP', 'FBA', 'TPI', 'GAPD', 'PGK', 'PGM', 'ENO', 'PYK'};

plotSampleHist(rxnList, {aerobic_samples, anaerobic_samples}, {model_aerobic, model_anaerobic}, [], [5,2]);
```







# 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
model = readCbModel('ecoli_core_model.mat');

% Set carbon source and oxygen uptake rates for wild type model
model = changeRxnBounds(model, 'EX_glc(e)', -5, '1');
model = changeRxnBounds(model, 'EX_o2(e)', -20, '1');
model = changeObjective(model, 'Biomass_Ecoli_core_N(w/GAM)-Nmet2');
FBAsolution = optimizeCbModel(model, 'max', 0, 0);
model_WT = model;

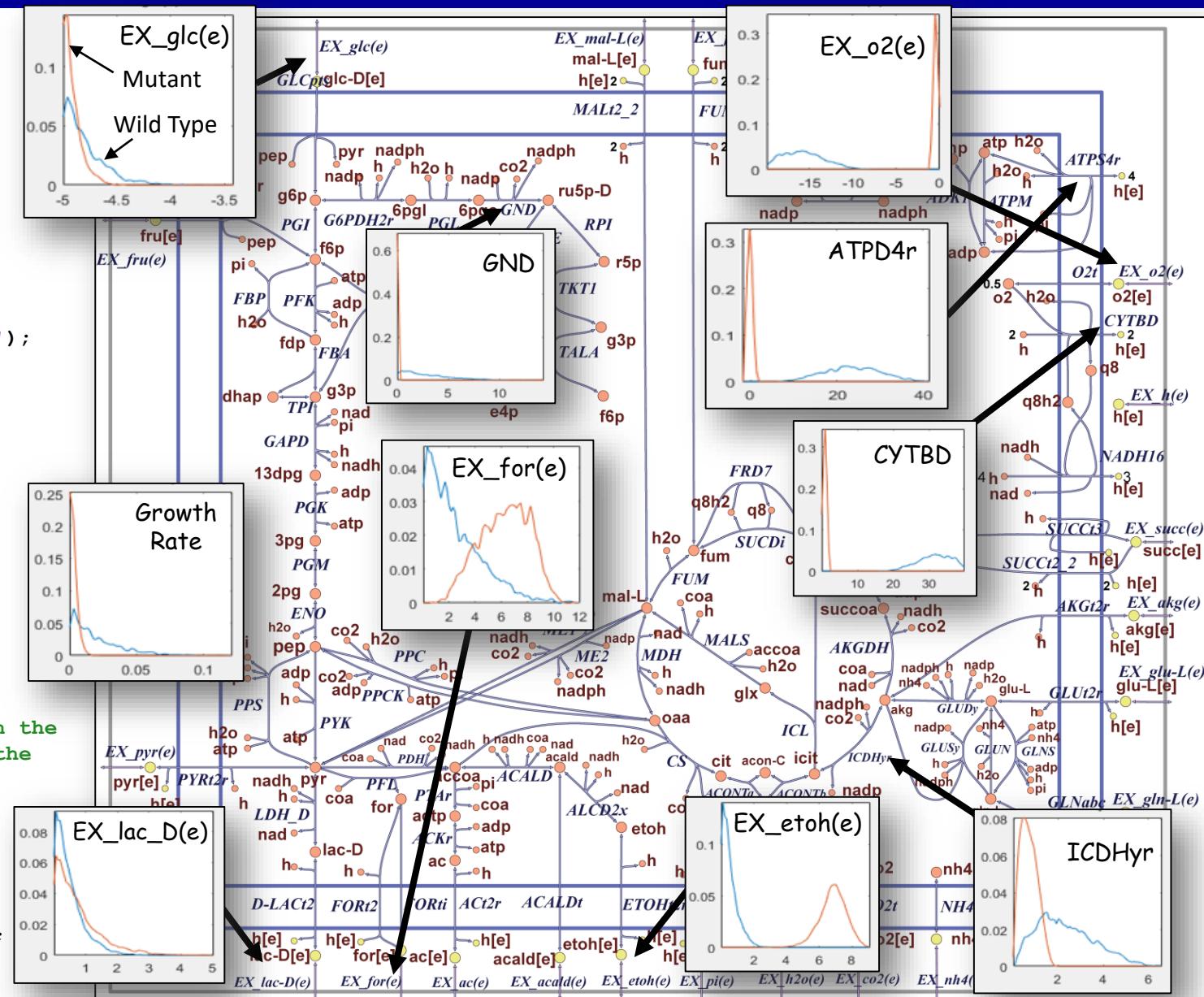
% Knockout reactions for mutant model
model = changeRxnBounds(model, 'NADH16', 0, 'b');
model = changeRxnBounds(model, 'PTAR', 0, 'b');
model = changeRxnBounds(model, 'TKT2', 0, 'b');
Mutantsolution = optimizeCbModel(model, 'max', 0, 0);
model_Mutant = model;

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_etoeh(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]);
```

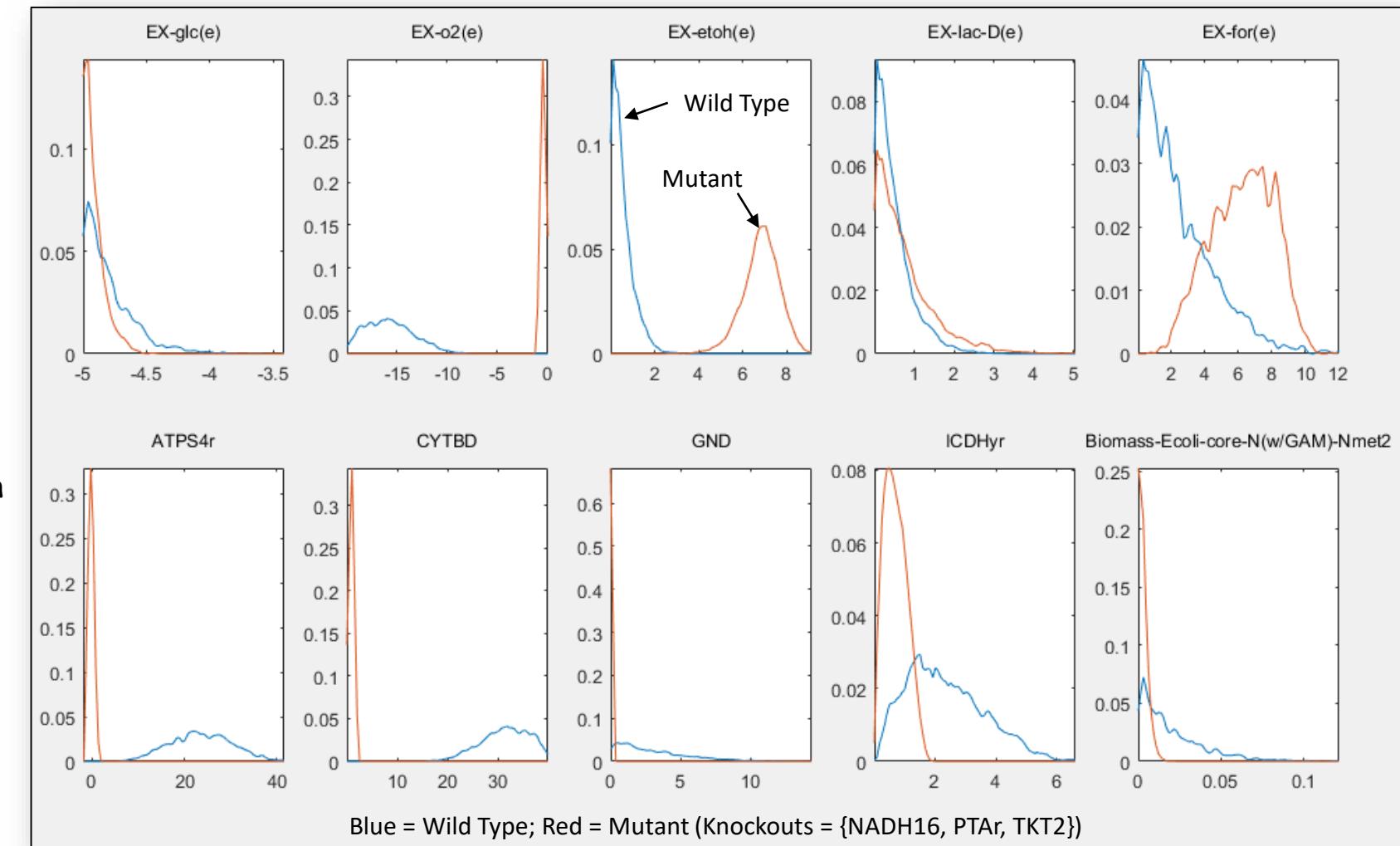
EthanolProduction\_Sampling.m





# 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
- ATPS4r 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



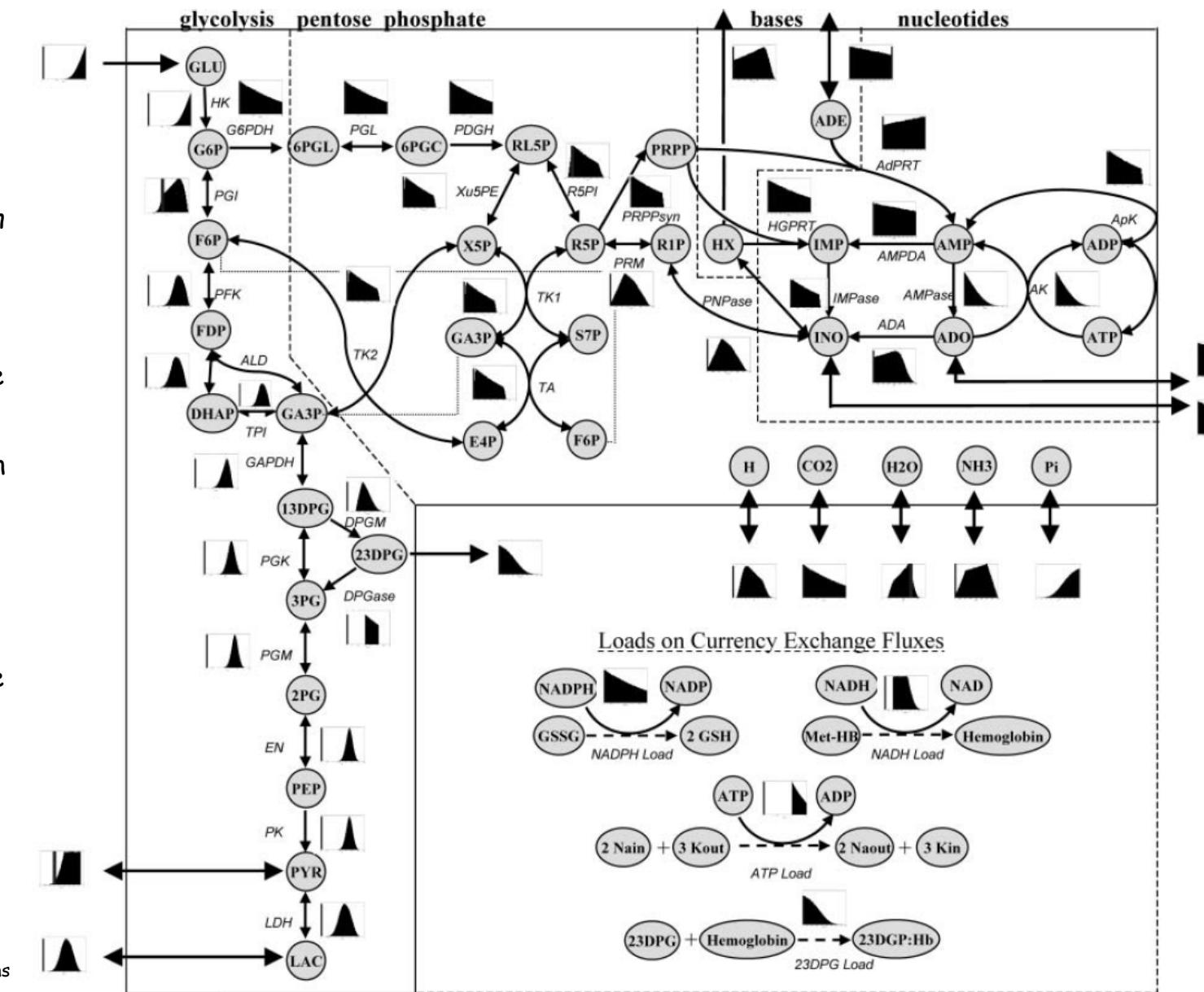
EthanolProduction\_Sampling.m



# 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.

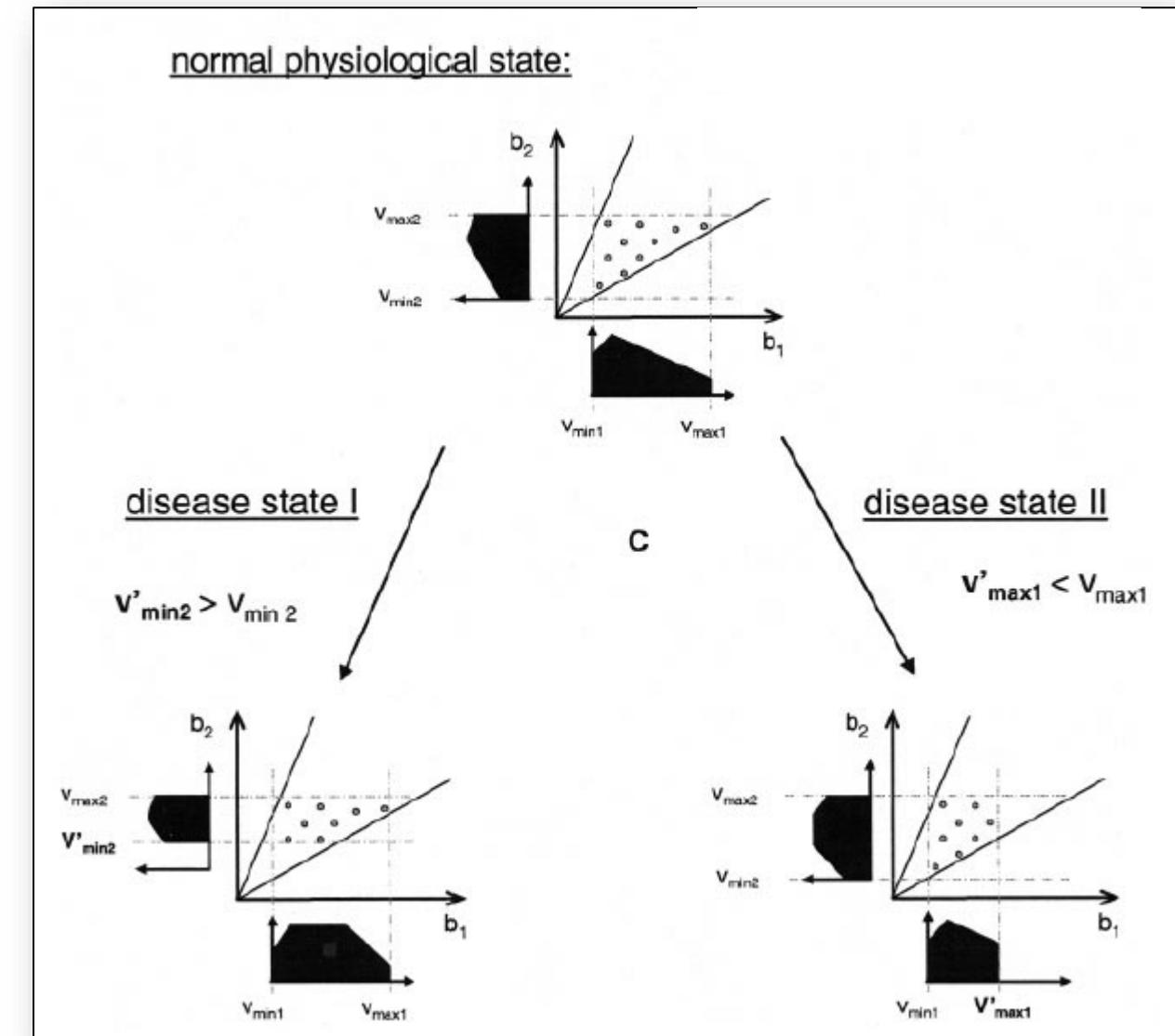
Price, N. D., J. Schellenberger, et al. (2004). "Uniform sampling of steady-state flux spaces: means to design experiments and to interpret enzymopathies." *Biophysical journal* 87(4): 2172-2186.





