FBA with CobraPy

Keesha Erickson
keeshae@lanl.gov
June 21, 2018
qBio Summer School

Test Installation

Open a python console in pycharm (or start a python shell).
Run:

```
from cobra.test import test_all
test_all()
```

Mine returns:

```
3 failed, 285 passed, 88 skipped, 5 xfailed, 1 xpassed in
159.96 seconds
```

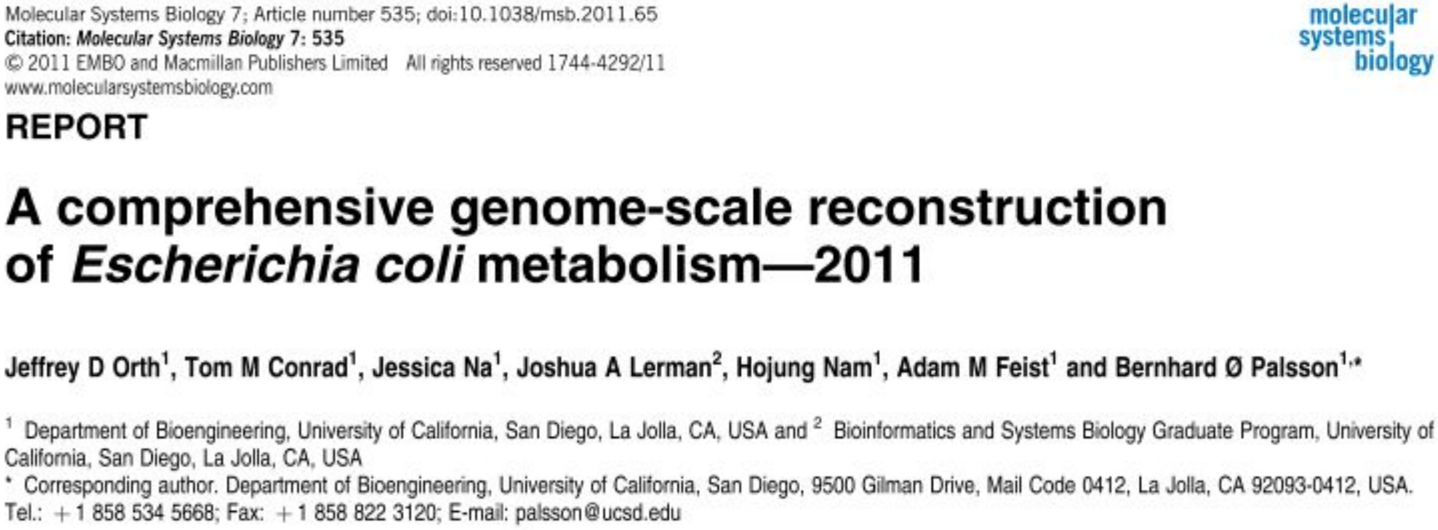
E. coli metabolic model iJO1366

E. coli and Salmonella SBML models are included in cobrapy!
Don’t need to download separately.

```
import cobra.test

#Load the model for genome scale E. coli iJO1366
model = cobra.test.create_test_model("ecoli")
```

Over 2600 different models available:
http://biomodels.caltech.edu/path2models?cat=genome-scale

Reference: iJO1366



Molecular Systems Biology 7; Article number 535; doi:10.1038/msb.2011.65
Citation: ***Molecular Systems Biology* 7: 535**
© 2011 EMBO and Macmillan Publishers Limited All rights reserved 1744-4292/11
www.molecularsystemsbiology.com

molecular systems biology

REPORT

A comprehensive genome-scale reconstruction of *Escherichia coli* metabolism—2011

