Thiele and Palsson,

"A protocol for generating a high-quality genome-scale metabolic reconstruction",

Supplemental methods 1

Supplemental tables

Table S1: List of cellular compartments used in reconstructions (may not be exhaustive). For example, membrane compartments were not considered as reconstructions generally do not account for these, and detailed measurements are difficult to do. [#] Symbol may vary between reconstructions. [*] Lysosome has been defined as a compartment but has not been used yet in any reconstruction. [a] Leishmania major[1]. [b] *Aspergillus nidulans*[2], [c] *Chlamydomonas reinhardtii*[3].

Compartment	Commonly used symbol[#]	Achaea	Bacteria	Eukaryotic pathogens[a]	Fungi [b]	Photosynthetic eukarya[c]	Baker's yeast	Human
Extracellular space	[e]							
Periplasm	[p]							
Cytoplasm	[c]							
Nucleus	[n]							
Mitochondrion	[m]							
Chloroplast	[h]							
Lysosome[*]	[l]							
Vacuole	[v]							
Golgi apparatus	[g]							
Endoplasmatic	[r]							

reticulum								
Peroxisome	[x]							
Flagellum	[f]							
Glyoxysome	[o]							
Glycosome	[y]							
Acidocalcisome	[a]							

Table S2: List of spontaneous reactions present in *E. coli*'s metabolic reconstruction[4]. Note that this list is not complete. H^+ proton. H_2O water. GTP guanosine triphosphates. XTP Xanthosine triphosphates.

Abbreviation	**Name**	**Reaction**	**Pathway**
AOBUTDs	L-2-amino-3-oxobutanoate decarboxylation (spontaneous)	[L-2-Amino-3-oxobutanoate + H^+ --> Aminoacetone + CO_2	Threonine and Lysine Metabolism
DHPTDCs	4,5-dihydroxy-2,3-pentanedione cyclization (spontaneous)	4,5-dihydroxy-2,3-pentane-dione --> H_2O + 4-hydroxy-5-methyl-3(2H)-furanone	Methionine Metabolism
FALGTHLs	formaldehyde glutathione ligase (spontaneous)	formaldehyde + Reduced glutathione <==> hydroxymethylglutathione	Cofactor and Prosthetic Group Biosynthesis
G5SADs	L-glutamate 5-semialdehyde dehydratase	L-glutamate 5-semialdehyde --> 1-pyrroline-5-carboxylate + H^+ + H_2O	Arginine and Proline Metabolism

	(spontaneous)		
GTPHs	GTP amine hydrolysis (spontaneous)	GTP + H^+ + H_2O --> Ammonium + XTP	Nucleotide Salvage Pathway
METOX1s	methionine oxidation (spontaneous)	hydrogen peroxide + L-methionine --> H_2O + L-methionine sulfoxide	Methionine Metabolism

Table S3: List of more elaborate, published tools that have been developed for metabolic reconstructions and that can be used to refine and expand the content. Many of these tools propose additional metabolic functions to the reconstruction to match observed phenotypes and thus propose new hypothesis that can be readily used to verify experimentally. Note that these tools are not implemented in the COBRA Toolbox and should be obtained directly from the references.

Tool Name	**Description**
ADOMET[5]	Gap filling: Identification of candidates genes based on local structure of the metabolic network using gene clustering, phylogentic information and others.
BOSS[6]	Alternate cellular objectives: identifies candidate objective reaction that can be an existing reaction, a combination of existing reactions, or a previously uncharacterized reaction. However this reaction must be linear and stoichiometric. The framework relies on a reconstructed network and experimental data.
GapFill[7]	Gap filling: proposes changes or additions to the network to connect blocked metabolites to the remainder of the network
GapFind[7]	Gap filling: systematic identification of metabolites along blocked reactions ('blocked metabolites')

GrowMatch[8]	An automated method for reconciling *in silico*/*in vivo* growth predictions.
OptFind[9]	Alternate cellular objectives: uses experimental data, a reconstructed network and a scoring system to identify a network reaction that is likely to be a component of the cellular objective function
SMILEY[10]	Gap filling: proposes possible missing functions given a metabolic model and physiological, growth data

Table S4: The soluble pool can contain numerous polyamines, vitamins and cofactors**.** This list represents the metabolites considered by the *E. coli* reconstruction[4].