# Metabolic Network States Under Normal And Disease Conditions

By applying constraints ( $V_{min}$ ,  $V_{max}$ ) 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.



Thiele, I., N. D. Price, et al. (2005). "Candidate metabolic network states in human mitochondria. Impact of diabetes, ischemia, and diet." *The Journal of biological chemistry* 280(12): 11683-11695.



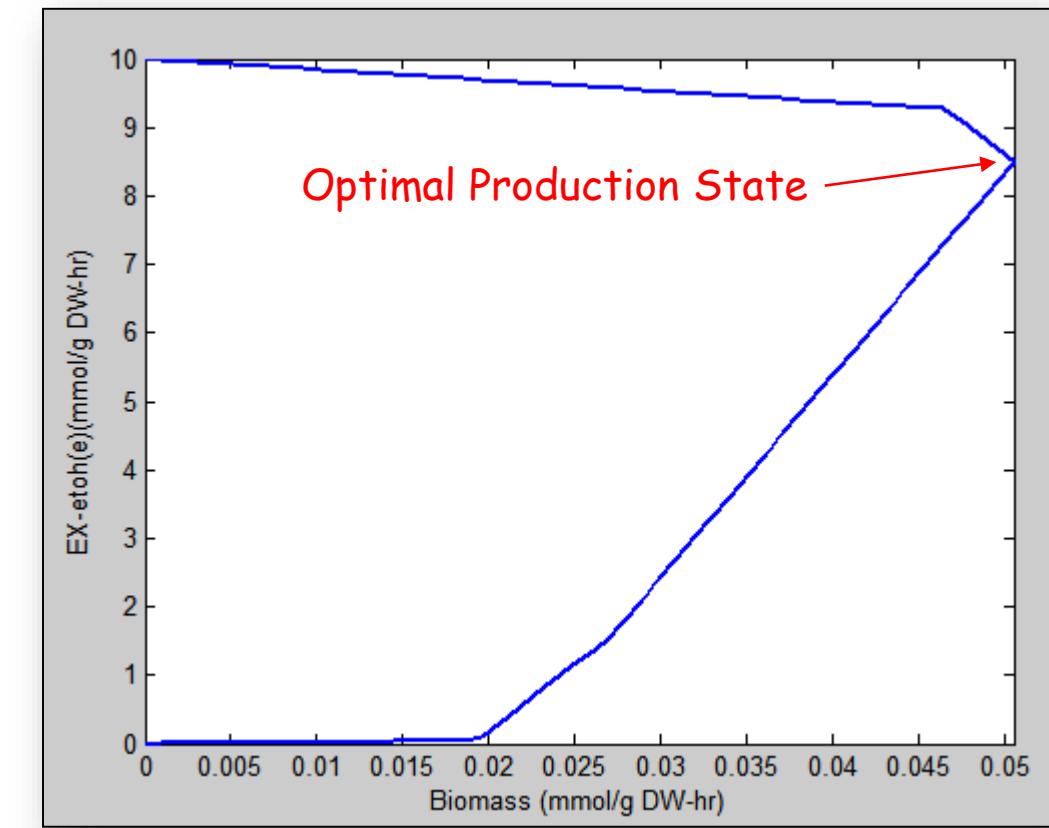
# Lesson Outline

- 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.



Production Envelope

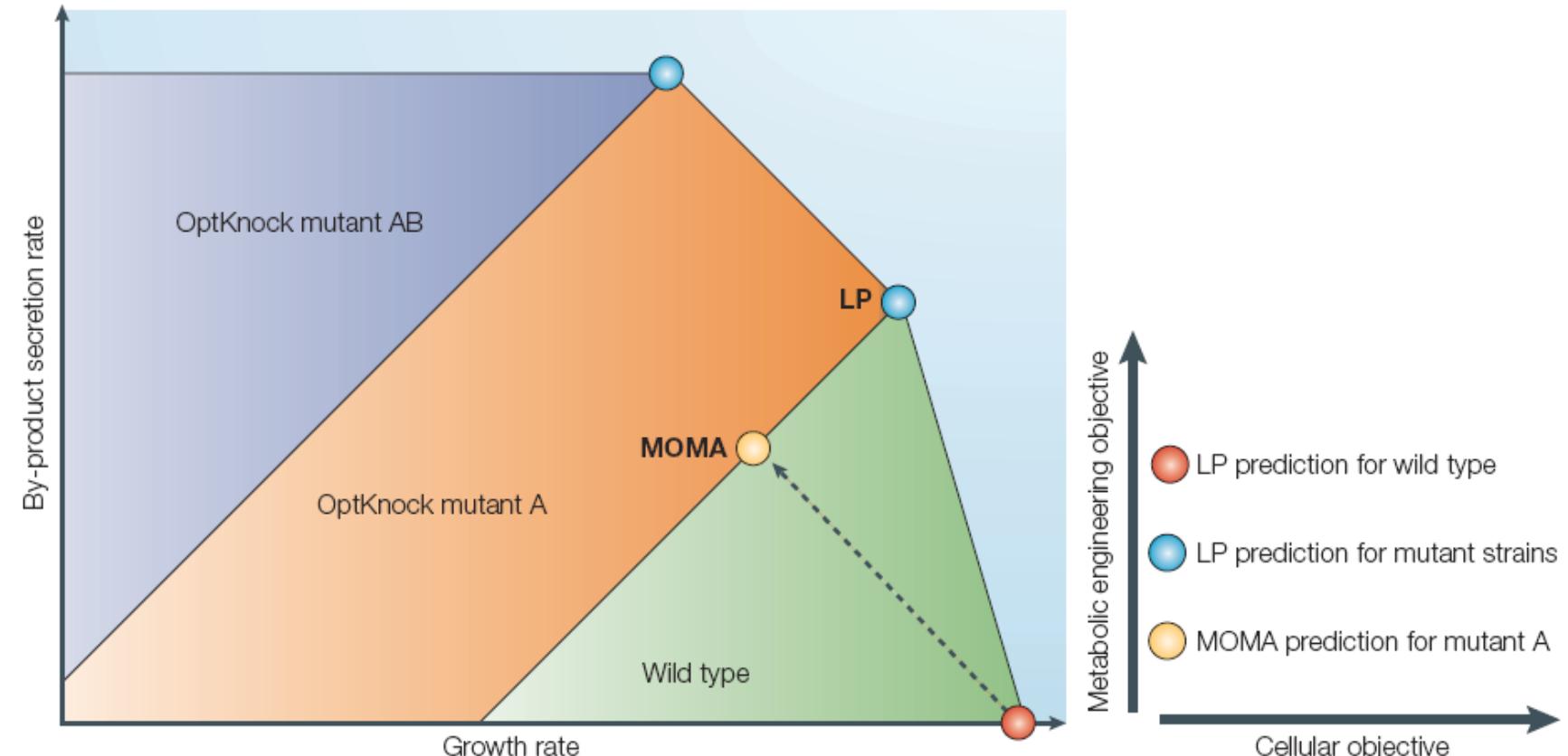
Segre, D., D. Vitkup, et al. (2002). "Analysis of optimality in natural and perturbed metabolic networks." Proceedings of the National Academy of Sciences of the United States of America 99(23): 15112-15117.



# Method of Minimization of Metabolic Adjustment (MOMA)

Segre, D., D. Vitkup, et al. (2002). "Analysis of optimality in natural and perturbed metabolic networks."

- 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



Segre, D., D. Vitkup, et al. (2002). "Analysis of optimality in natural and perturbed metabolic networks." Proceedings of the National Academy of Sciences of the United States of America 99(23): 15112-15117.

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.



```
% EthanolProduction_GDLS_Mutants_MOMA.m  
clear;  
  
% Load the E.coli core model  
model = readCbModel('ecoli_core_model.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)
```

# MOMA Example



# Flux Differences: Wild Type, GDLS, & MOMA

Reaction	WT	GDLS	MOMA
ACALD	2.27E-13	-7.38652	-5.38586
ACONTa	3.44346	1.87357	0.010514
ACONTb	3.44346	1.87357	0.010514
AKGDH	2.99507	1.81911	0
ALCD2x	0	-7.38652	-5.38586
ATPM	8.39	8.39	8.39
ATPS4r	24.8712	0.094332	1.27156
Biomass	0.415598	0.050473	0.009745
CO2t	-12.314	-3.6298	-5.36298
CS	3.44346	1.87357	0.010514
CYTBD	23.6671	1.81911	0
D_LACT2	0	0	-2.39504
ENO	7.58501	9.76307	7.88854
ETOHt2r	0	-7.38652	-5.38586
EX_co2(e)	12.314	3.6298	5.36298
EX_etoh(e)	0	7.38652	5.38586
EX_for(e)	0	9.44925	0.068288
EX_glc(e)	-5	-5	-3.96714
EX_h2o(e)	15.3414	-3.38905	0.048112
EX_h(e)	8.33689	10.4617	2.65882

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

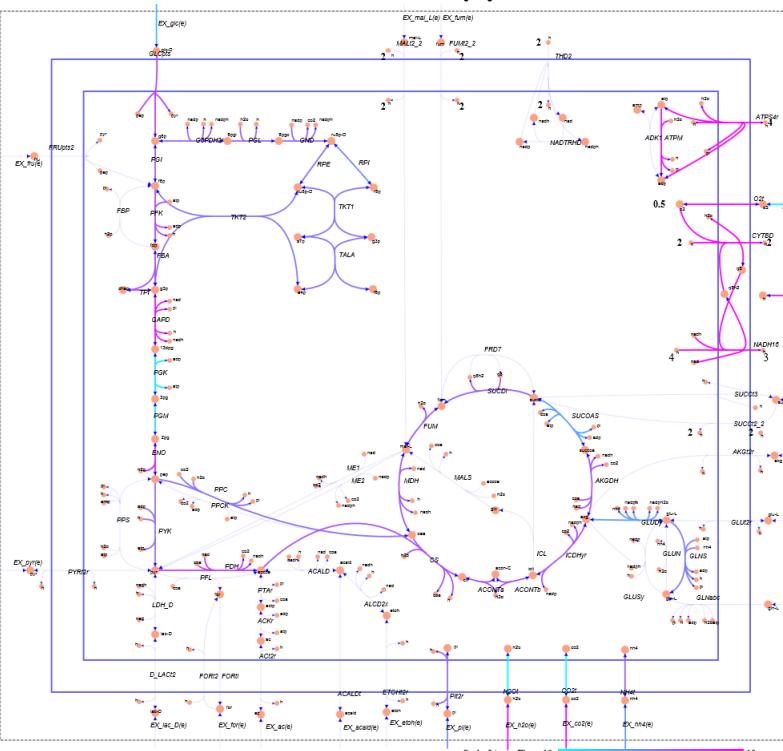
Reaction	WT	GDLS	MOMA
NADTRHD	0	1.1172	0
NH4t	2.26617	0.275221	0.05314
O2t	11.8336	0.909557	0
PDH	5.00103	0	5.36461
PFK	3.89814	4.92254	3.95219
PFL	0	9.44925	0.068288
PGI	2.8494	4.9079	3.94936
PGK	-8.20675	-9.83858	-7.90311
PGL	2.06541	0.081752	0.015785
PGM	-7.58501	-9.76307	-7.88854
Plt2r	1.52886	0.185676	0.035851
PPC	1.19094	0.144636	0.163454
PYK	1.17834	4.59223	3.75288
RPE	1.07821	0.018221	0.003518
RPI	-0.9872	-0.06353	-0.01227
SUCDi	2.99507	1.81911	1.49752
SUCOAS	-2.99507	-1.81911	0
TALA	0.614118	0.018221	0.003518
TKT1	0.614118	0.018221	0.003518
TKT2	0.464088	0	0
TPI	3.89814	4.92254	3.95219

EthanolProduction\_GDLS\_Mutants\_MOMA.m



# Flux Maps: Wild Type, GDLS, & MOMA

# Wild Type



## EthanolProduction\_WT.m

## Formate

5500/6500

EthanolProduction\_GDLS\_Mutant.m

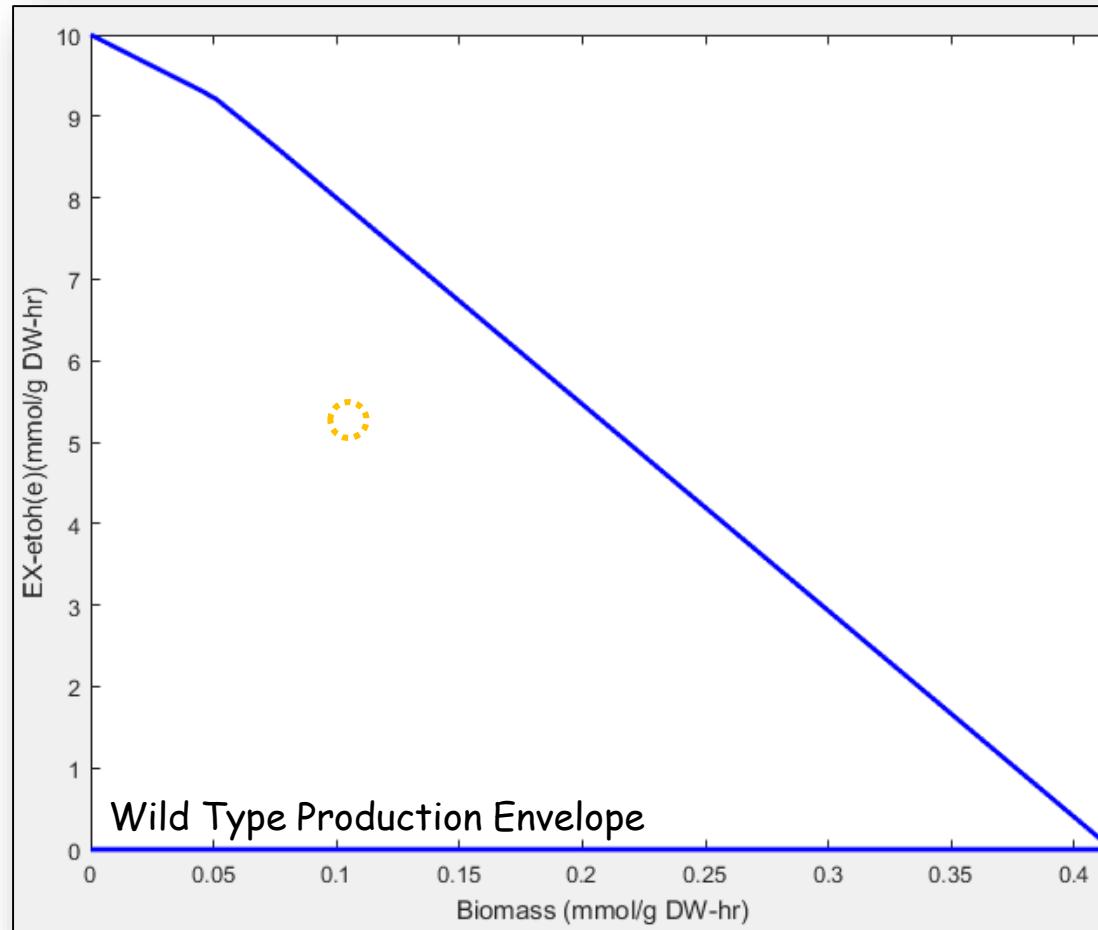
## Lactate

EthanolProduction\_GDLS\_Mutants\_MOMA.m

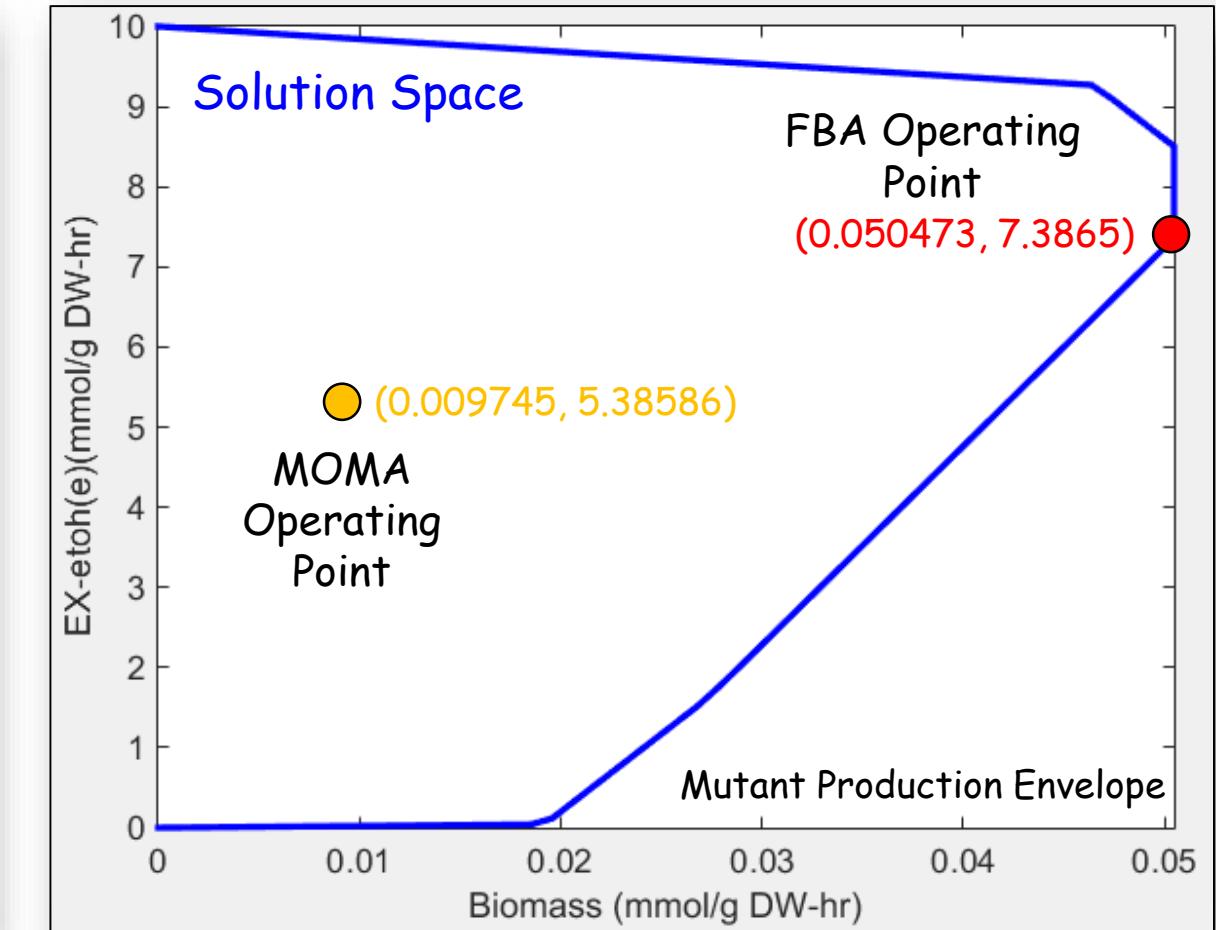
Lesson: Randomized Sampling & Adaptive Laboratory Evolution



# Production Envelope - Ethanol Production Mutant



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?