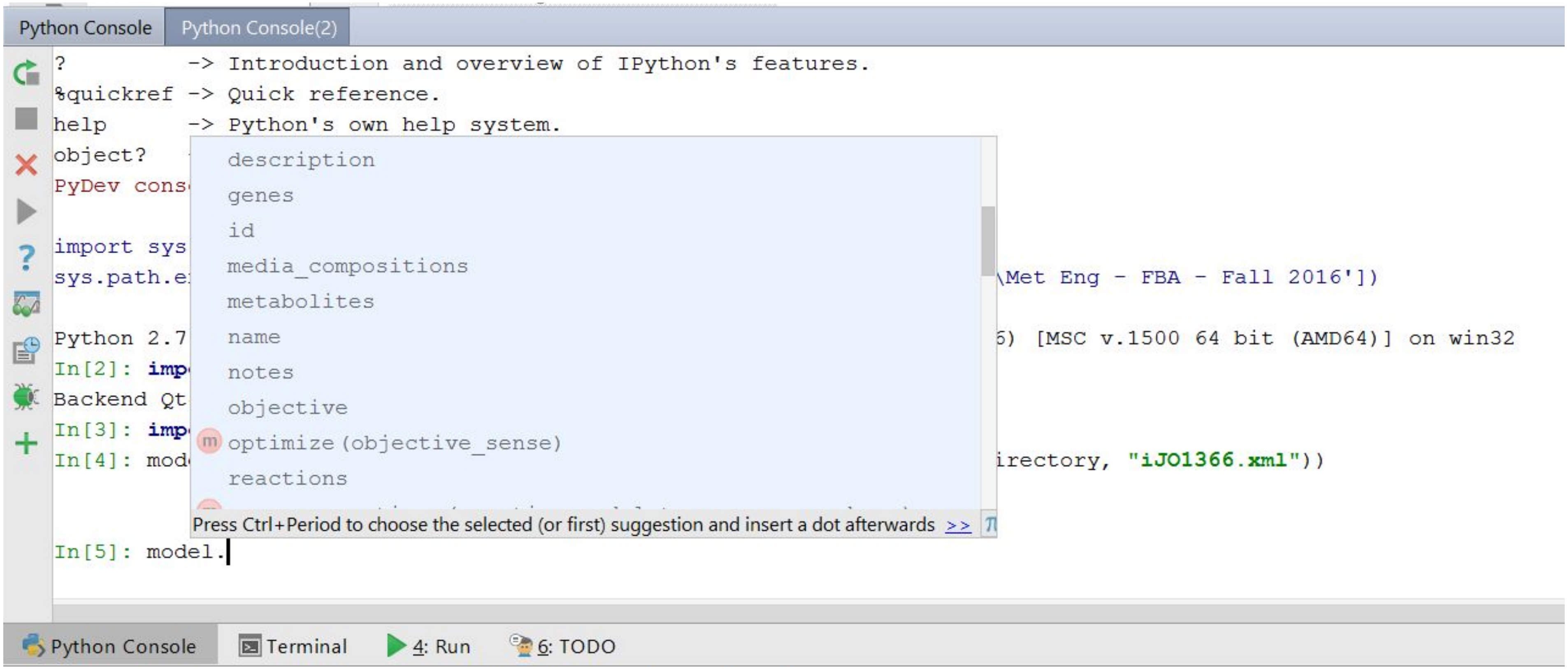
Jeffrey D Orth[1], Tom M Conrad[1], Jessica Na[1], Joshua A Lerman[2], Hojung Nam[1], Adam M Feist[1] and Bernhard Ø Palsson[1,*]

[1] Department of Bioengineering, University of California, San Diego, La Jolla, CA, USA and [2] Bioinformatics and Systems Biology Graduate Program, University of California, San Diego, La Jolla, CA, USA
[*] Corresponding author. Department of Bioengineering, University of California, San Diego, 9500 Gilman Drive, Mail Code 0412, La Jolla, CA 92093-0412, USA. Tel.: +1 858 534 5668; Fax: +1 858 822 3120; E-mail: palsson@ucsd.edu

Supplementary information *has iJO1366 reference file (Excel) – very useful!*

What is in a model?

```python
import cobra.test

#Load the model for genome scale E. coli iJO1366
model = cobra.test.create_test_model("ecoli")
```



```
?         -> Introduction and overview of IPython's features.
%quickref -> Quick reference.
help      -> Python's own help system.
object?   
PyDev cons

import sys
sys.path.e                                                          \Met Eng - FBA - Fall 2016'])

Python 2.7                                                          5) [MSC v.1500 64 bit (AMD64)] on win32
In[2]: imp
Backend Qt
In[3]: imp
In[4]: mod                                                          irectory, "iJO1366.xml"))


In[5]: model.
```

```
description
genes
id
media_compositions
metabolites
name
notes
objective
optimize(objective_sense)
reactions
```

Press Ctrl+Period to choose the selected (or first) suggestion and insert a dot afterwards >>