Abbr	Name
putre	Putrescine
spmd	Spermidine
accoa	Acetyl-CoA
coa	Coenzyme A (CoA)
succoa	Succinyl-CoA
malcoa	Malonyl-CoA
nad	Nicotinamide adenine dinucleotide
nadh	Nicotinamide adenine dinucleotide - reduced
nadp	Nicotinamide adenine dinucleotide phosphate
nadph	Nicotinamide adenine dinucleotide phosphate - reduced
udcpdp	Undecaprenyl diphosphate
10fthf	10-Formyltetrahydrofolate
thf	5,6,7,8-Tetrahydrofolate
mlthf	5,10-Methylenetetrahydrofolate
5mthf	5-Methyltetrahydrofolate
chor	Chorismate
enter	Enterochelin
gthrd	Reduced glutathione
pydx5p	Pyridoxal 5'-phosphate (Vitamin B6)

amet	S-Adenosyl-L-methionine
thmpp	Thiamine diphosphate
adocbl	Adenosylcobalamin
q8h2	Ubiquinol-8
2dmmql8	2-Demethylmenaquinol 8
mql8	Menaquinol 8
hemeO	Heme O
pheme	Protoheme
sheme	Siroheme
ribflv	Riboflavin
fad	Flavin adenine dinucleotide oxidized

Table S5: Calculation of biomass coefficient of ions, many of which are necessary for structure and/or catalytic activity of enzymes. * Ion abbreviations in *E. coli* reconstruction[4]. [a] Assumed to be the same as most other metals. [b] Assumed to be second most abundant cation (based on Neidhardt *et al.*[11]). [c] Assumed to be the same as phosphate (PO_4). CyberCell Database (CCDB, see Table 1 for the link).

Type	Concentration (CCDB)	Concentration (mM)	Species*	Molar fraction (mM species/mM total)
				ratio
Number of K ions	9 *10^7 (200-250 mM)	225	k	0.7142
Number of Fe ions	7 *10^6 (18 mM)	9	fe2	0.0286
		9	fe3	0.0286
Number of Mg ions	4 *10^6 (10 mM)	10	mg2	0.0317
Number of Cl ions	2 *10^6 (6 mM)	6	cl	0.0190
Number of Ca ions	2 *10^6 (6 mM)	6	ca2	0.0190
Number of Na ions	2 *10^6 (5 mM)	5	na	0.0159
Number of PO_4 ions	2 *10^6 (5 mM)	5	pi	0.0159
Number of Cu ions	1.7 *10^6 (4 mM)	4	cu2	0.0127
Number of Mn ions	1.7 *10^6 (4 mM)	4	mn2	0.0127
Number of Mo ions	1.7 *10^6 (4 mM)	4	mobd	0.0127
Number of Zn ions	1.7 *10^6 (4 mM)	4	zn2	0.0127
Number of Cobalt ions	1.7 *10^6 (4 mM)[a]	4	cobalt2	0.0127
Number of NH_4 ions	6 *10^6 (15 mM)[b]	15	nh4	0.0476
Number of SO_4 ions	2 *10^6 (5 mM)[c]	5	so4	0.0159
	Total ion concentration	315		

Table S6: Useful functions in the COBRA Toolbox for reconstruction purposes. * For details on syntax, please refer to the COBRA Toolbox and Becker et al.[12].

Action	**COBRA Toolbox command***	**Comment**
Add reaction	model = addReaction(model,'newRxn1','A + 2 B -> C')	Can be used to add reactions such as: A + 2 B --> C
Add demand reaction	[model,rxnNames] = addDemandReaction(model,metaboliteNameList)	Can be used to add reactions such as: A -->
Add sink reaction	[model] = AddSinkReactions(model,metabolites,bounds)	Can be used to add reactions such as: A <==>
Remove reaction	modelOut = removeRxns(model,rxnRemoveList)	
Write model in file	writeCbModelRecon(model,format,fileName)	Format of the file can be sbml, plain text, or xls
Open sbml model	model = readCbModel(fileName)	
Open reconstruction from xls	model = xls2model(RxnFileName,MetFileName)	
Change gene association	model = changeGeneAssociation(model,rxnName,grRule,geneNameList,systNameList)	
Change reaction bounds	model = changeRxnBounds(model,rxnNameList,value,boundType)	
Change objective function	model = changeObjective(model, rxnNameList, objectiveCoeff)	Multiple reactions can be set as objective function. They have to be satisfied

		in the specified ratio as define by objectiveCoeff (default = 1).
Print constraint reactions	PrintConstraints(model,MinInf, MaxInf)	The function identifies all reactions whose upper bounds (or lower bounds) are less (or greater) than the maximum value (MaxInf) (or minimum value – MinInf)
Load reconstruction into Matlab	model = xls2model(RxnFileName,MetFileName);	

Supplemental figures

Figure S1: Collecting information for draft reconstruction using, e.g., Entrez Gene as annotation source.