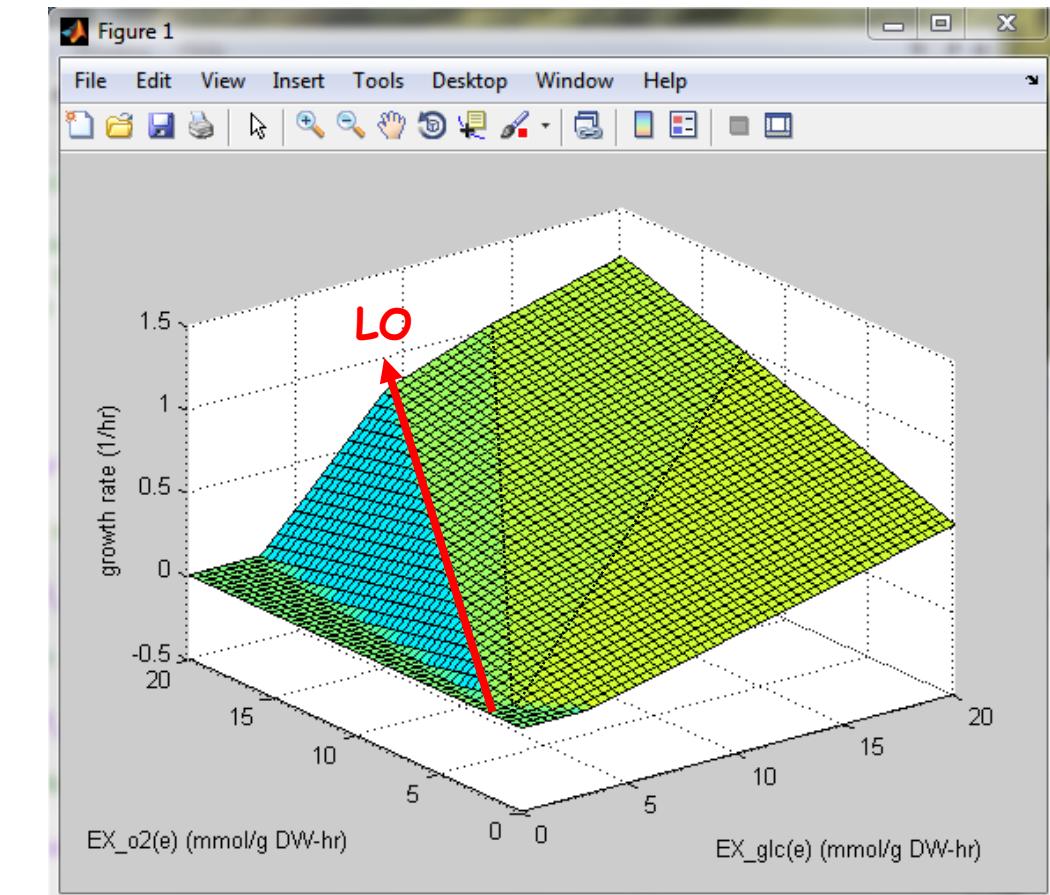
# Lesson Outline

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



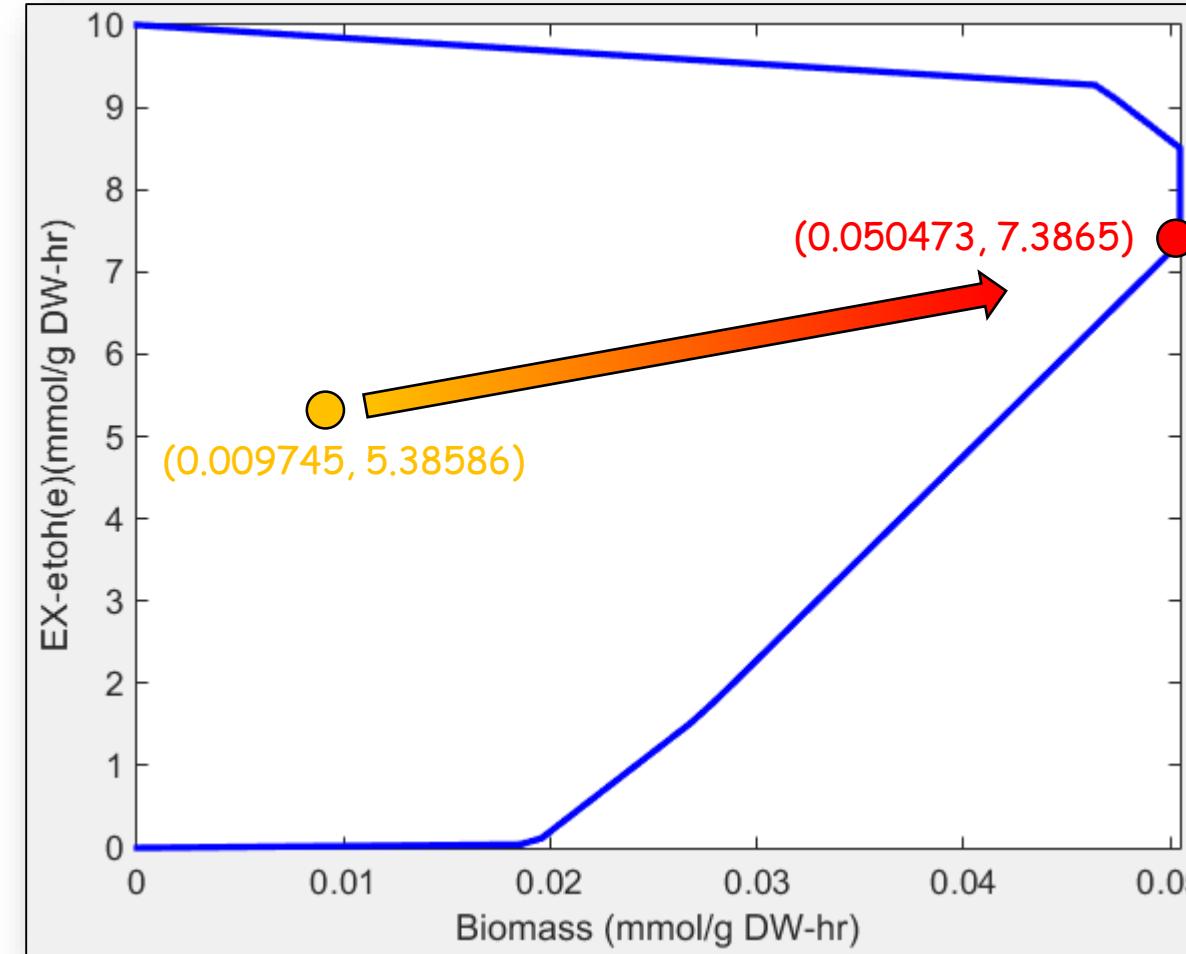
# Line of Optimality

- 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.



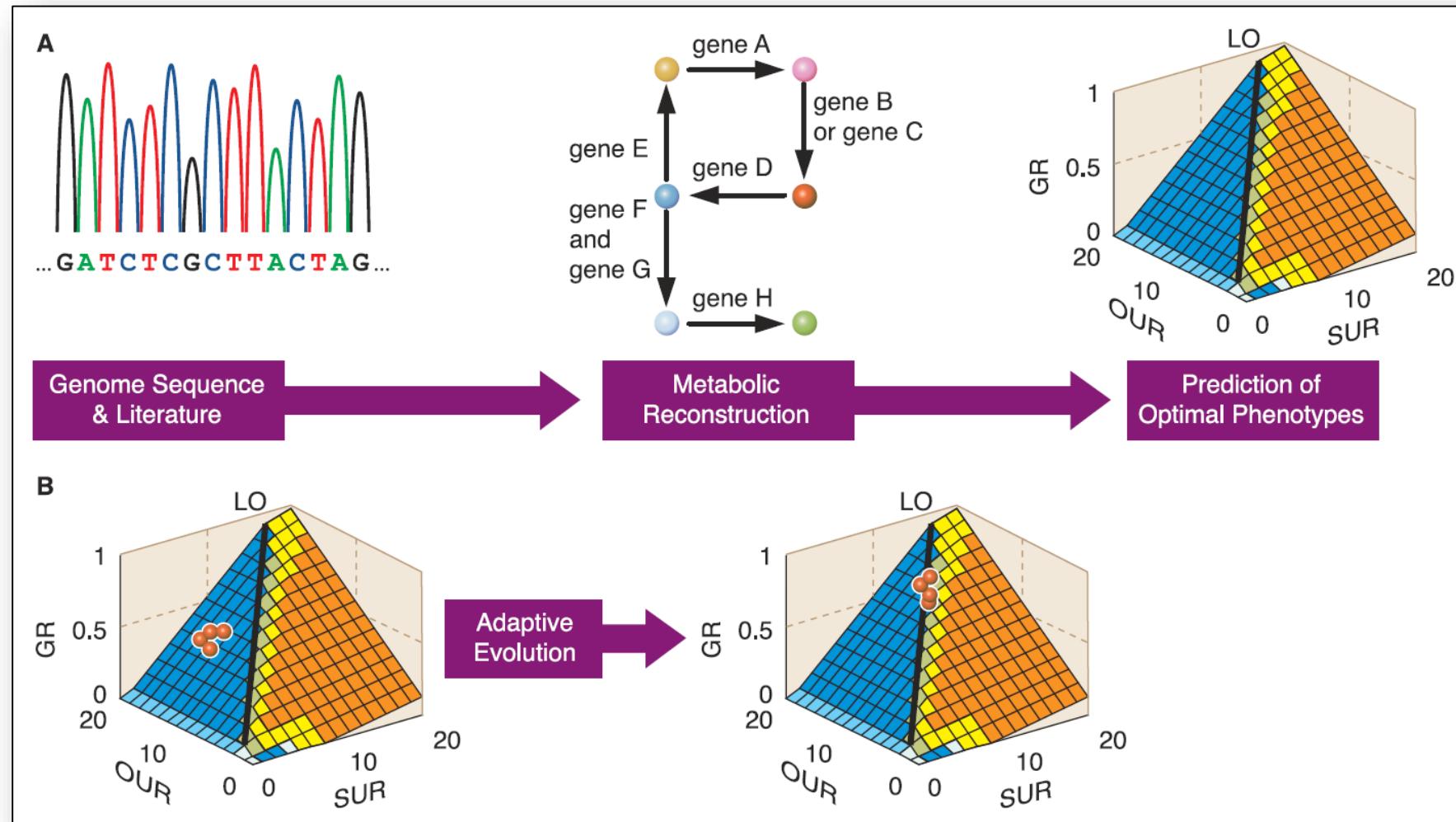


# Desired Cell Evolution





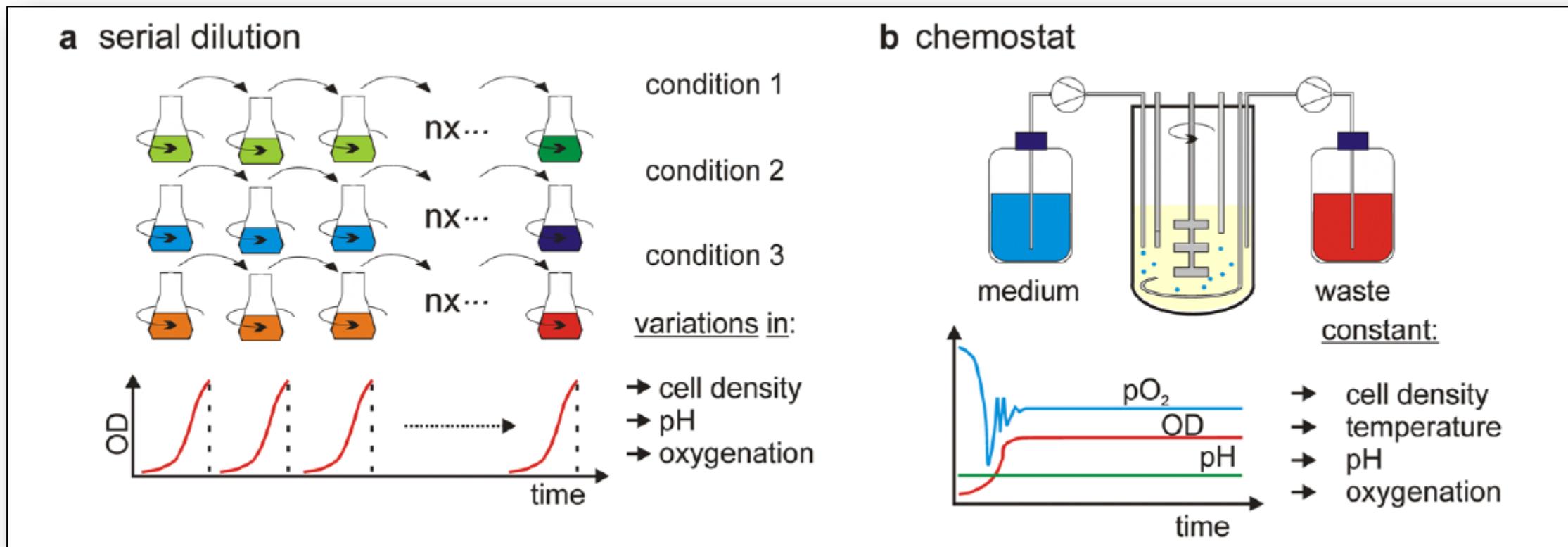
# Adaptive Laboratory Evolution



B. Palsson (2010). "Adaptive Laboratory Evolution." *Microbe*, 6(2):69-74



# Adaptive Laboratory Evolution Methods

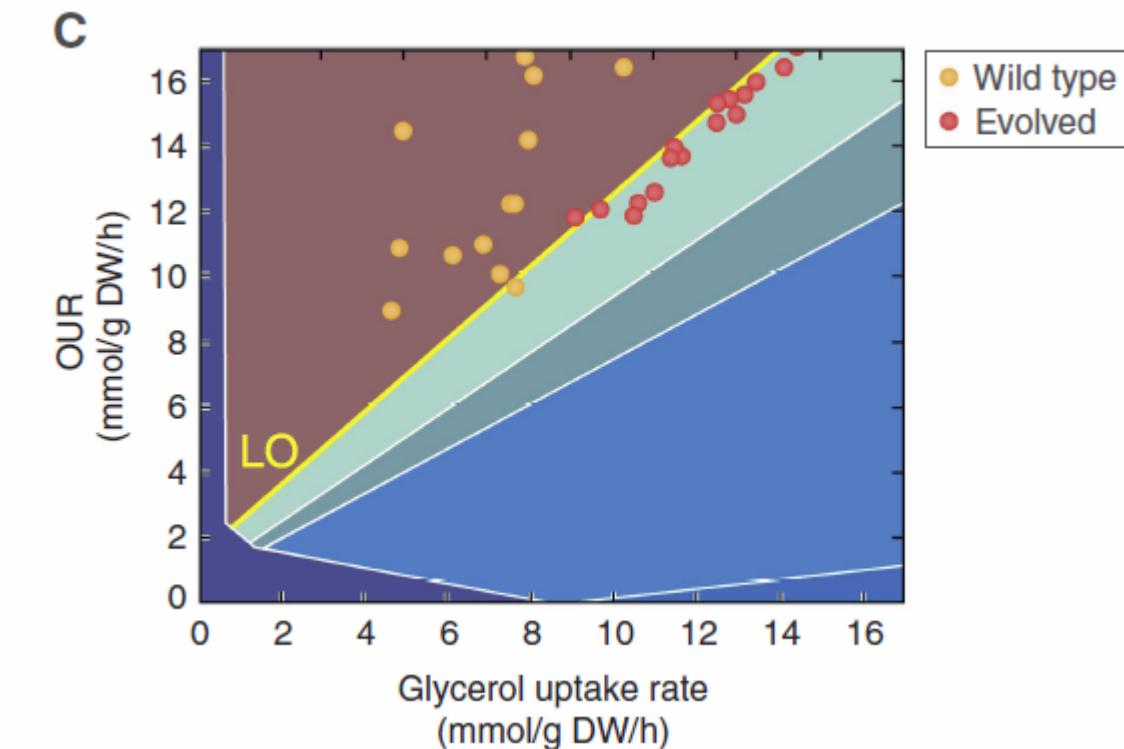


Dragosits, M. and D. Mattanovich (2013). "Adaptive laboratory evolution -- principles and applications for biotechnology." *Microbial cell factories* 12: 64.