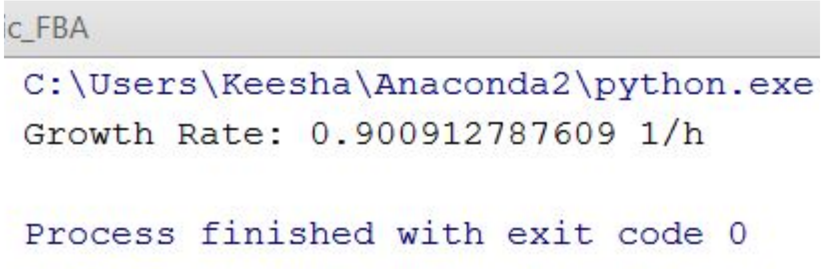
What is in a model?

```
import cobra.test

#Load the model for genome scale E. coli iJO1366
model = cobra.test.create_test_model( "ecoli")


model.reactions[47].id

'EX_ade_e'

model.reactions[47].lower_bound

0.0

model.reactions[47].reaction

'ade_e --> `

model.objective

{<Reaction Ec_biomass_iJO1366_core_53p95M at 0xd5e6748>: 1.0}
```

Basic FBA

```
##This code lets us run Flux Balance Analysis to maximize flux through biomass (growth)
import cobra.test

#Load the model for genome scale E. coli iJO1366
model = cobra.test.create_test_model("ecoli")

#Set the objective to the genome scale biomass reactions
model.reactions.get_by_id("Ec_biomass_iJO1366_core_53p95M").objective_coefficient = 0
model.reactions.get_by_id("Ec_biomass_iJO1366_WT_53p95M").objective_coefficient = 1.0

#Set constrants for aerobic growth in glucose minimal media
model.reactions.get_by_id("EX_glc_e").lower_bound= -10
model.reactions.get_by_id("EX_o2_e").lower_bound = -15

#Solve
solution = model.optimize()

#Output solution
print('Growth Rate: '+str(solution.objective_value)+' 1/h')

# Output more information
model.summary()
```

Solution



```
C:\Users\Keesha\Anaconda2\python.exe
Growth Rate: 0.900912787609 1/h

Process finished with exit code 0
```

1. What happens to the growth rate if uptake of glucose is decreased? Increased?

2. What attributes does the solution have?

Solution

- **f:** the objective value
- **status:** the status from the linear programming solver

Flux through each reaction (mmol/gdcw/hr):
x_dict: a dictionary of {reaction_id: flux_value}.
x: just the values for x_dict
“primal”

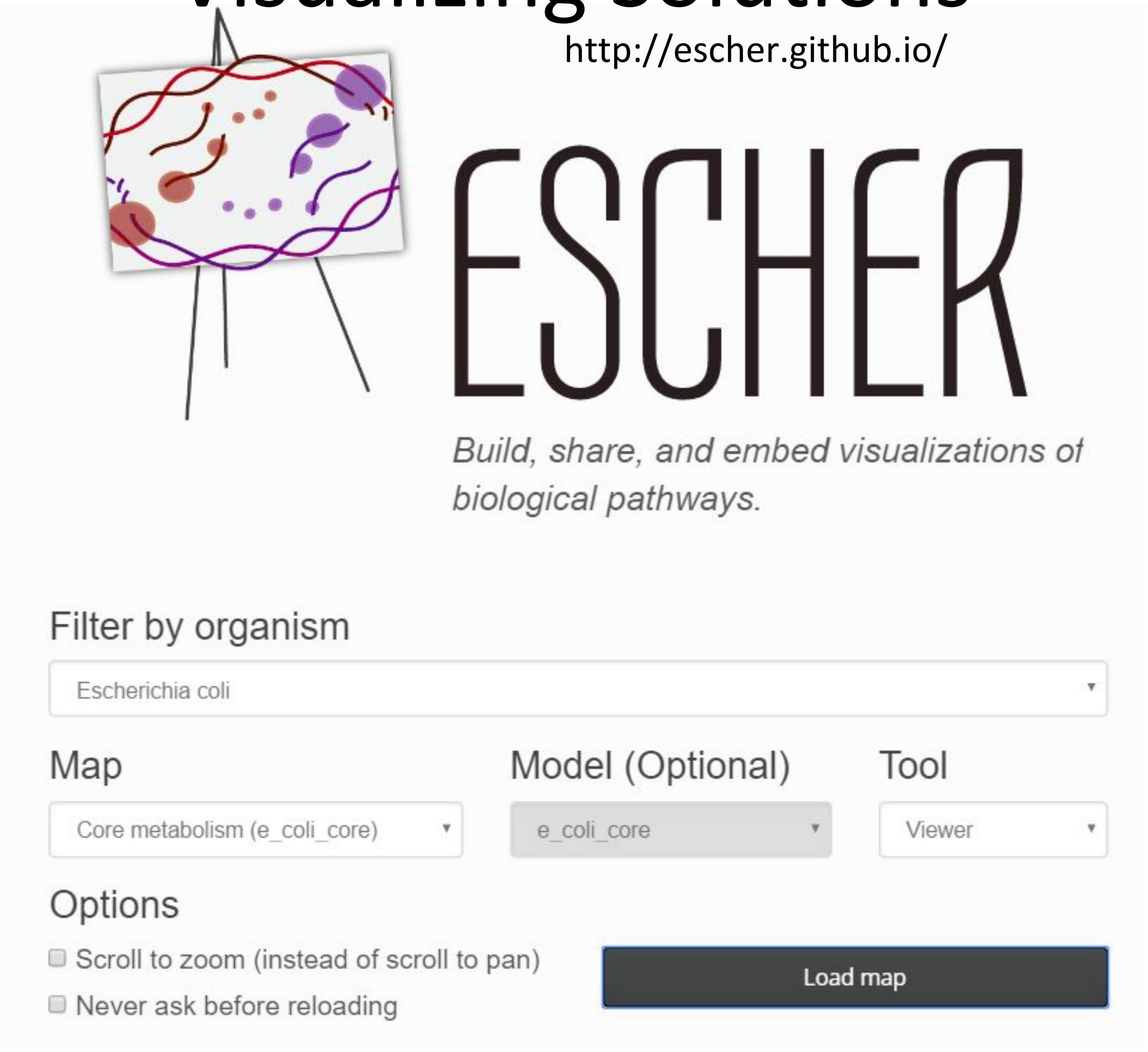
Shadow prices (how much does objective change for unit change in each constraint):