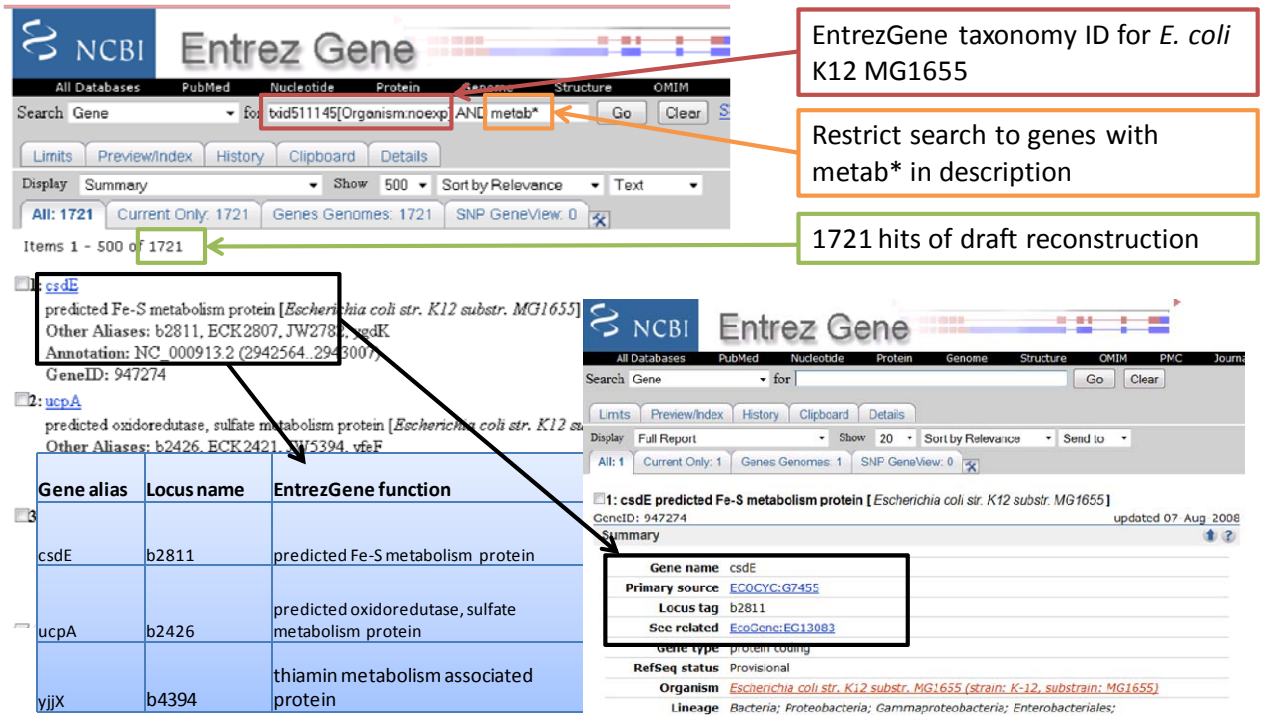

Figure S2: Assessing the metabolic "environment" or "connectivity" of a metabolite. Here, Maltose-6-phosphate is highlighted on the KEGG[13] map for "Starch and Sucrose Metabolism".

All annotated *E. coli* genes (MG1655) in KEGG[13] were colored green. Enzymes that are currently not annotated or not found are shown with white boxes. Maltose-6-phosphate is a dead-end metabolite in *E. coli*'s metabolic reconstruction[4]. The enzyme 3.2.1.122 is currently not annotated (in KEGG[13]). There are only two enzymes in the KEGG database[13] that seem to produce/consume Maltose-6-phosphate: 2.7.1.69 and 3.2.1.122. In contrast, D-Glucose-6-Phosphate is highly connected in the *E. coli* reconstruction[4].