# Adaptive Laboratory Evolution

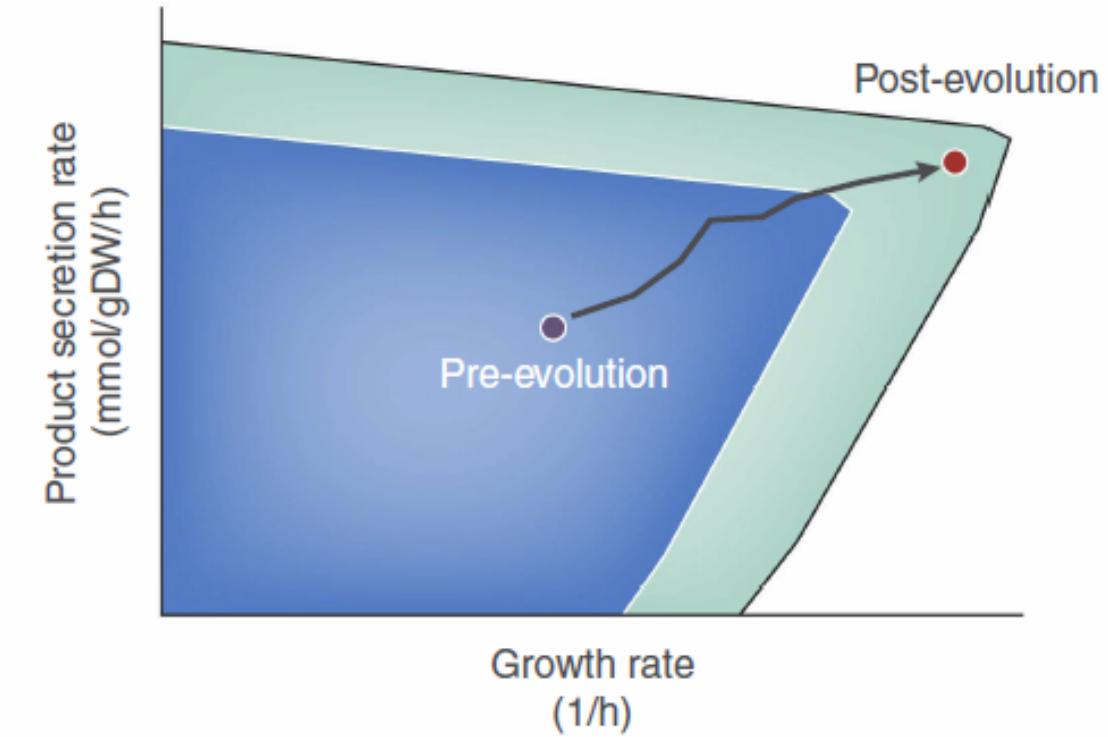
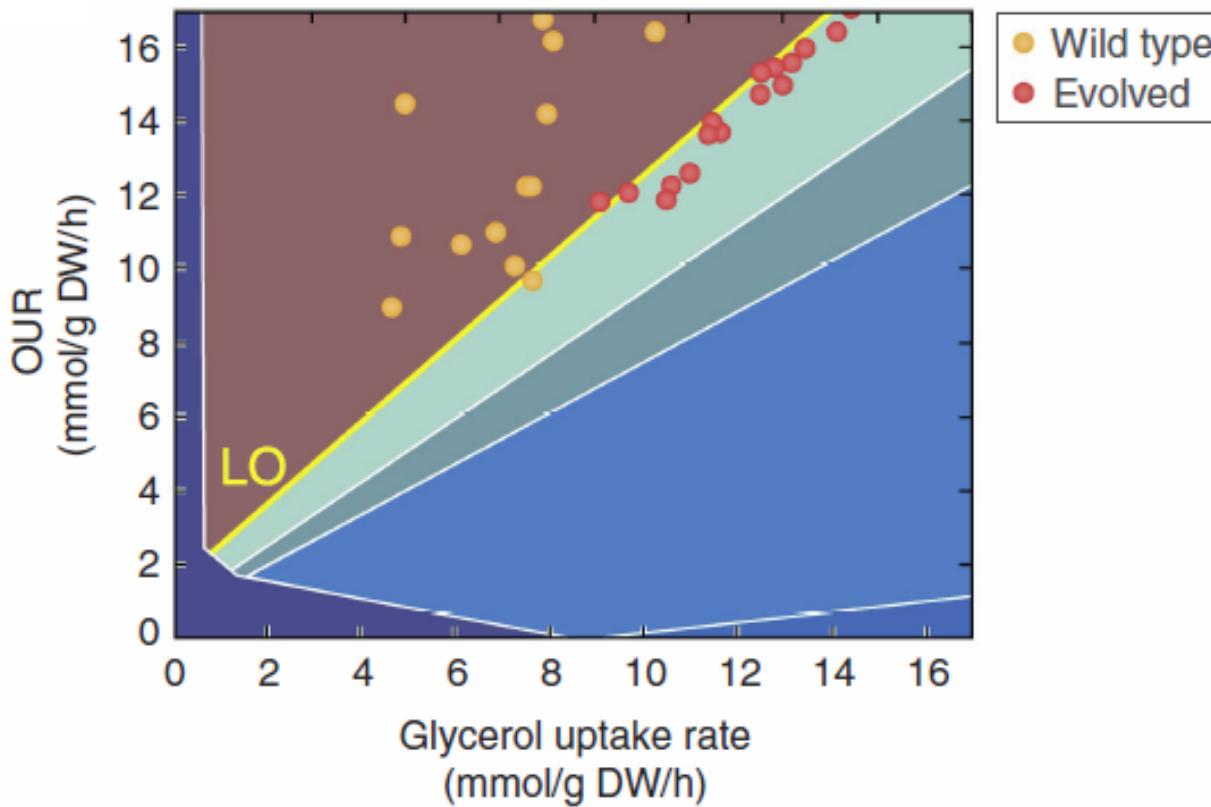
- Cells will modify their genotype to optimize their fitness in the given environment.
- FBA results can be presented on phenotypic phase-plane (PPP) plots of model-predicted optimal biomass production (growth rate) versus carbon source uptake rate (SUR) and oxygen uptake rate (OUR).
- The line of optimality (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



Conrad, T. M., N. E. Lewis, et al. (2011). "Microbial laboratory evolution in the era of genome-scale science." *Molecular Systems Biology* 7: 509.



# Adaptive Laboratory Evolution

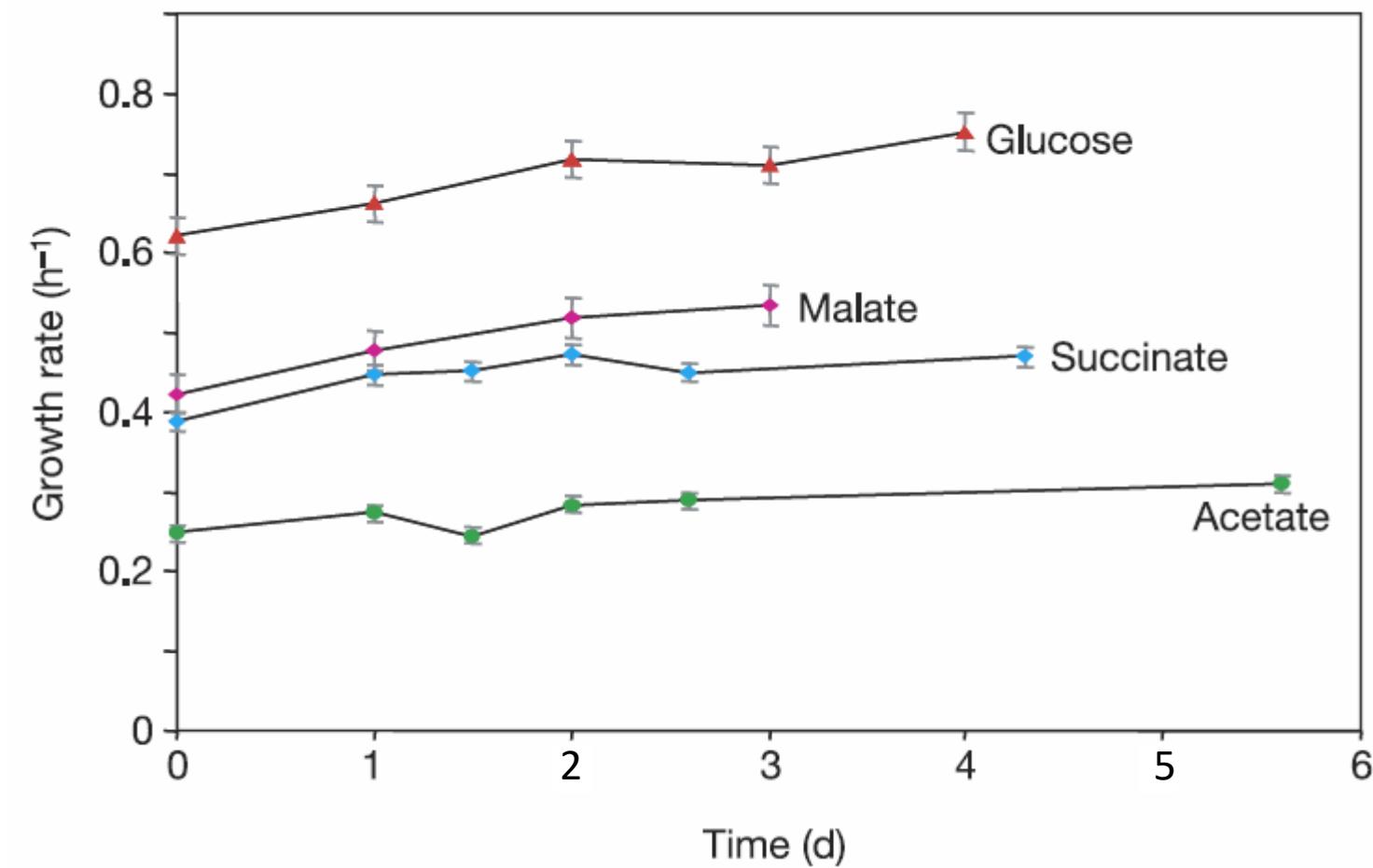


Conrad, T. M., N. E. Lewis, et al. (2011). "Microbial laboratory evolution in the era of genome-scale science." *Molecular Systems Biology* 7: 509.



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

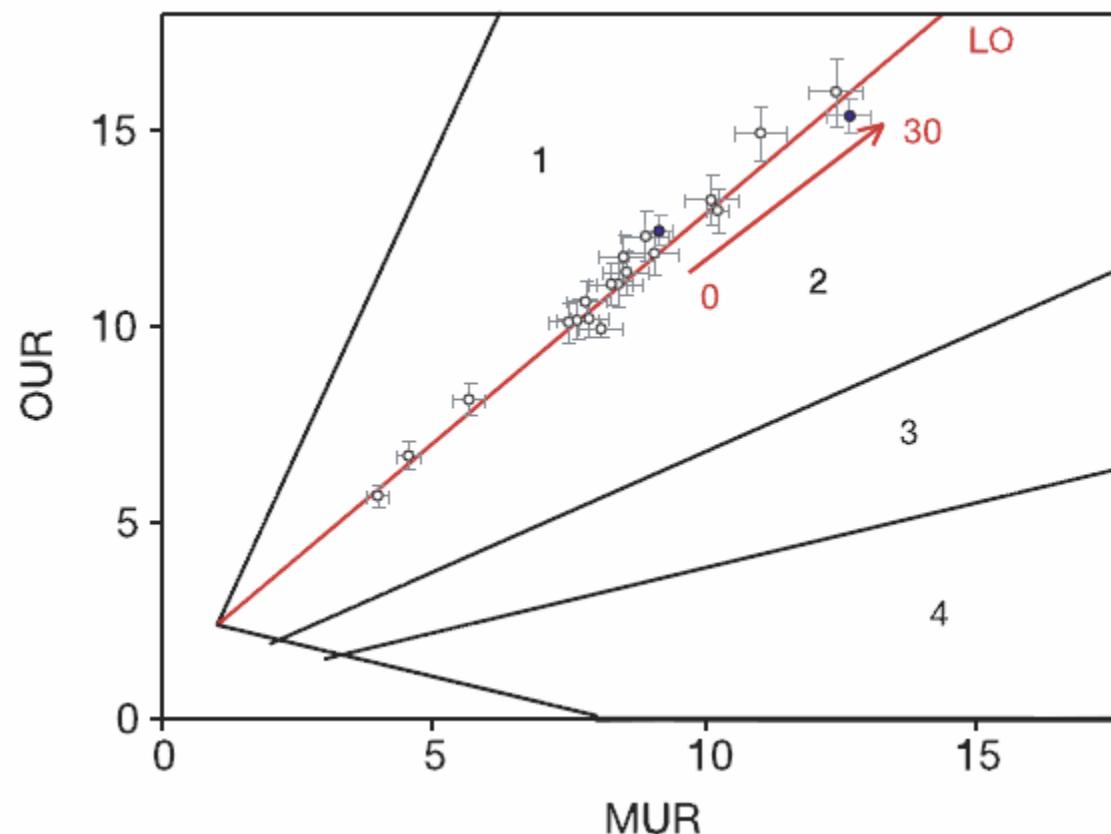
- Growth rate (exponential phase) during adaptive evolution on glucose, malate, succinate and acetate.
- The increases in growth rate over time were as follows:
  - ✓ glucose (18%),
  - ✓ malate (21%),
  - ✓ succinate (17%) and
  - ✓ acetate (20%).
- The number of generations for each adaptive evolution was: glucose (500), malate (500), succinate (1,000) and acetate (700).