- **y_dict:** a dictionary of {metabolite_id: dual_value}
- **y:** just the values for y_dict

“dual”

Visualizing Solutions

http://escher.github.io/



ESCHER

Build, share, and embed visualizations of biological pathways.

Filter by organism

Escherichia coli ▾

Map: Core metabolism (e_coli_core) ▾

Model (Optional): e_coli_core ▾

Tool: Viewer ▾

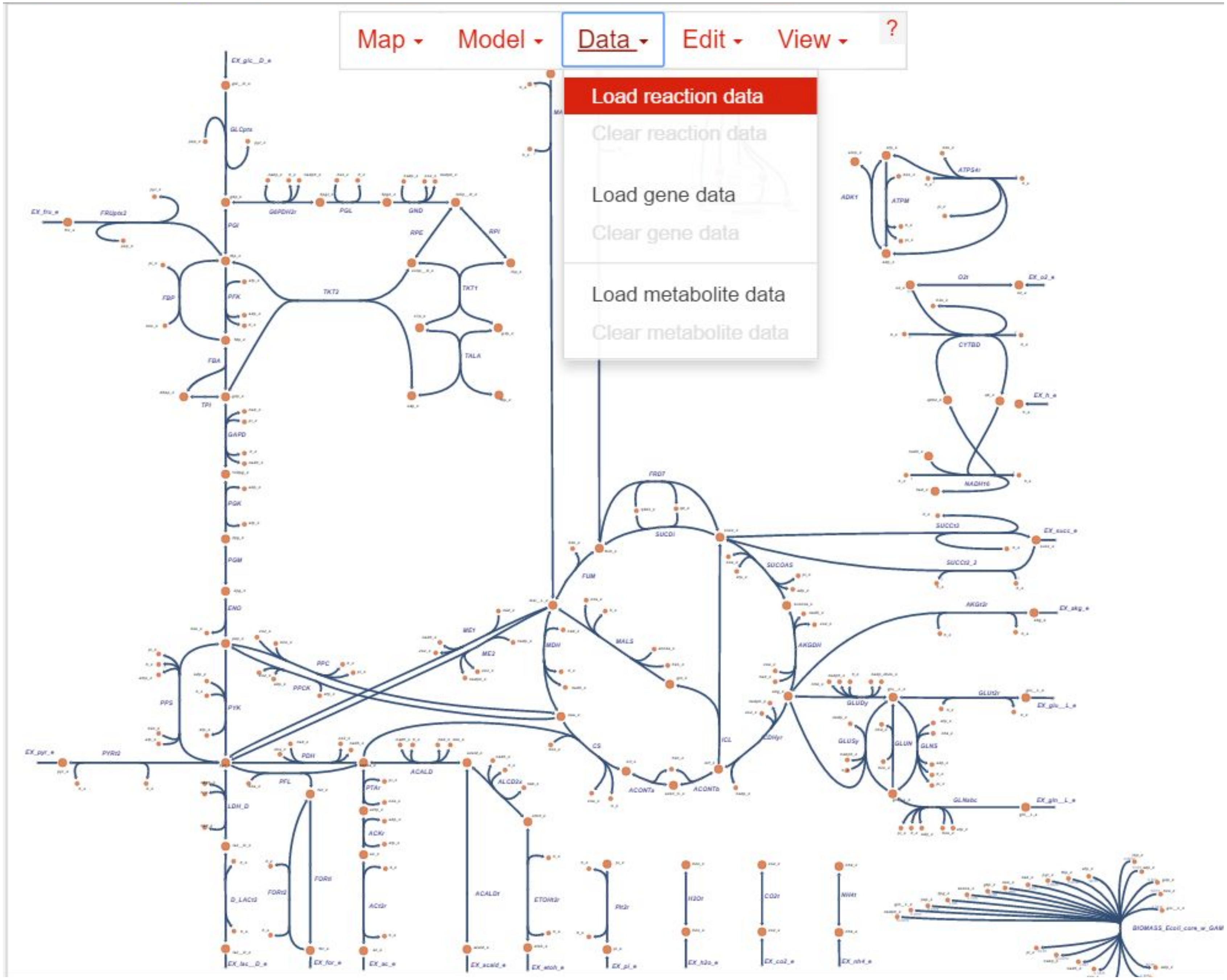
Options

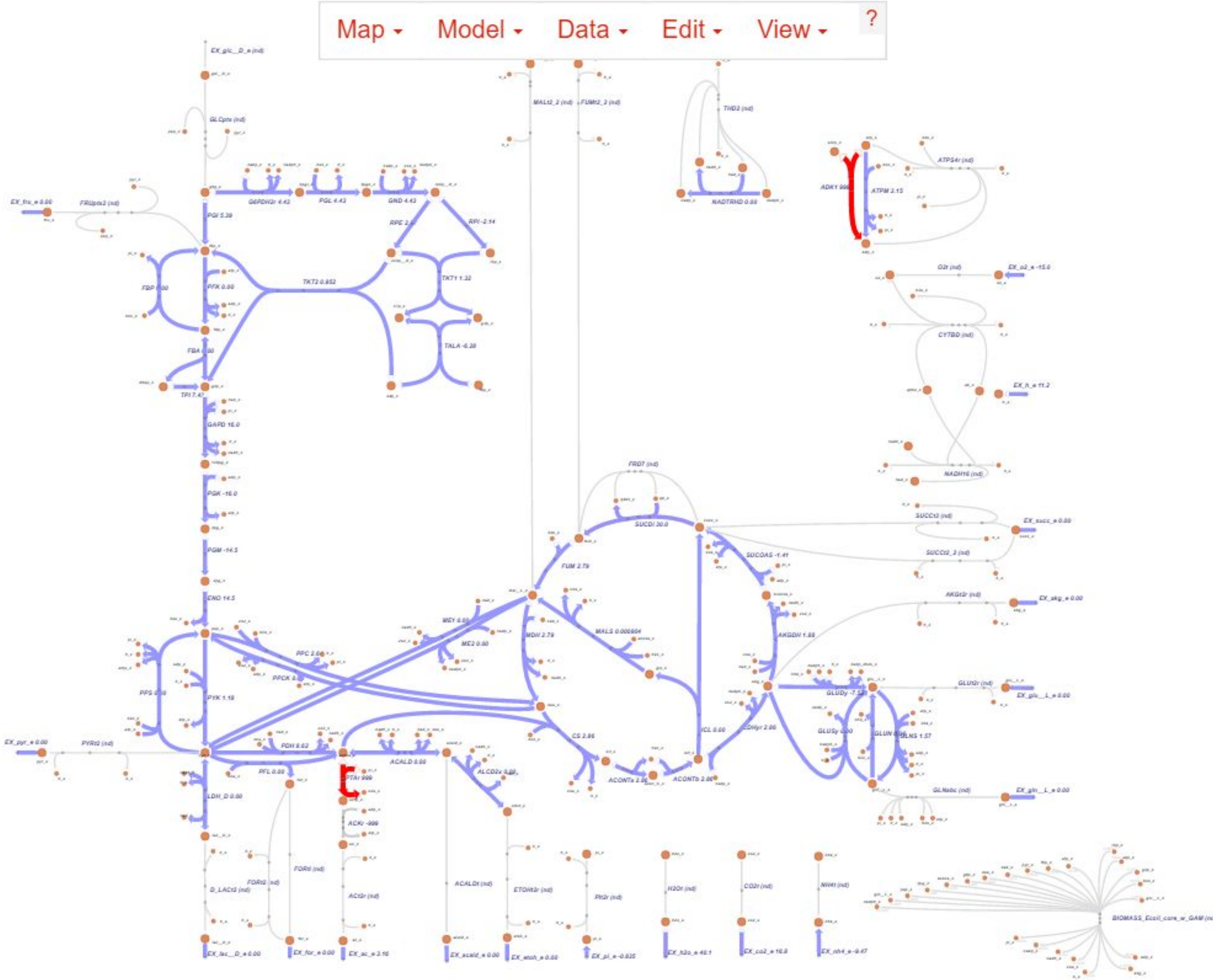
□ Scroll to zoom (instead of scroll to pan)