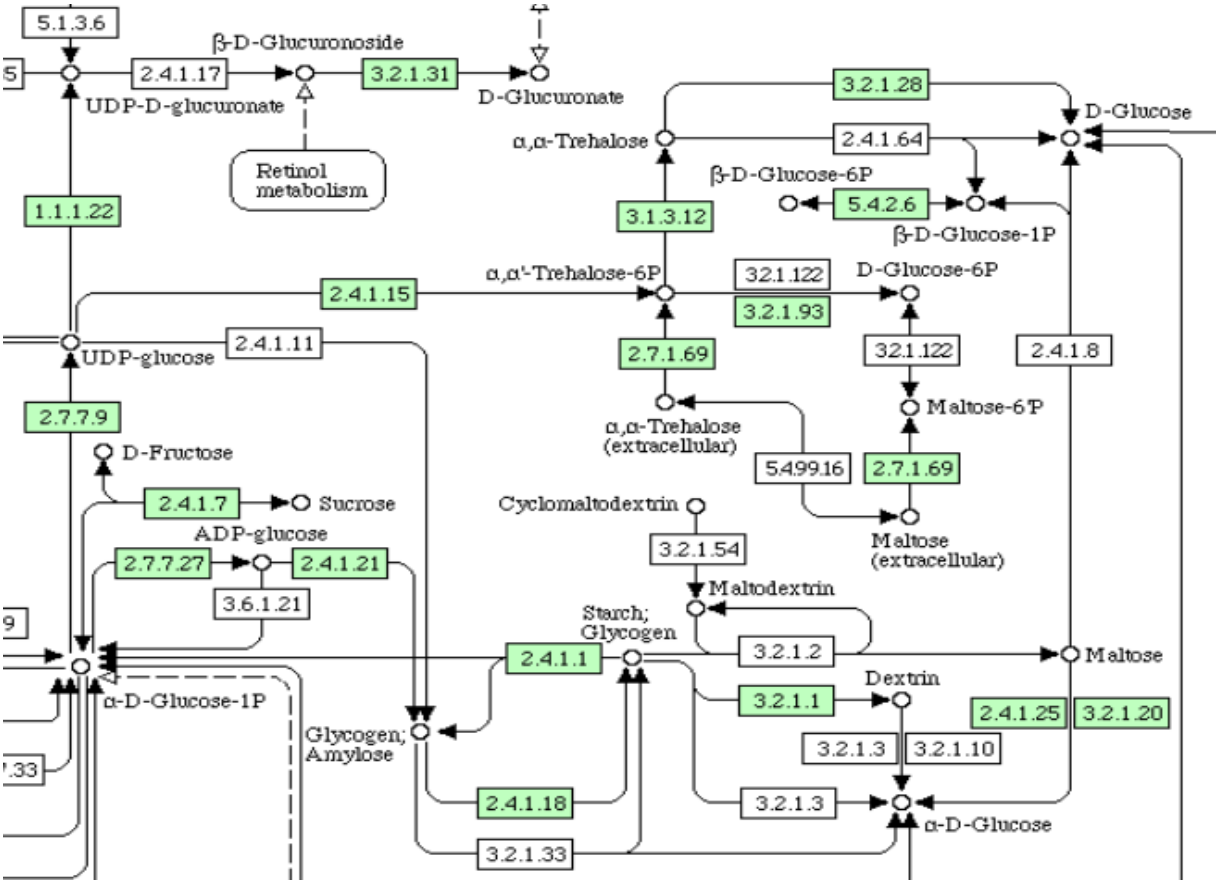

Figure S3: Example of biomass composition determination for *Pseudomonas putida* KT 2440. **A.** Chemical composition of *E. coli* adopted from[11] and utilized as a template for *P. putida* KT2440, since no extensive information was available. **B.** Protein composition in *P. putida* broken down by monomer contributions in mmol/g_{DW}. **C.** Phospholipid contributions to the biomass function where PE is phosphatidylethanolamine, PG is phosphatidylglycerol, and CL is cardiolipin. **D.** dNTP composition of the entire *P. putida* chromosomal genome. The data are

obtained from direct measurements, literature, or can be estimated from genome information (blue arrows).

A

Cellular Component	Cellular Content %(w/w)
Protein	55%
RNA	20.5%
DNA	3.1%
Lipids	9.1%
LPS	3.4%
Peptidoglycan	2.5%
Glycogen	2.5%
Polyamines	0.4%
Other	3.5%
Total	100.00%

B

Monomer	(mmol/g_{DW})	Monomer	(mmol/g_{DW})
Ala	0.482	Leu	0.507
Arg	0.286	Lys	0.145
Asn	0.128	Met	0.098
Asp	0.230	Phe	0.154
Cys	0.045	Pro	0.212
Glu	0.243	Ser	0.243
Gln	0.203	Thr	0.206
Gly	0.348	Trp	0.063
His	0.102	Tyr	0.110
Ile	0.197	Val	0.314

C

Phospholipids	(mol/mol) %	Average MW	Content % (w/w)	mmol/g_{DW}
PE	64.95%	699.1	58.20%	0.0325
PG	21.50%	700.3	19.30%	0.0324
CL	10.06%	1508	19.45%	0.0151

D

DNA Monomer	Number of bp	Content % (mol/mol)	(mmol/g_{DW})
dATP	1186504	19.19%	0.0122
dCTP	1889954	30.57%	0.0197
dGTP	1913381	30.95%	0.0195
dTTP	1192024	19.28%	0.0123
Total	6181863	100.00%	

List of standards that have been used in numerous metabolic reconstructions

Naming conventions:

- Reaction abbreviations are capitalized.