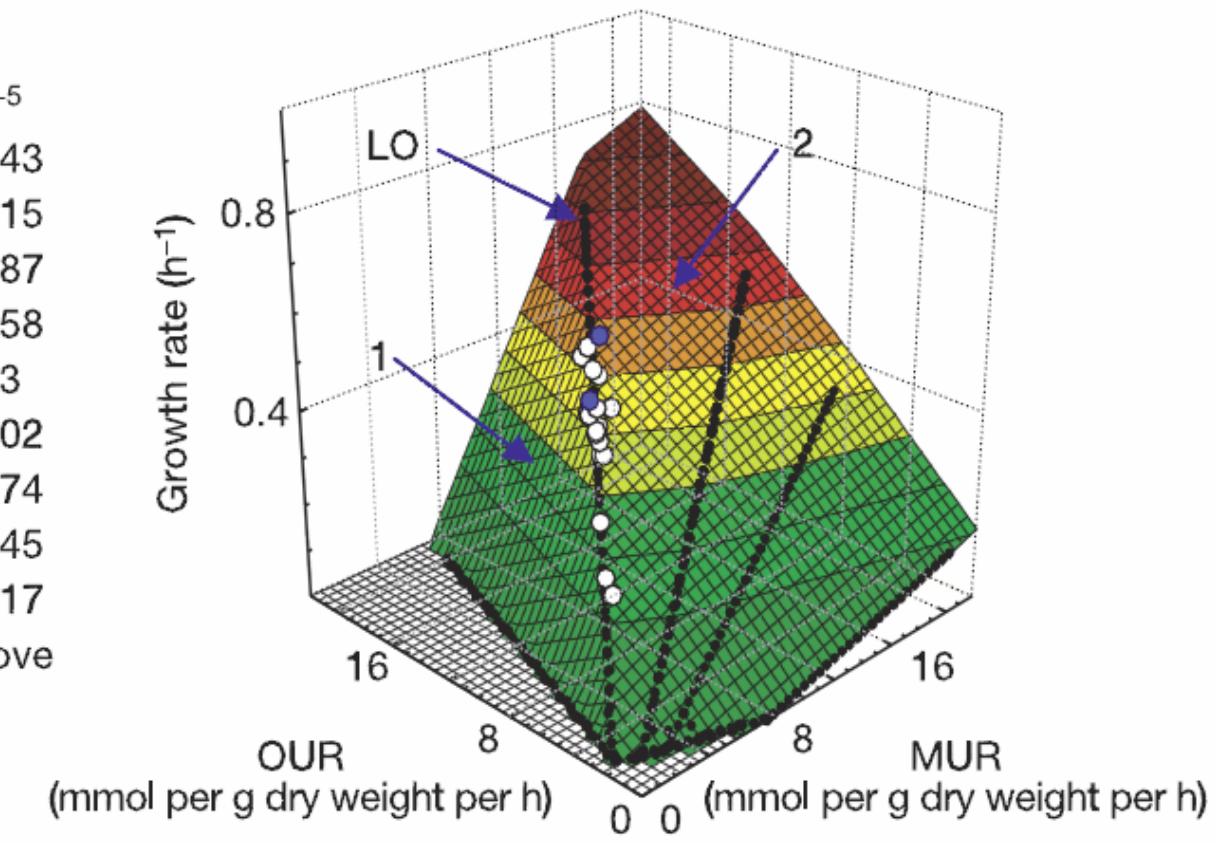
Ibarra, R. U., J. S. Edwards, et al. (2002). "Escherichia coli K-12 undergoes adaptive evolution to achieve *in silico* predicted optimal growth." Nature 420(6912): 186-189.



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



- $10^{-5}$
- 0.143
- 0.215
- 0.287
- 0.358
- 0.43
- 0.502
- 0.574
- 0.645
- 0.717
- above

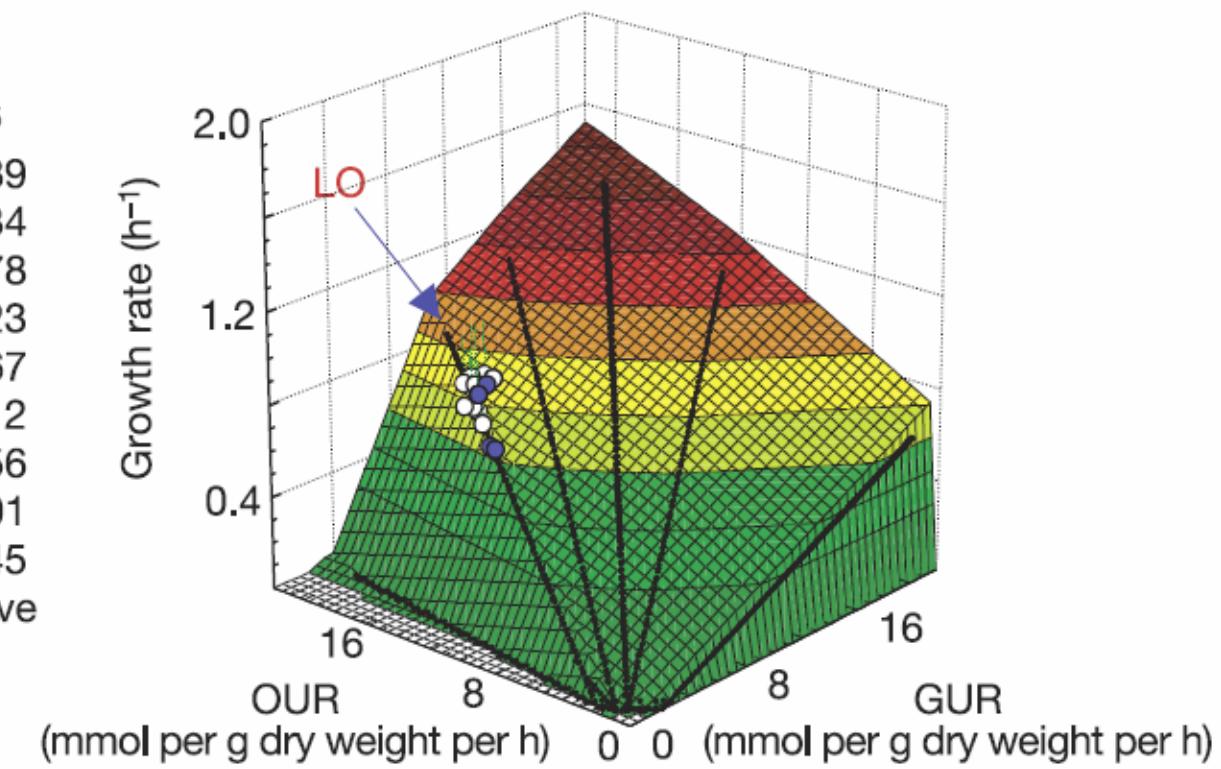
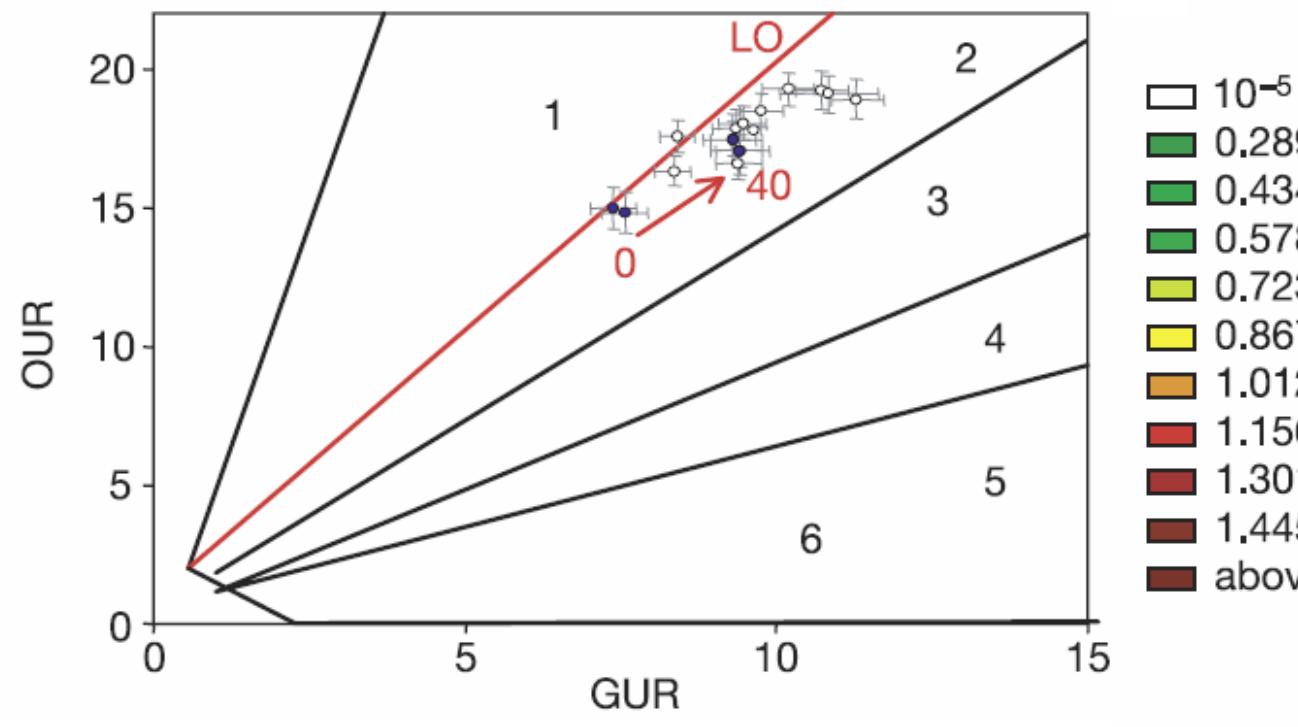


Blue dots are starting and ending points

Ibarra, R. U., J. S. Edwards, et al. (2002). "Escherichia coli K-12 undergoes adaptive evolution to achieve *in silico* predicted optimal growth." Nature 420(6912): 186-189.



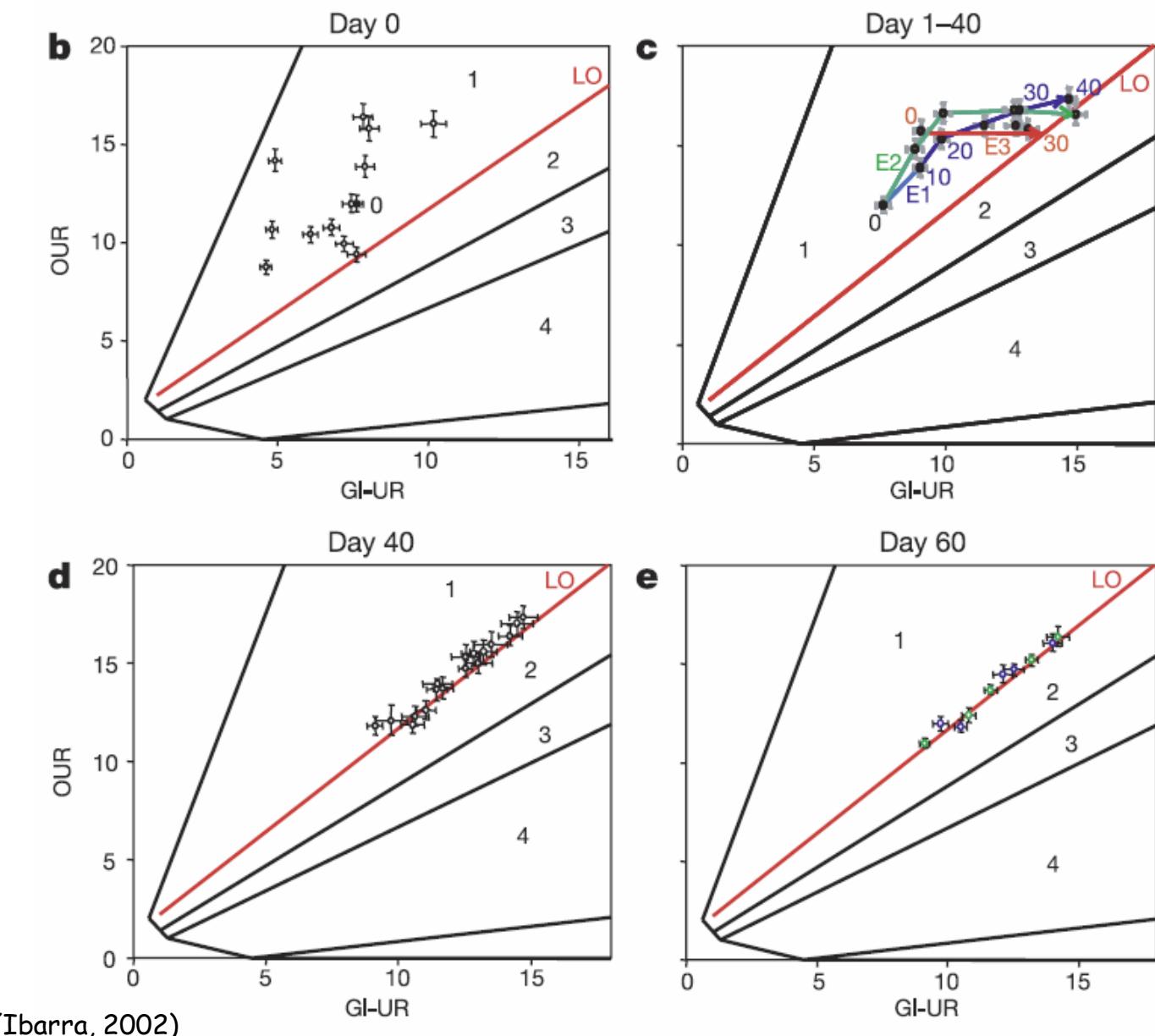
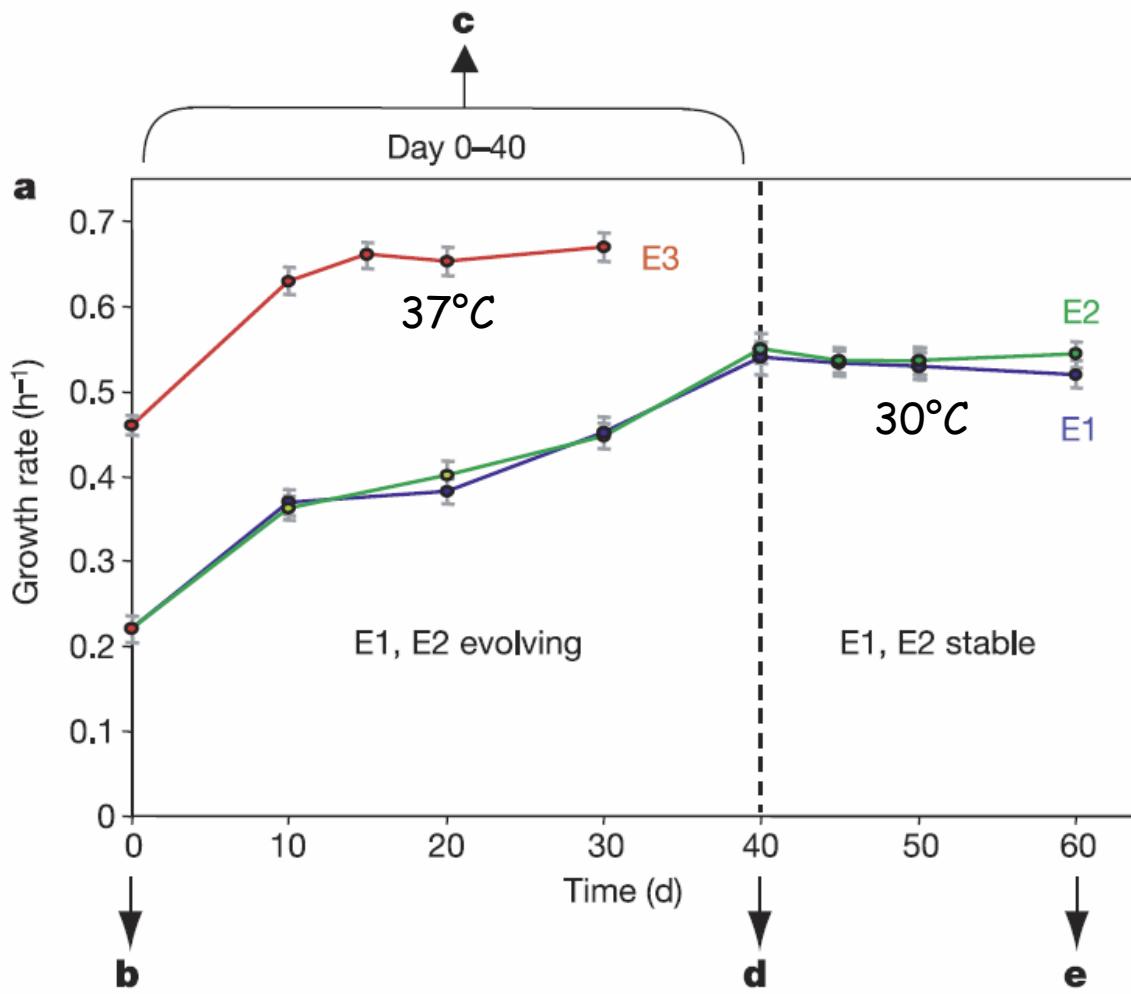
# Growth of *E.coli* K-12 on Glucose



Ibarra, R. U., J. S. Edwards, et al. (2002). "Escherichia coli K-12 undergoes adaptive evolution to achieve *in silico* predicted optimal growth." Nature 420(6912): 186-189.