□ Never ask before reloading

Visualizing Solutions

```python
##This code lets us run Flux Balance Analysis to maximize flux through biomass (growth)
and output a .csv of the flux values in the solution

import cobra.test
import pandas

#Load the model for genome scale E. coli iJO1366
model = cobra.test.create_test_model("ecoli")

#Set the objective to biomass
model.reactions.get_by_id("Ec_biomass_iJO1366_core_53p95M").objective_coefficient = 0
model.reactions.get_by_id("Ec_biomass_iJO1366_WT_53p95M").objective_coefficient = 1.0

#Set constraints for aerobic growth in glucose minimal media
model.reactions.get_by_id("EX_glc_e").lower_bound= -10
model.reactions.get_by_id("EX_o2_e").lower_bound = -15

#Solve
solution = model.optimize() #solution is stored at model.solution

#Output solution
print('Growth Rate: '+str(solution.objective_value)+' 1/h')
df=pandas.DataFrame.from_dict([solution.x_dict]).T
df.to_csv('FBA_max_biomass.csv')
```


Map
Model
Data
Edit
View
?
Load reaction data
Clear reaction data
Load gene data
Clear gene data
Load metabolite data
Clear metabolite data
EX_glc__D_e
GLCpts
PGI
G6PDH2r
PGL
GND
RPE
RPI
FRUpts2
EX_fru_e
FBP
PFK
TKT2
TKT1
TALA
FBA
TPI
GAPD
PGK
PGM
ENO
PPS
PYK
PPC
PPCK
ME1
ME2
MDH
MALS
FUM
FRD7
SUCDi
SUCOAS
AKGDH
ICDHyr
ICL
CS
ACONTa
ACONTb
PYRt2
EX_pyr_e
PDH
PFL
LDH_D
D_LACt2
FORt2
FORti
PTAr
ACKr
ACt2r
ACALD
ALCD2x
ACALDt
ETOHt2r
PIt2r
H2Ot
CO2t
NH4t
ADK1
ATPM
ATPS4r
O2t
EX_o2_e
CYTBD
EX_h_e
NADH16
SUCCt3
SUCCt2_2
EX_succ_e
AKGt2r
EX_akg_e
GLUDy
GLUt2r
EX_glu__L_e
GLUSy
GLUN
GLNS
GLNabc
EX_gln__L_e
EX_lac__D_e
EX_for_e
EX_ac_e
EX_acald_e
EX_etoh_e
EX_pi_e
EX_h2o_e
EX_co2_e
EX_nh4_e
BIOMASS_Ecoli_core_w_GAM