- Reaction names suffix standards:

Reaction Type	Suffix	Example
ABC transporter	-abc	ALAabc
Transport reactions	-t	GLCt1
Reversible reactions	-r	GLCt1r
Irreversible reactions	-i	PTRCt3i
Dehydrogenase reactions	-DH	PDH
Synthetase reactions	-S	ATPS
Kinase reactions	-K	ACKr
Chloroplast reactions	-h	HEX1h
Endoplasmic Reticular reactions	-er	CERASE124er
Extracellular reactions	-e	AKGDHe
Golgi reactions	-g	S6T12g
Lysosomal reactions	-l	10FTHFtl
Mitochondrial reactions	-m	AKGDm
Nucleus reactions	-n	UMPK3n
Peroxisomal reactions	-x	SCP3x
Periplasmic reactions	-pp	PPTHpp
Vacuole	-v	GLCGSDv

Of course, combinations of suffixes are possible, for example see 10FTHFtl above. When choosing abbreviations, try to construct the root

- Try to construct the root of the reaction abbreviation based on the enzyme name, for example AKGDHe = Alpha-ketoglutarate Dehydrogenase (in the extracellular compartment).

- Metabolites are lower case.

- Metabolite formulas in the charged state are based on the chemical structure at a pH of 7.2. The charge state can be defined using tools (such as pKaDB) and consulting Adam, Ines, Monica, and Neema.

- Do not use wildcard characters in abbreviations: no apostrophes, no parentheses, etc. The exceptions to this are the use of parentheses in sink and demand reactions.

Notes Fields (reactions and compounds):

- Add references whenever possible (e.g. PMID, KEGG ID, PubChem ID, PubSubstance ID), if these identifiers are not available, make sure to state this explicitly.

- Add any detailed descriptions necessary to understand any specific rationale for the addition.

- Reactions must always be charge balanced. If not balanced, state why.

- Always add your full name or the initials to the note field. This increases traceability.

Example use of functions listed in the protocol

In the following we will illustrate example input for the COBRA Toolbox functions used in Step 38 to 96 using the E. coli core reconstruction as the model of choice. When Matlab/Toolbox commands are following by a ';' the output is not printed. Omitting the ';' invokes printing of the variable content. The output is given in gray in the following section, while commands are listed in orange and start with a '>>'.

38| **Initialize the COBRA Toolbox.**

Setting the linear programming (LP) solver, for example, 'glkp':

```
>> solverOK = changeCobraSolver('glpk','LP');
```

39| **Load reconstruction into Matlab**.

The reaction list is contained by the file 'RxnList.xls' and the metabolite list is in file 'MetList.xls':

```
>> modelEcoliCore = xls2model('RxnList.xls','MetList.xls');
```

The reconstruction is contained in the resulting model structure: modelEcoliCore. See Figure 10 for the data contained in the different structure fields. We will use this model structure for all consequent computation if not noted differently.

The content of the structure can be assessed as follows:

- You wish to see the abbreviation of the first reaction in the model:

  ```
  >> modelEcoliCore.rxns{1}
  ```

- You wish to see the entry of the stoichiometric matrix of the 10th reaction (column) and 4th metabolite (row):

  ```
  >> modelEcoliCore.S(4,10)
  ```

- You want to change the lower bound (lb) of the 5th reaction to 10 mmol/g_{DW}/h (without using any COBRA Toolbox functions):

  ```
  >> modelEcoliCore.lb(5) = 10;
  ```

- You want to add afield to the model structure.
  - A note:

```
>> modelEcoliCore.newField = ‘ABC – a note’;
```

- An array B = [1 2 3]

```
>> modelEcoliCore.newField = B;
```

- Create a list of strings:

```
>> ListStrings = {‘A’ ‘B’ ‘C’};
```

- Create a list of numbers:

```
>> ListNumbers = [1 2 3];
```

- Transpose a list :

```
>> ListTranspose = List’;
```

- Find the index of a reaction, e.g., ‘ATPM', in the model

```
>> RxnNumber = strmatch('ATPM', modelEcoliCore.rxns,'exact')
RxnNumber =
   15
```

40| Verify S matrix.

```
>>spy(modelEcoliCore.S)
```

41| Set objective function. We will set the biomass reaction (Biomass_Ecoli_core_w_GAM) of the *E. coli* core model as objective function:

```
>> modelEcoliCore = changeObjective(modelEcoliCore,
'Biomass_Ecoli_core_w_GAM')
New objective:
17      Biomass_Ecoli_core_w_GAM
```

If you wish to check wish reaction(s) make up the objective function use the following function:

```
>> [ObjFunction, ObjCoeff] = checkObjFunction(modelEcoliCore)
ObjFunction =
   'Biomass_Ecoli_core_w_GAM'
ObjCoeff =
    1
```

42| **Set simulation constraints**.