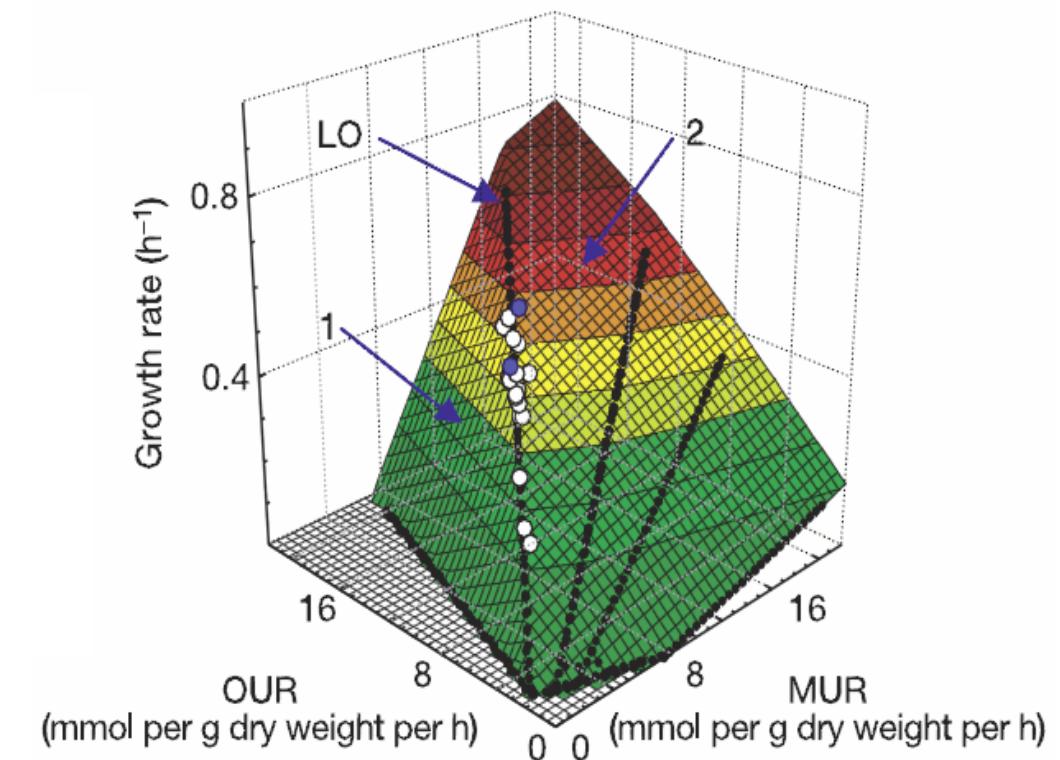
# Growth of *E.coli* K-12 on Glycerol





# 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?



Ibarra, R. U., J. S. Edwards, et al. (2002). "Escherichia coli K-12 undergoes adaptive evolution to achieve *in silico* predicted optimal growth." Nature 420(6912): 186-189.

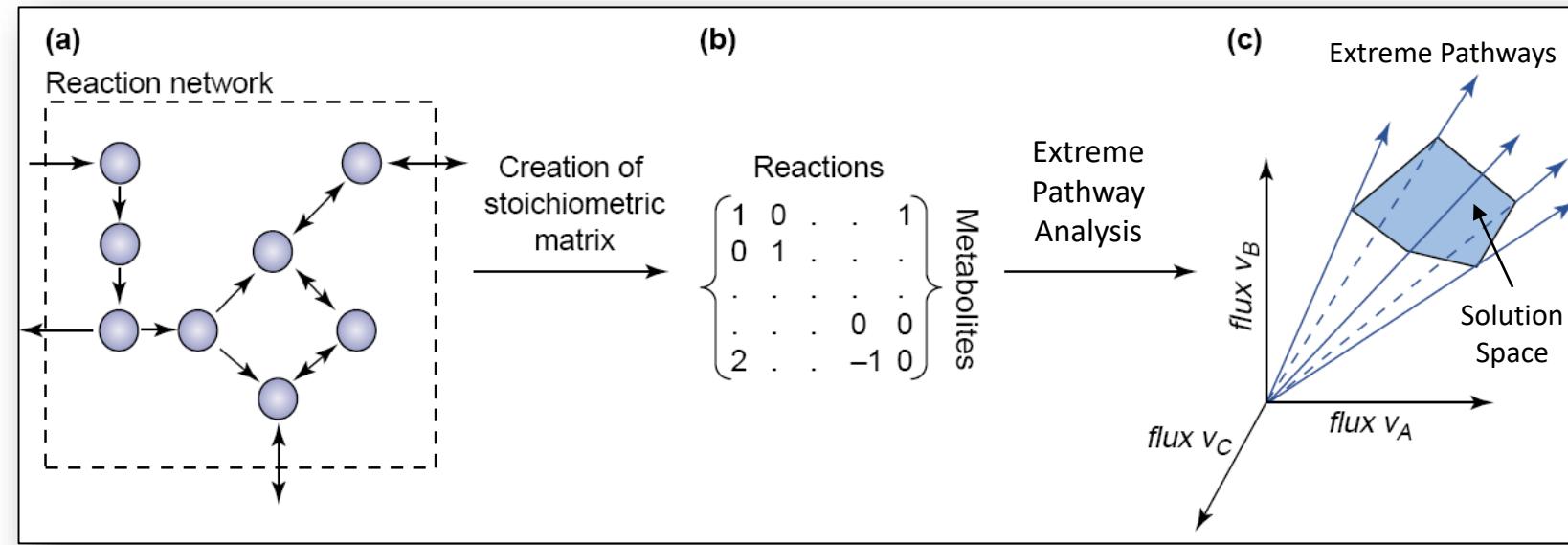


# Lesson Outline

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



# 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.

Papin, J. A., N. D. Price, et al. (2003). "Metabolic pathways in the post-genome era." Trends in biochemical sciences 28(5): 250-258.