Map
Model
Data
Edit
View
?
EX_glc__D_e (nd)
GLCpts (nd)
EX_fru_e 0.00
FRUpts2 (nd)
G6PDH2r 4.43
PGL 4.43
GND 4.43
PGI 5.39
RPE 2.2
RPI -2.14
FBP 0.00
PFK 0.00
TKT2 0.852
TKT1 1.32
TALA -6.28
FBA 0.00
TPI 7.47
GAPD 16.0
PGK -16.0
PGM -14.5
ENO 14.5
PPS 0.00
PYK 1.18
PPC 2.66
PPCK 0.00
ME1 0.00
ME2 0.00
EX_pyr_e 0.00
PYRt2 (nd)
PDH 8.62
PFL 0.00
ACALD 0.00
ALCD2x 0.00
PTAr 999
ACKr -999
LDH_D 0.00
D_LACt2 (nd)
FORt2 (nd)
FORt (nd)
ACt2r (nd)
ACALDt (nd)
ETOHt2r (nd)
PIt2r (nd)
H2Ot (nd)
CO2t (nd)
NH4t (nd)
EX_lac__D_e 0.00
EX_for_e 0.00
EX_ac_e 3.16
EX_acald_e 0.00
EX_etoh_e 0.00
EX_pi_e -0.835
EX_h2o_e 40.1
EX_co2_e 16.8
EX_nh4_e -9.47
MALt2_2 (nd)
FUMt2_2 (nd)
THD2 (nd)
NADTRHD 0.00
ADK1 999
ATPM 3.15
ATPS4r (nd)
O2t (nd)
EX_o2_e -15.0
CYTBD (nd)
EX_h_e 11.2
NADH16 (nd)
FRD7 (nd)
SUCDi 30.0
FUM 2.79
MDH 2.79
MALS 0.000604
SUCOAS -1.41
AKGDH 1.88
ICDHyr 2.86
ICL 0.00
CS 2.86
ACONTa 2.86
ACONTb 2.86
SUCCt2 (nd)
EX_succ_e 0.00
SUCCt2_2 (nd)
AKGt2r (nd)
EX_akg_e 0.00
GLUDy -7.52
GLUt2r (nd)
EX_glu__L_e 0.00
GLUSy 0.00
GLUN 0.00
GLNS 1.57
GLNabc (nd)
EX_gln__L_e 0.00
BIOMASS_Ecoli_core_w_GAM (nd)

Useful COBRA Functions

Knock out gene or reaction:

```
    model.genes.b4025.knockout()
    model.reactions.PFK.knockout()
```

Change objective of optimization:

```
#Set objective to isopropanol export
model.reactions.get_by_id("Ec_biomass_iJO1366_WT_53p95M").objective_coefficient =
0
model.reactions.get_by_id("EX_2ppoh_e").objective_coefficient = 1.0
```

Access any flux value in the solution:

```
    ##Get any value in the solution
    solution.x_dict.get('EX_glc_e')
```

Change constraints on a reaction:

```
    #ACACCT made reversible
    model.reactions.get_by_id("ACACCT").lower_bound = -1000
```

Useful COBRA Functions

Add reaction:

```
from cobra import Metabolite
co2_c = model.metabolites.get_by_id('co2_c') #CO2
acac_c = model.metabolites.get_by_id('acac_c') #Acetoacetate
#Create new metabolites
acetone_c = Metabolite('acetone_c', formula='C3H6O',
name='Acetone', compartment='c')
```

```
from cobra import Reaction
#add adc:
reaction1 = Reaction('ADC')
reaction1.name = 'Acetoacetate decarboxylase from Clostridium
acetobutylicum'
reaction1.subsystem = 'Isopropanol production'
reaction1.lower_bound = -1000
reaction1.upper_bound = 1000
reaction1.add_metabolites({acac_c: -1.0, co2_c: 1.0,
acetone_c: 1.0})
model.add_reaction(reaction1)
```

Helpful references

E. coli database
https://ecocyc.org

Description of all COBRA functions:
https://cobrapy.readthedocs.io/en/latest/index.html

Escher help:
https://escher.readthedocs.io/en/latest/getting_started.html

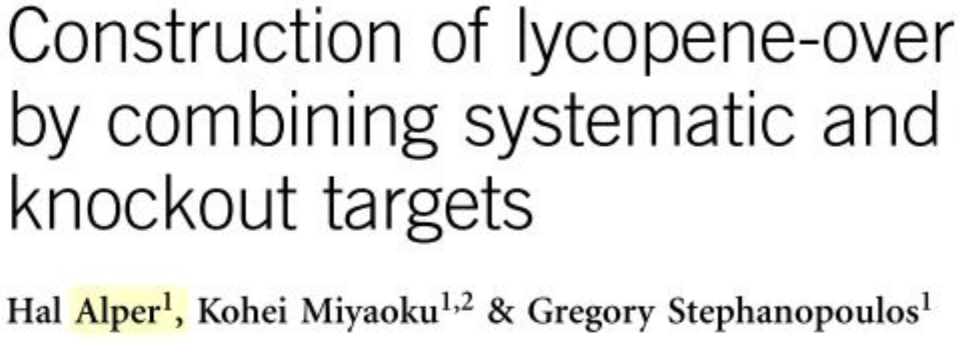
Other SBML models:
http://biomodels.caltech.edu/path2models?cat=genome-scale

FBA tutorial from Orth, Thiele, & Palsson[4]:
http://www.nature.com/nbt/journal/v28/n3/extref/nbt.1614-S1.pdf