Let's assume that you would like to set the lower bound of the ATP maintenance reaction ('ATPM') to 8.39 mmol/g_{DW}/h:

```
>>modelEcoliCore = changeRxnBounds(modelEcoliCore,'ATPM',8.39,'l');
```

and the upper bound of the 'ATPM' reaction to 8.39 mmol/g_{DW}/h:

```
>>modelEcoliCore = changeRxnBounds(modelEcoliCore,'ATPM',8.39,'u');
```

The same effect could have been achieved using:

```
>>modelEcoliCore = changeRxnBounds(modelEcoliCore,'ATPM',8.39,'b');
```

Let's assume that you would like to set the lower bound of the 'ATPM' reaction to 8.39 mmol/g_{DW}/h and the ATP synthetase ('ATPS4r') to an upper bound of 100 mmol/g_{DW}/h:

```
>>modelEcoliCore = changeRxnBounds(modelEcoliCore,'ATPM' ,8.39,'l' );
>>modelEcoliCore = changeRxnBounds(modelEcoliCore, 'ATPS4r',100, 'u');
```

The set constraints can be checked using the following function:

```
>>PrintConstraints(modelEcoliCore, -1000,1000)
MinConstraitns:
ATPM 8.390000e+000
ATPS4r          8.390000e+000
EX_glc(e)       -1.000000e+001
MaxConstraints:
ATPM 8.390000e+000
ATPS4r          1.000000e+002
```

Note that the 'MinInf' and 'MaxInf' were set to -1000 and 1000, respectively, as this numbers represent infinity in the E. coli core model. Other models may have different infinities.

43| Test if network is mass- and charge balanced.

Check for one reaction (e.g., ATPM):

```
>> [UnbalancedRxns] = CheckMassChargeBalance(modelEcoliCore,'ATPM')
```

Check for multiple reactions:

```
>> [UnbalancedRxns] = CheckMassChargeBalance(modelEcoliCore,{'ATPM'
'ATPS4r'})
```

Check for entire network:

>> [UnbalancedRxns] = CheckMassChargeBalance(modelEcoliCore)

Or

>> [UnbalancedRxns] = CheckMassChargeBalance(modelEcoliCore, modelEcoliCore.rxns)

45| Identify metabolic dead-ends

>> [Gaps] = GapAnalysis(modelEcoliCore)

Gaps =

"

Test for stoichiometrically balanced cycles (SBCs).

51| Test for Type III pathways. Therefore, use the following function: TestForTypeIIIPathways. The indices of the exchange reactions are given by [24:43], which creates a list from 23 to 43. If the exchange reactions are distributed across the network, a list of all indices can be given as input (e.g., [1 2 5]). The output filename can be specified with 'test', it receives automatically the extension '.expa'. The filename is optional, the default name is: 'ModelTestTypeIII'

>> TestForTypeIIIPathways(modelEcoliCore,[24:43],'test');

Error with X3.exe:

ERROR:: Insufficient Input.

Either no reactions or metabolites were detected.

This program will now terminate

This error message is returned from X3.exe since there are no SBCs in the E. coli core network.

Test if biomass precursors can be produced in standard medium

60| Obtain the list of biomass components:

>> [BiomassComponent, BiomassFraction] = PrintBiomass(modelEcoliCore,17);

pep[c] -5.191000e-001

f6p[c] -7.090000e-002

pyr[c] -2.832800e+000

atp[c] -5.981000e+001

adp[c] 5.981000e+001

h[c] 5.981000e+001

g6p[c] -2.050000e-001

coa[c] 3.747800e+000

nad[c] -3.547000e+000

accoa[c] -3.747800e+000

….

61| Add demand function for each biomass precursor:

>> [modelEcoliCore_NEW ,rxnNames] = addDemandReaction(modelEcoliCore, BiomassComponent);

DM_pep[c] pep[c] ->

DM_f6p[c] f6p[c] ->

DM_pyr[c] pyr[c] ->

DM_atp[c] atp[c] ->

DM_adp[c] adp[c] ->

DM_h[c] h[c] ->

DM_g6p[c] g6p[c] ->

….

For each biomass component i, perform the following test:

62| Change objective function to the demand function:

>> modelEcoliCore_NEW = changeObjective(modelEcoliCore_NEW, 'DM_pep[c]');

New objective:

96 DM_pep[c]

Coefficient	Metabolite	#	Reaction	#
-1	pep[c]	2	DM_pep[c]	96

63| Maximize (‘max’) for new objective function (Demand function)

>> FBAsolution = optimizeCbModel (modelEcoliCore_NEW,'max');

```
FBAsolution =
    f: 20
    x: [118x1 double]
 stat: 1
 solver: 'tomlab_cplex'
 time: 0.0066
```

FBAsolution is a structure containing the result of the optimization. FBAsolution.f gives the maximal value of the objective reaction (i.e., 'DM_pep[c]'), which is 20 mmol/gDW/h. This means that our *E. coli* core model can produce pep[c]. Just to remember us, the model constraints are:

```
>> PrintConstraints(modelEcoliCore_NEW, -1000,1000)
MinConstraitns:
ATPM 8.390000e+000
ATPS4r      8.390000e+000
EX_glc(e)   -1.000000e+001
MaxConstraints:
ATPM 8.390000e+000
ATPS4r      1.000000e+002
```

FBAsolution.x contains the flux value for each reaction in the network. Use:

```
>> PrintFluxVector(modelEcoliCore_NEW,FBAsolution.x)
FRUpts2     0
PFK  10
PGI  10
GLCpts      10
ACALD       0
ACALDt      0
ACKr  0
ACONTa      0
```

To see which network reactions participate in the optimal solution. Keep in mind that there may be more than one optimal solution (so-called alternate optimal solutions, which

have the same optimal value for the objective function but the internal flux distribution may be different)[14-17] .