# 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)
```



# References

## Sampling Methods

- [David E. Kaufman, Robert L. Smith, \(1998\) Direction Choice for Accelerated Convergence in Hit-and-Run Sampling. Operations Research 46\(1\):84-95.](#)
- [Lovasz, \(1999\)."Hit-and-run mixes fast," Mathematical Programming, 86:443-461.](#)

## Sampling Applications

- [Thiele, I., N. D. Price, et al. \(2005\). "Candidate metabolic network states in human mitochondria. Impact of diabetes, ischemia, and diet." The Journal of biological chemistry 280\(12\): 11683-11695.](#)
- [Price, N. D., J. Schellenberger, et al. \(2004\). "Uniform sampling of steady-state flux spaces: means to design experiments and to interpret enzymopathies." Biophysical journal 87\(4\): 2172-2186.](#)

## Adaptive Evolution

- [Dragosits, M. and D. Mattanovich \(2013\). "Adaptive laboratory evolution -- principles and applications for biotechnology." Microbial cell factories 12: 64.](#)
- [B. Palsson \(2010\). "Adaptive Laboratory Evolution." Microbe, 6\(2\):69-74](#)
- [Conrad, T. M., N. E. Lewis, et al. \(2011\). "Microbial laboratory evolution in the era of genome-scale science." Molecular Systems Biology 7: 509](#)
- [Fong, S. S., A. P. Burgard, et al. \(2005\). "In silico design and adaptive evolution of Escherichia coli for production of lactic acid." Biotechnology and bioengineering 91\(5\): 643-648.](#)
- [Fong, S. S., J. Y. Marciniak, et al. \(2003\). "Description and interpretation of adaptive evolution of Escherichia coli K-12 MG1655 by using a genome-scale in silico metabolic model." Journal of Bacteriology 185\(21\): 6400-6408.](#)
- [Ibarra, R. U., J. S. Edwards, et al. \(2002\). "Escherichia coli K-12 undergoes adaptive evolution to achieve in silico predicted optimal growth." Nature 420\(6912\): 186-189.](#)



# References (2)

## Extreme Pathways

- Price, N. D., J. L. Reed, et al. (2003). "Analysis of metabolic capabilities using singular value decomposition of extreme pathway matrices." *Biophysical journal* 84(2 Pt 1): 794-804.
- Papin, J. A., N. D. Price, et al. (2003). "Metabolic pathways in the post-genome era." *Trends in biochemical sciences* 28(5): 250-258.
- Papin, J. A., N. D. Price, et al. (2002). "The genome-scale metabolic extreme pathway structure in *Haemophilus influenzae* shows significant network redundancy." *J Theor Biol* 215(1): 67-82.
- Papin, J. A., N. D. Price, et al. (2002). "Extreme pathway lengths and reaction participation in genome-scale metabolic networks." *Genome research* 12(12): 1889-1900.
- Price, N. D., J. A. Papin, et al. (2002). "Determination of redundancy and systems properties of the metabolic network of *Helicobacter pylori* using genome-scale extreme pathway analysis." *Genome research* 12(5): 760-769.
- Wiback, S. J. and B. O. Palsson (2002). "Extreme pathway analysis of human red blood cell metabolism." *Biophysical journal* 83(2): 808-818.

## Sampling Low-dimensional Solution Space

- Price, N. D., J. Schellenberger, et al. (2004). "Uniform sampling of steady-state flux spaces: means to design experiments and to interpret enzymopathies." *Biophysical journal* 87(4): 2172-2186.
- Wiback, S. J., I. Famili, et al. (2004). "Monte Carlo sampling can be used to determine the size and shape of the steady-state flux space." *J Theor Biol* 228(4): 437-447.

## Sampling High-dimensional Solution Space

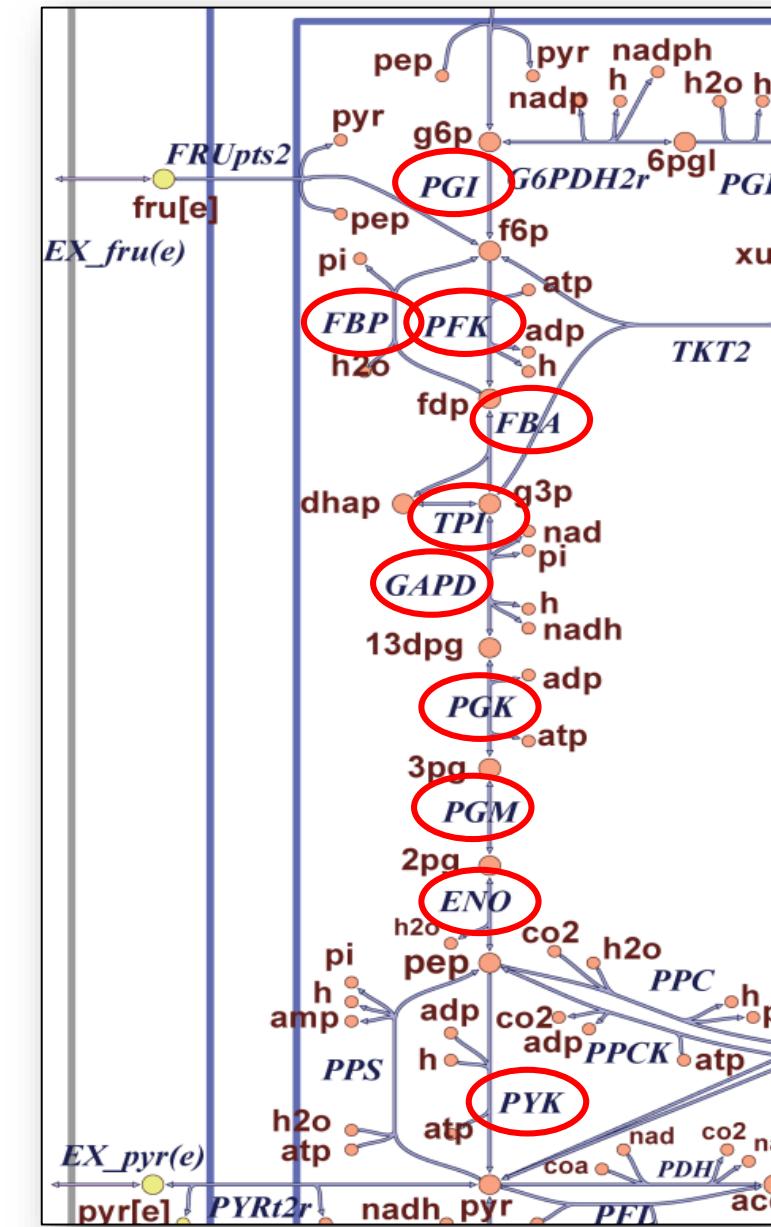
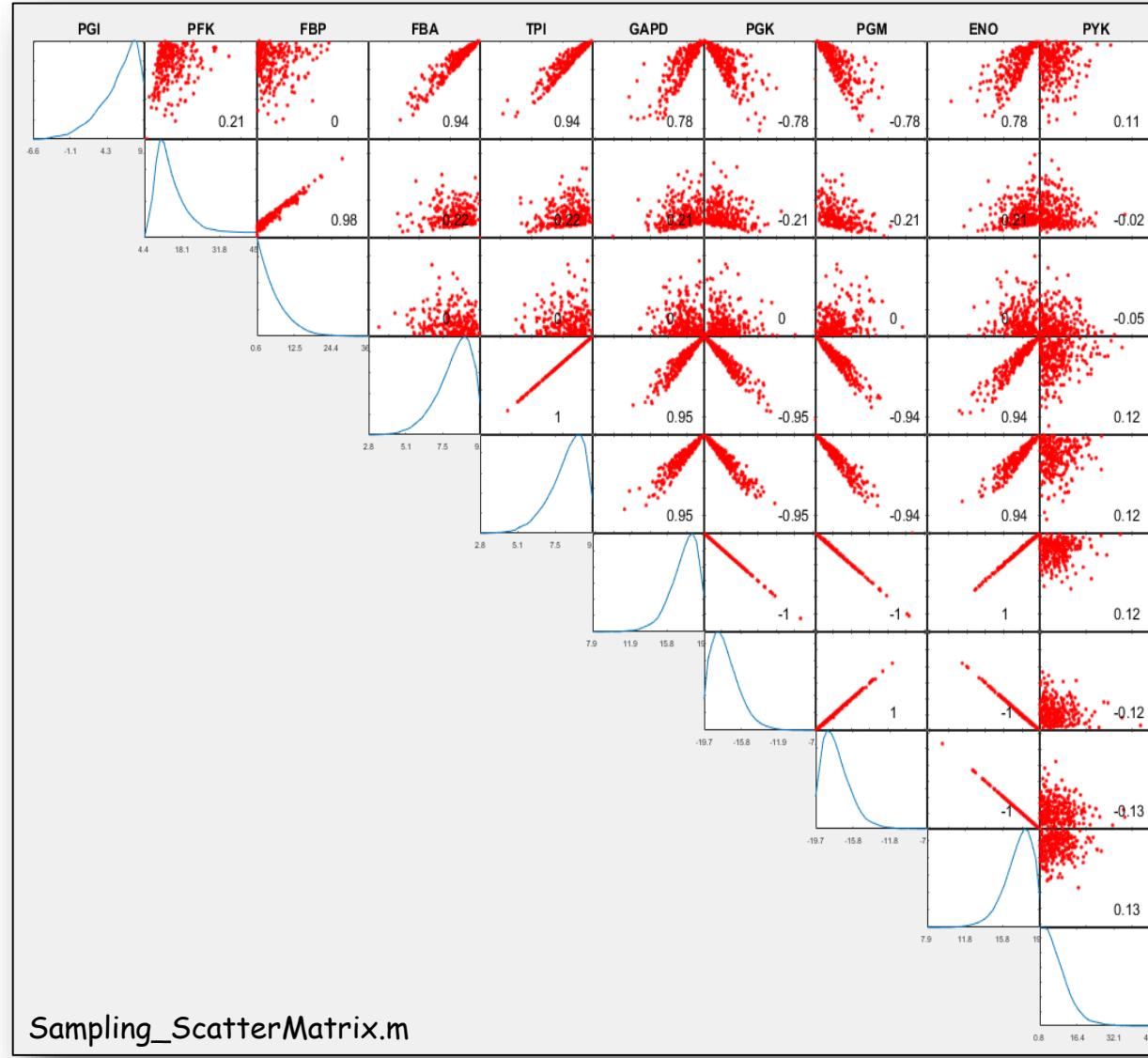
- Almaas, E., B. Kovacs, et al. (2004). "Global organization of metabolic fluxes in the bacterium *Escherichia coli*." *Nature* 427(6977): 839-843.
- Schellenberger, J. and B. O. Palsson (2009). "Use of randomized sampling for analysis of metabolic networks." *The Journal of biological chemistry* 284(9): 5457-5461.



# Extra Slides

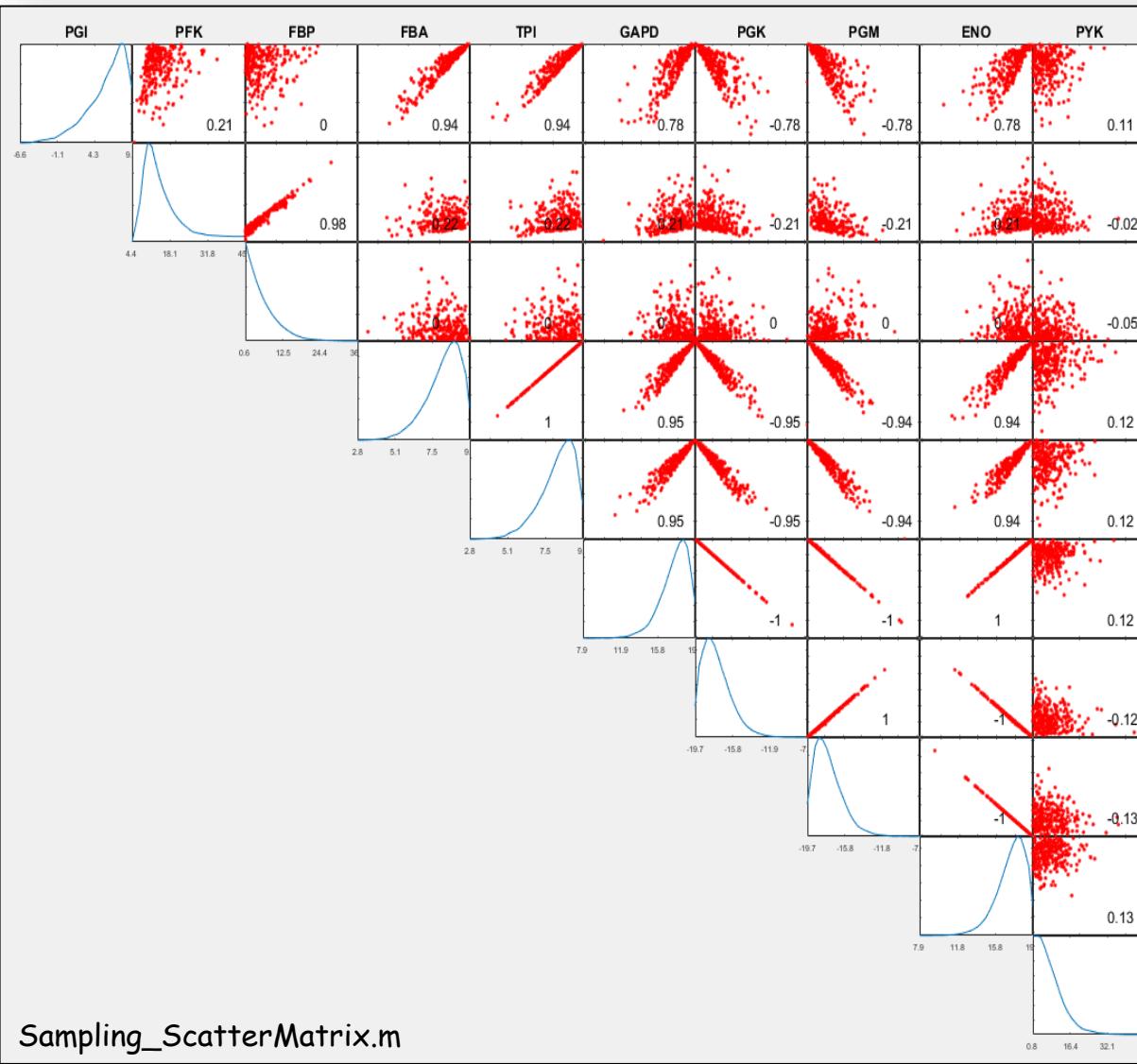


# Scatter Matrix Results

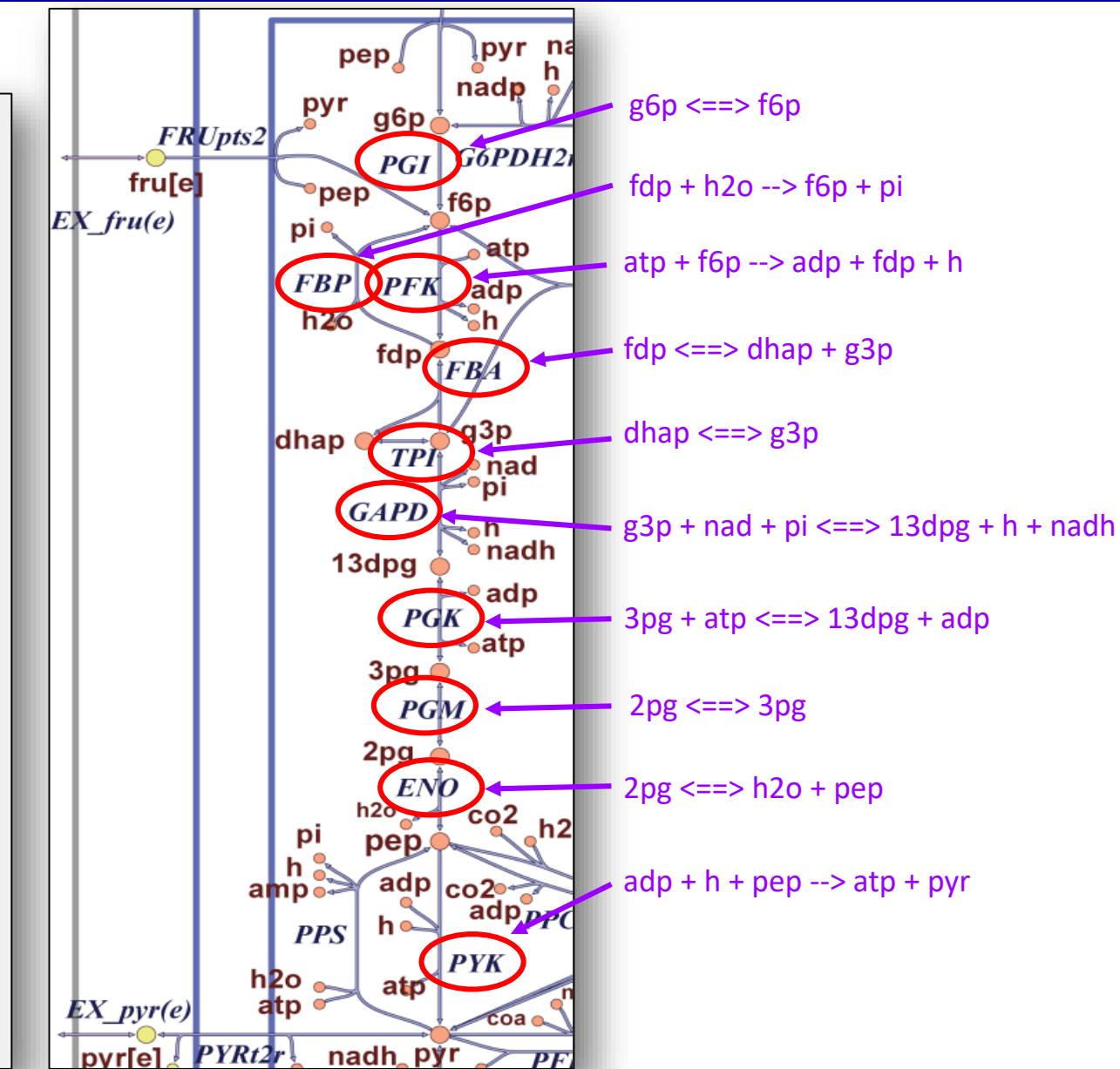




# Scatter Matrix Results



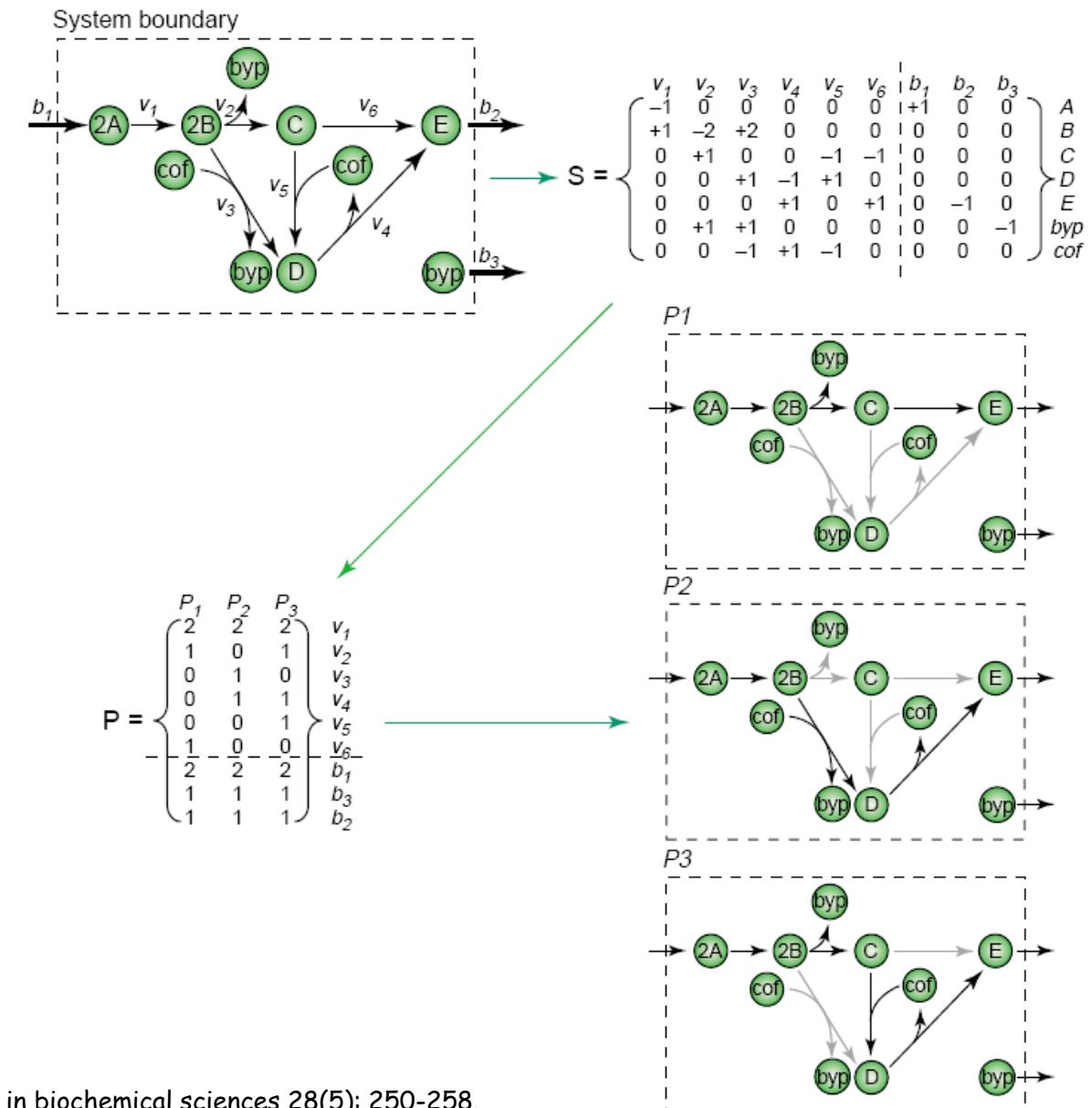
Sampling\_ScatterMatrix.m





# Network-based Pathways of a Sample Reaction Network

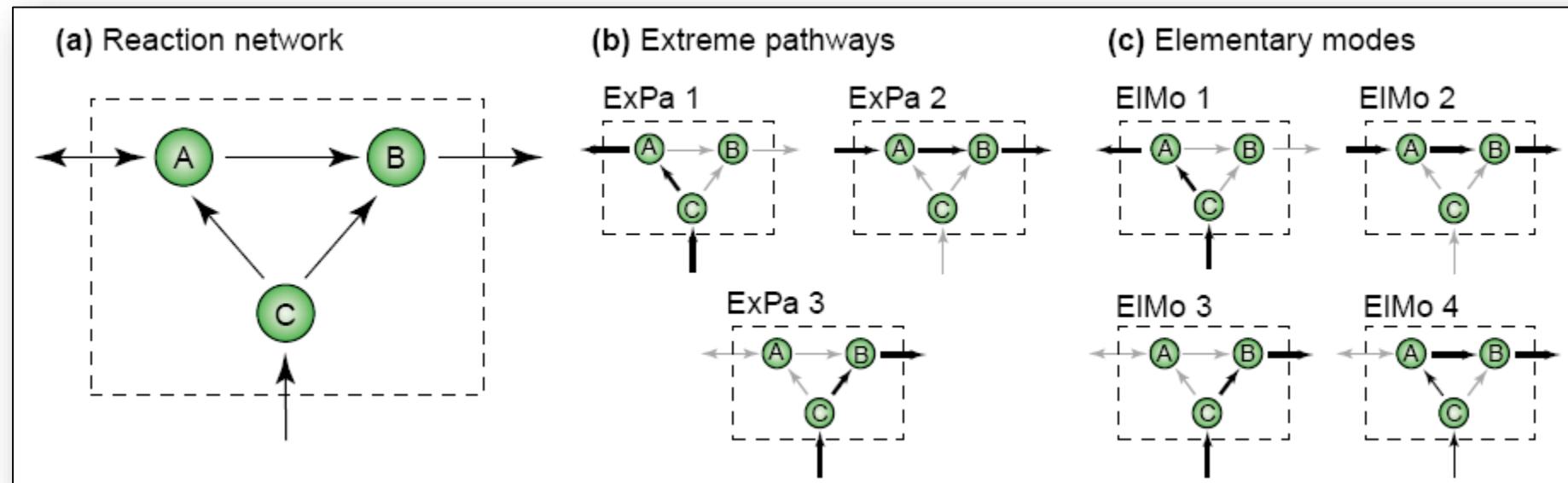
- The reaction network is represented by a stoichiometric matrix ( $S$ ) with rows representing the participation of metabolites in reactions and columns as the stoichiometric coefficients for the individual reactions.
- Matrix is analyzed with pathway analysis (elementary modes or extreme pathways).
- 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 ( $P1$ ,  $P2$ ,  $P3$ ).



Papin, J. A., N. D. Price, et al. (2003). "Metabolic pathways in the post-genome era." Trends in biochemical sciences 28(5): 250-258.



# Extreme Pathways vs Elementary Modes



- The simple reaction network can be decomposed into three extreme pathways (ExPa) and four elementary modes (ElMo).
- 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 systematically independent; note that ElMo 1 (ExPa 1) and ElMo 2 (ExPa 2) can be combined to give ElMo 4.

Papin, J. A., N. D. Price, et al. (2003). "Metabolic pathways in the post-genome era." Trends in biochemical sciences 28(5): 250-258.



# Ethanol Production

```
% Input the E.coli core model
```

```
model = readCbModel('ecoli_core_model.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);  
modelWT = model;
```

```
% Knockout reactions for mutant model
```

```
model = changeRxnBounds(model,'NADH16',0,'b');
```

```
model = changeRxnBounds(model,'PTAr',0,'b');
```

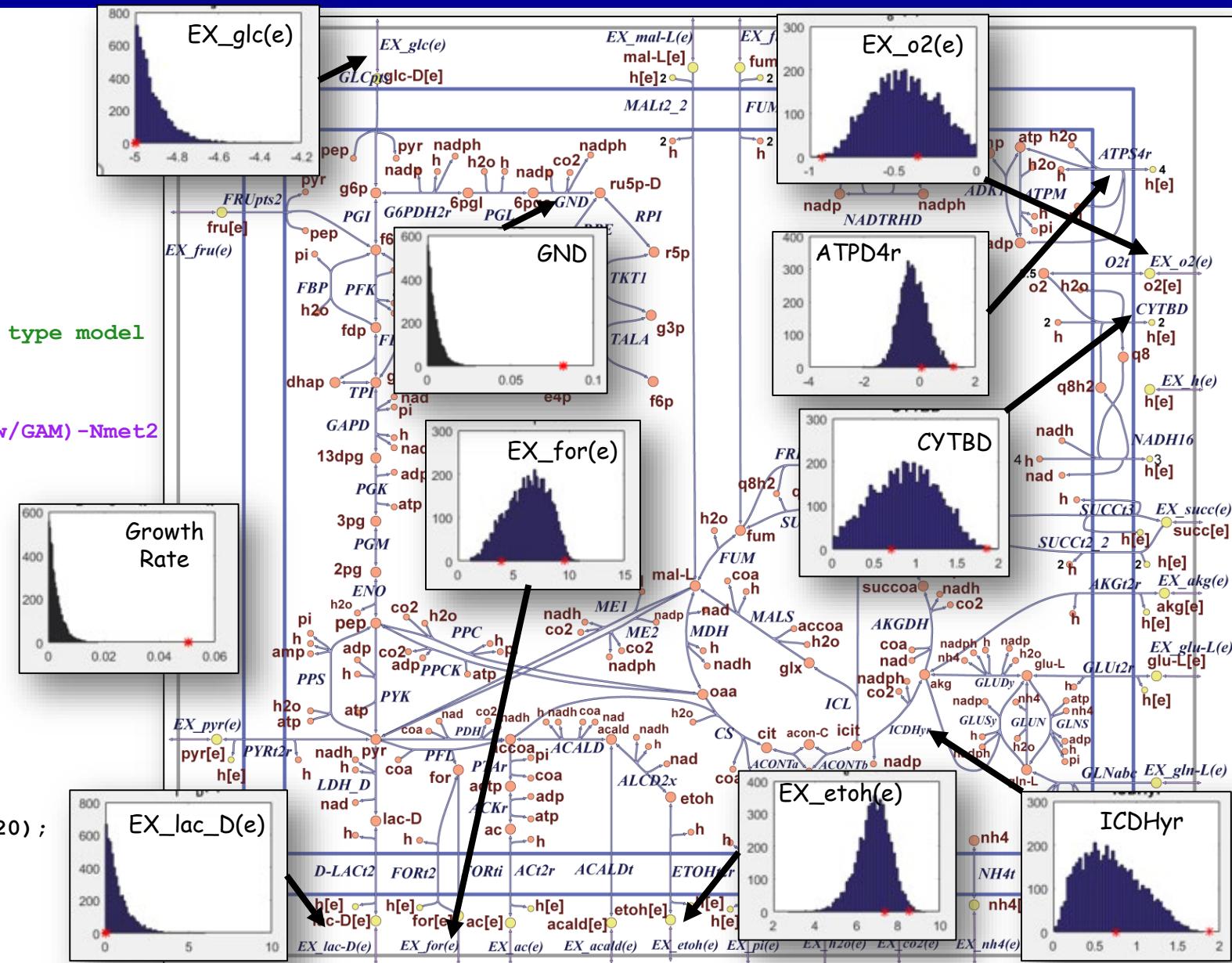
```
model = changeRxnBounds(model,'TKT2',0,'b');
```

```
Mutantsolution = optimizeCbModel(model,'max',0,0);  
modelMutant = model;
```

```
% Sample model
```

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

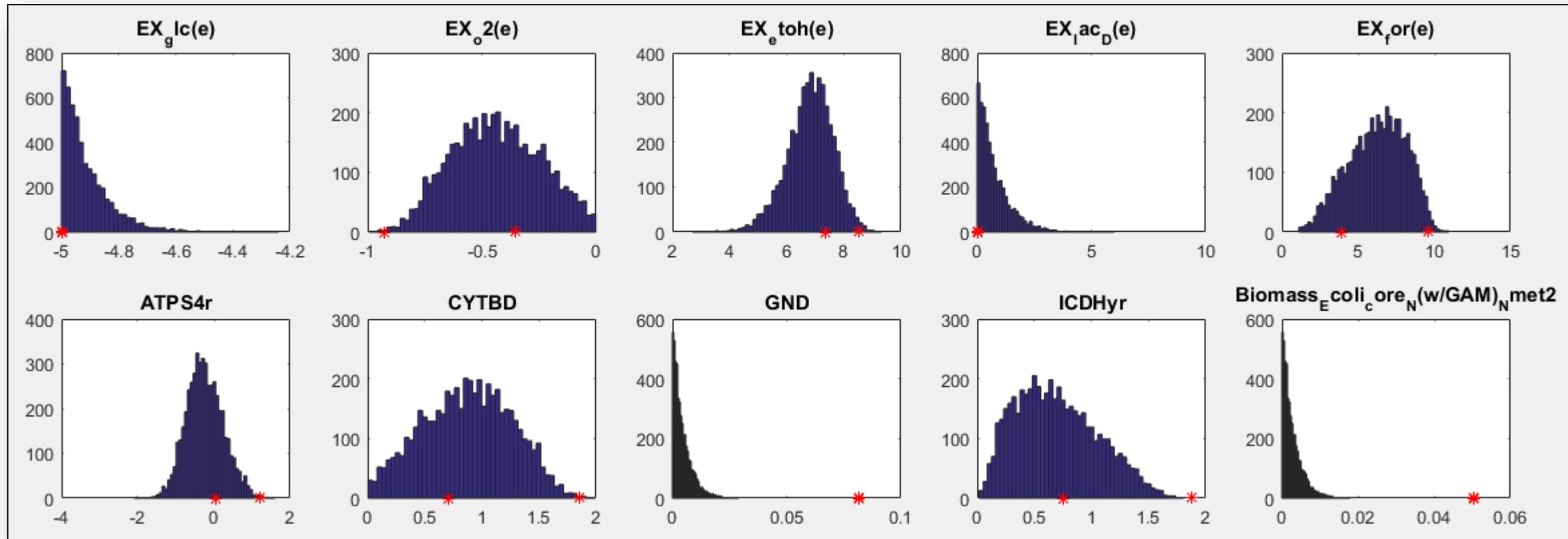
EthanolProduction\_GDLS\_MOMA\_Sampling.m





# Ethanol Production Example

$\text{EX_glc}(e) > -5$ ;  $\text{EX_o}_2(e) > -20$ ; Knockouts = {NADH16, PTAr, TKT2}; Points = 5000, MaxTime = 120



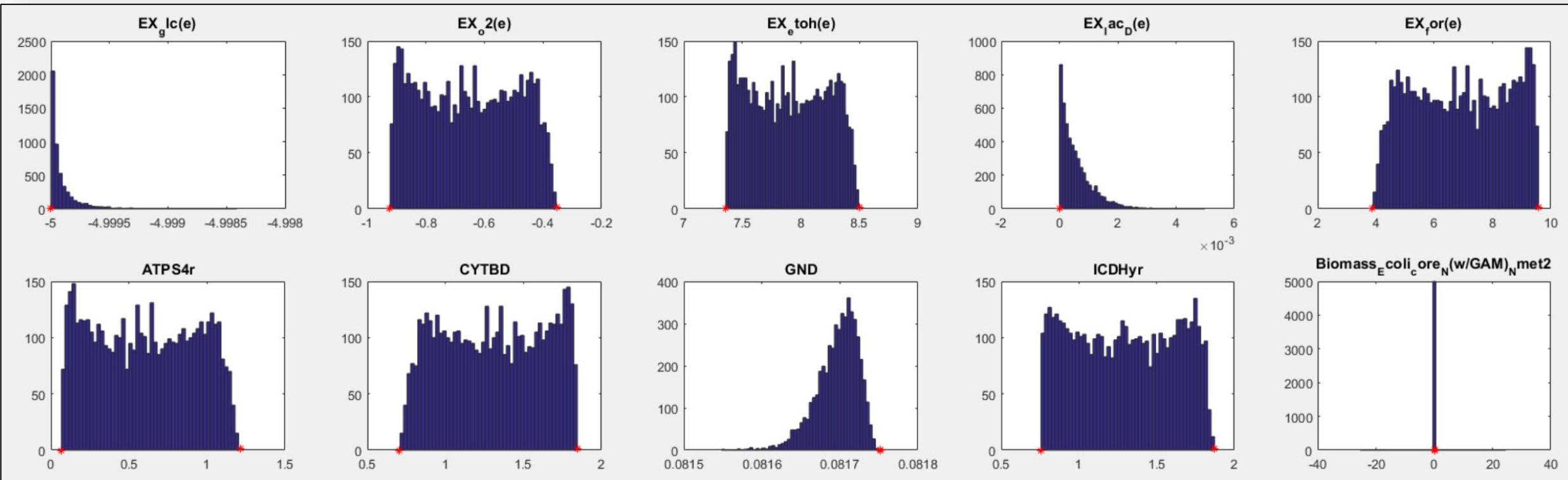
EthanolProduction\_GDLS\_MOMA\_Sampling.m



# Ethanol Production Example

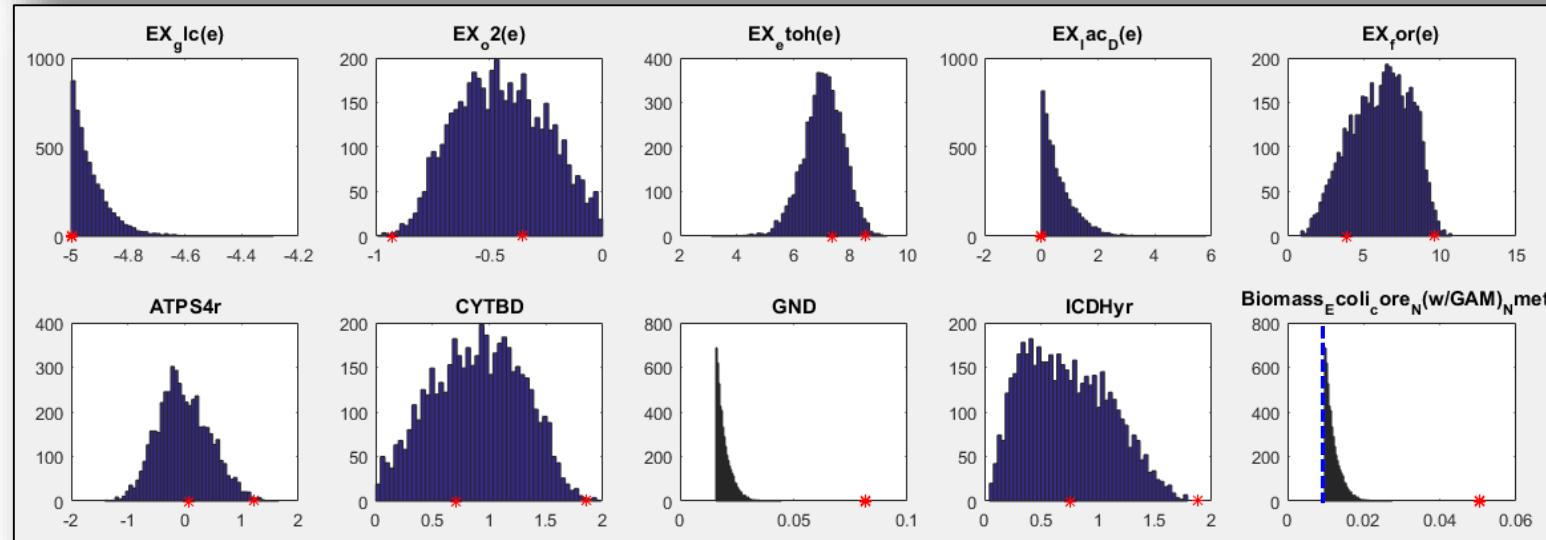
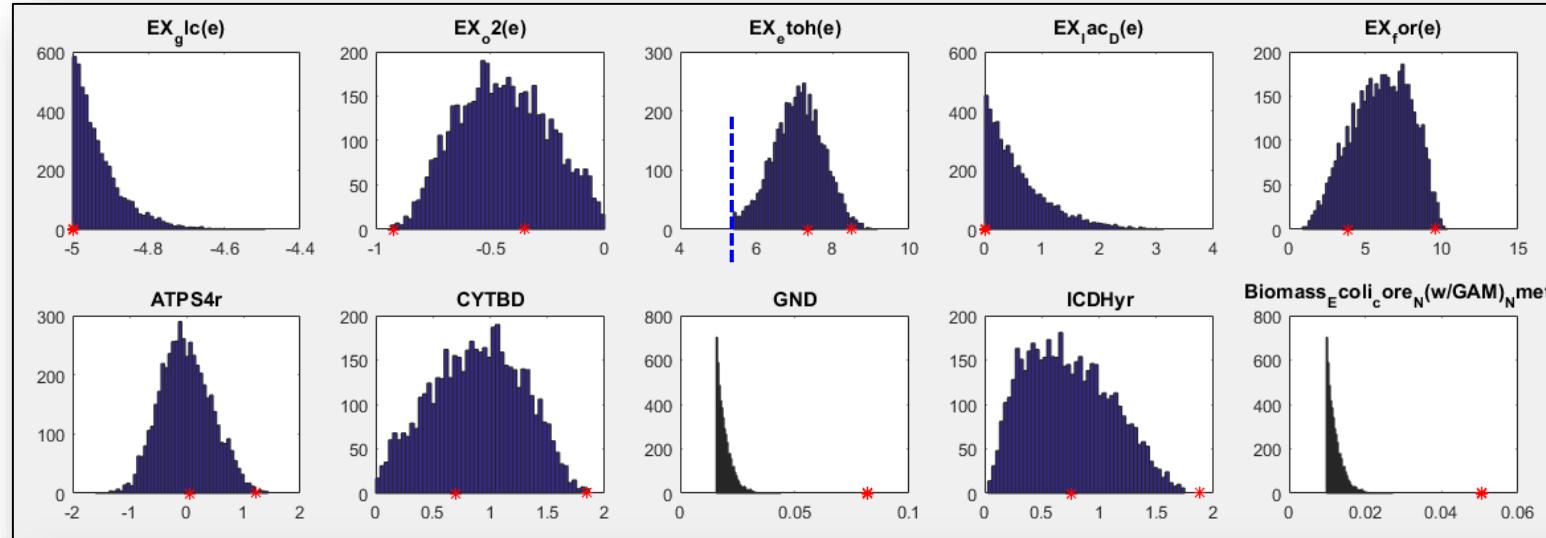
Objective Function set to the Maximum Value used in FVA

$EX_{glc}(e) > -5$ ;  $EX_{o2}(e) > -20$ ; Knockouts = {NADH16, PTAr, TKT2}; Points = 5000, MaxTime = 120

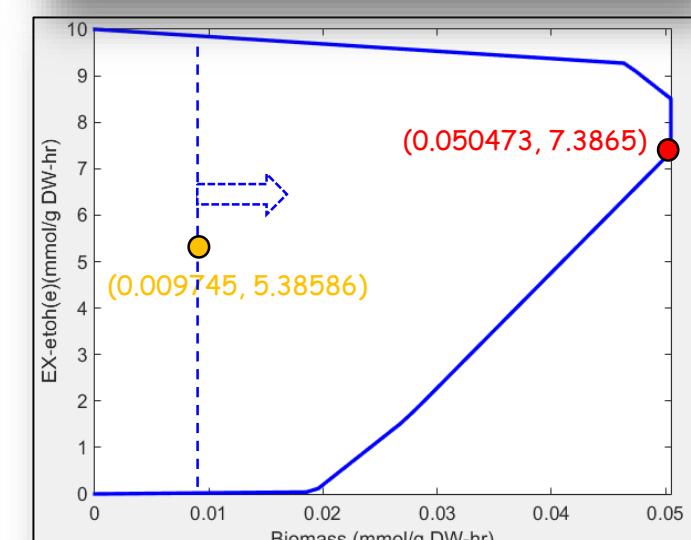
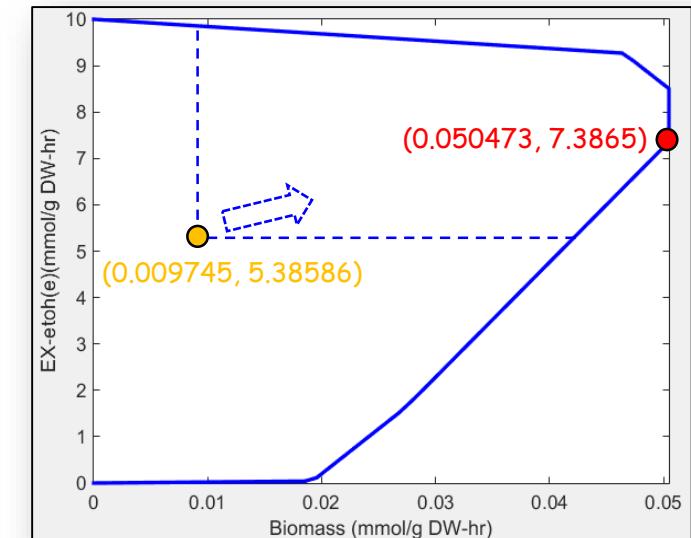




# Using MOMA results as Lower Constraint for Ethanol Production



EthanolProduction\_GDLS\_MOMA\_Sampling.m



EthanolProduction\_ProductionEnvelope\_GDLS\_Mutants.m