Test if model can produce known secretion products.

68| Set the constraints to the desired medium condition (e.g., minimal medium + carbon source). For changing the constraints use the following function:

```
>> modelEcoliCore = changeRxnBounds(modelEcoliCore, {'EX_glc(e)' 'EX_o2(e)'},[-10 -18.5],'l')
```

i. If the model shall be required to grow in addition to producing the by-product, set the lower bound (boundType = 'l') of the biomass reaction ('rxnNameList ') to the corresponding value ('value').

```
>> modelEcoliCore = changeRxnBounds(modelEcoliCore, 'EX_ac(e)',2,'l')
```

Acetate secretion (EX_ac) was chosen as example.

69| Change the objective function to the exchange reaction of your secretion product (i.e., acetate):

```
>> modelEcoliCore = changeObjective(modelEcoliCore, 'EX_ac(e)');
New objective:
24      EX_ac(e)
Coefficient     Metabolite      #               Reaction                #
-1                      ac[e]           23      EX_ac(e)                24
```

70| Maximize ('max') for the new objective function (as a secretion is expected to have a positive flux value, see Figure 7):

```
>> FBAsolution = optimizeCbModel (modelEcoliCore,'max')
FBAsolution =
f: 20.0000
 x: [95x1 double]
stat: 1
solver: 'tomlab_cplex'
time: 0.0065
```

It seems that the model can produce 20 mmol/g_{DW}/h of acetate with the following constraints:

```
>> PrintConstraints(modelEcoliCore, -1000,1000)
MinConstraitns:
ATPM 8.390000e+000
ATPS4r          8.390000e+000
EX_ac(e)        2.000000e+000
EX_glc(e)       -1.000000e+001
EX_o2(e)        -1.850000e+001
MaxConstraints:
ATPM 8.390000e+000
ATPS4r          1.000000e+002
```

Test if model can produce a certain ratio of two secretion products.

73| Add a row to the S matrix to couple the by-product secretion reactions:

```
>> modelEcoliCore_NEW = AddRatioReaction(modelEcoliCore, {'EX_ac(e)' 'EX_for(e)'},[1 1]);
```

Acetate and Formate secretion are coupled to a ratio of 1:1.

Optimize for growth:

```
modelEcoliCore_NEW = changeObjective(modelEcoliCore_NEW,'Biomass_Ecoli_core_w_GAM');
>> FBAsolution = optimizeCbModel (modelEcoliCore_NEW,'max')
FBAsolution =
    f: 0.7873
    x: [95x1 double]
    stat: 1
    solver: 'tomlab_cplex'
    time: 0.0068
```

Test the secretion of Acetate (reaction #: 24) and Formate (#29) in the optimal growth solution:

```
>> FBAsolution.x(24)
ans =
    2.8337
>> FBAsolution.x(29)
```

```
ans =
   2.8337
```

If the model shall be required to growth in addition to producing the by-product, set the lower bound of the biomass reaction to the corresponding value.

```
>> modelEcoliCore_NEW = changeRxnBounds(modelEcoliCore_NEW, 'Biomass_Ecoli_core_w_GAM',0.7873,'b')
```

74| Change the objective function to the exchange reaction of one of your secretion product:

```
>> modelEcoliCore_NEW = changeObjective(modelEcoliCore_NEW, 'EX_ac(e)');
```

75| Maximize for the new objective function (as a secretion is expected to have a positive flux value, see Figure 7):

```
>> FBAsolution = optimizeCbModel (modelEcoliCore_NEW,'max')
```

Check for blocked reactions.

76| Change simulation conditions to rich medium or open all exchange reactions:

```
>> modelEcoliCore_Open = changeRxnBounds(modelEcoliCore, modelEcoliCore.rxns(24:43), -1000,'l');
>> modelEcoliCore_Open = changeRxnBounds(modelEcoliCore, modelEcoliCore.rxns(24:43), 1000,'u');
```

77| Run analysis for blocked reactions. The function returns a list of blocked reactions ('BlockedReactions').

```
>> BlockedReactions = FindBlockedReaction(modelEcoliCore_New);
BlockedReactions =
        []
```

The answer is an empty array since the *E. coli* core network has no blocked reactions.

Compute single gene deletion phenotypes

79| Gene deletion: use the following function in the COBRA Toolbox:

```
>> [grRatio,grRateKO,grRateWT] = singleGeneDeletion (modelEcoliCore);
```

Test if the model can grow fast enough.

89| Determine reduced cost associated with network reactions when optimizing for objective function. Use

```
>> FBAsolution = optimizeCbModel(modelEcoliCore,'max',false)
FBAsolution =
            f: 20.0000
            x: [95x1 double]
            y: [72x1 double]
            w: [95x1 double]
         stat: 1
       solver: 'tomlab_cplex'
         time: 0.0071
```

FBAsolution.y contains the shadow price for each metabolite and FBAsolution.w contains the reduced cost for each network reaction.

Test if the model grows too fast.

93| Single reaction deletion.

```
>> [grRatio,grRateKO,grRateWT] = singleRxnDeletion(modelEcoliCore);
```

94| **Reduced cost**.

See Step 89.

95| Print Matlab model content.

```
>>writeCBmodel(modelEcoliCore,'xls', 'EColiCore.xls')
>>writeCBmodel(modelEcoliCore,'sbml', 'EColiCore.xml')
```

A detailed documentation on how to use the COBRA Toolbox can be found in the protocol[12].

References

1. Chavali, A.K., Whittemore, J.D., Eddy, J.A., Williams, K.T. & Papin, J.A. Systems analysis of metabolism in the pathogenic trypanosomatid Leishmania major. *Molecular systems biology* **4**, 177 (2008).
2. David, H., Ozcelik, I.S., Hofmann, G. & Nielsen, J. Analysis of Aspergillus nidulans metabolism at the genome-scale. *BMC genomics* **9**, 163 (2008).
3. Manichaikul, A. et al. Metabolic network analysis integrated with transcript verification for sequenced genomes. *Nature methods* **6**, 589-592 (2009).
4. Feist, A.M. et al. A genome-scale metabolic reconstruction for *Escherichia coli* K-12 MG1655 that accounts for 1260 ORFs and thermodynamic information. *Molecular systems biology* **3** (2007).
5. Kharchenko, P., Chen, L., Freund, Y., Vitkup, D. & Church, G.M. Identifying metabolic enzymes with multiple types of association evidence. *BMC Bioinformatics* **7** (2006).
6. Gianchandani, E.P., Oberhardt, M.A., Burgard, A.P., Maranas, C.D. & Papin, J.A. Predicting biological system objectives de novo from internal state measurements. *BMC bioinformatics* **9**, 43 (2008).
7. Satish Kumar, V., Dasika, M.S. & Maranas, C.D. Optimization based automated curation of metabolic reconstructions. *BMC Bioinformatics* **8**, 212 (2007).
8. Satish Kumar, V., Maranas, C. D. GrowMatch: An automated method for reconciling in silico/in vivo growth predictions. *PLoS Computational Biology* (in press).
9. Burgard, A.P. & Maranas, C.D. Optimization-based framework for inferring and testing hypothesized metabolic objective functions. *Biotechnology and bioengineering* **82**, 670-677 (2003).
10. Reed, J.L. et al. Systems approach to refining genome annotation. *Proceedings of the National Academy of Sciences of the United States of America* **103**, 17480-17484 (2006).
11. Neidhardt, F.C., Ingraham, J.L. & Schaechter, M. Physiology of the bacterial cell: a molecular approach. (Sinauer Associates, Sunderland, Mass.; 1990).
12. Becker, S.A. et al. Quantitative prediction of cellular metabolism with constraint-based models: The COBRA Toolbox. *Nat. Protocols* **2**, 727-738 (2007).
13. Kanehisa, M. et al. From genomics to chemical genomics: new developments in KEGG. *Nucleic Acids Res* **34**, D354-357 (2006).
14. Reed, J.L. & Palsson, B.O. Genome-Scale In Silico Models of E. coli Have Multiple Equivalent Phenotypic States: Assessment of Correlated Reaction Subsets That Comprise Network States. *Genome Res* **14**, 1797-1805 (2004).
15. Price, N.D., Reed, J.L. & Palsson, B.O. Genome-scale models of microbial cells: evaluating the consequences of constraints. *Nature reviews* **2**, 886-897 (2004).
16. Lee, S., Phalakornkule, C., Domach, M.M. & Grossmann, I.E. Recursive MILP model for finding all the alternate optima in LP models for metabolic networks. *Comp Chem Eng* **24**, 711-716 (2000).
17. Mahadevan, R. & Schilling, C.H. The effects of alternate optimal solutions in constraint-based genome-scale metabolic models. *Metabolic engineering* **5**, 264-276 (2003).