LAB

2005

nature biotechnology

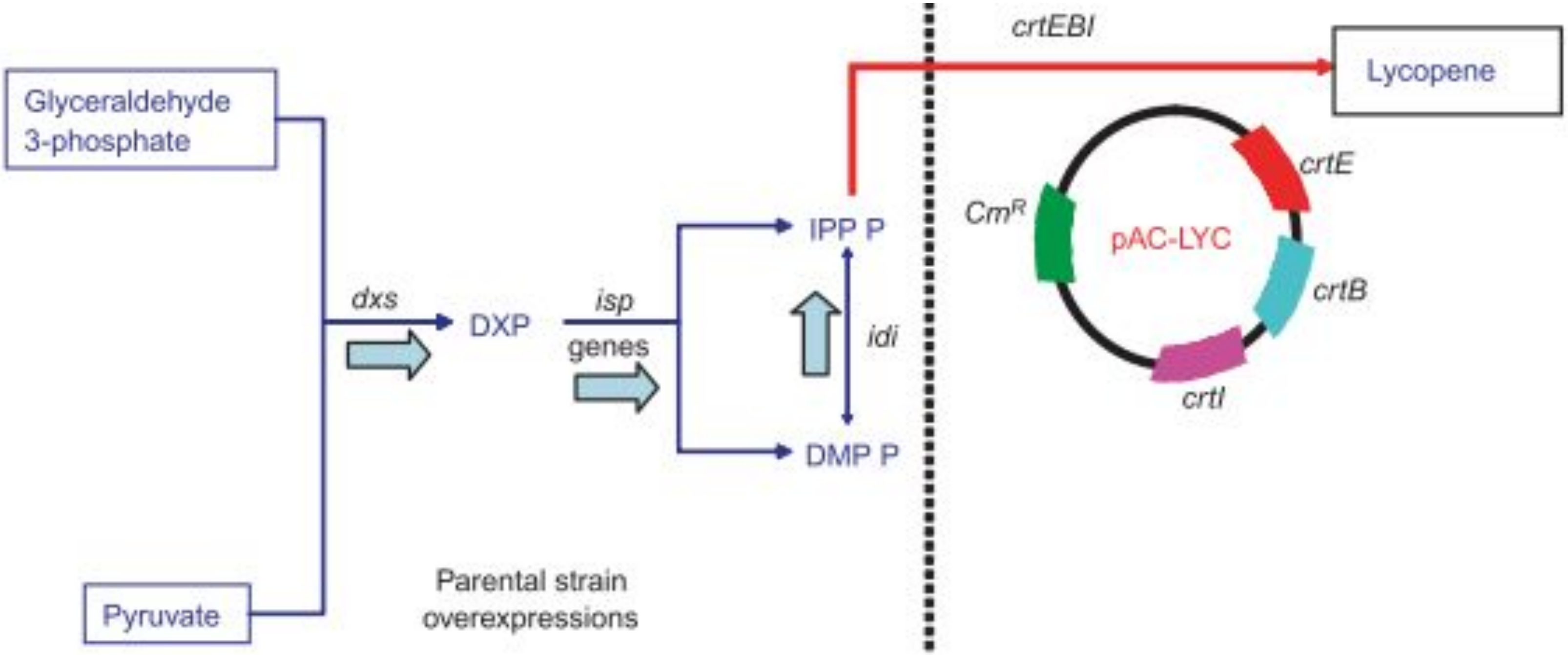


Construction of lycopene-overproducing *E. coli* strains by combining systematic and combinatorial gene knockout targets

Hal Alper[1], Kohei Miyaoku[1,2] & Gregory Stephanopoulos[1]

Also helpful:
Alper et. al. “Identifying gene targets for the metabolic engineering of lycopene biosynthesis in Escherichia coli,” *Metabolic Engineering*, 2005.

Goal: Overproduce lycopene in *E. coli*



- This strain overproduces *dxs*, *idi*, and *ispFD*
- This strain also harbors the pAC-LYC plasmid containing the *crtEBI* operon (pathway for lycopene production)
- Media contains glucose as carbon source
- Conditions are aerobic

Alper et al (2005) *Nature Biotech.*

Set up model for E. coli K12 MG1655

```
##Flux Balance Analysis to simulate Alper et al "Construction of
lycopene-overproducing E. coli.."
## 2005 Nature Biotech

import cobra.test

#Load the model for genome scale E. coli iJO1366
model = cobra.test.create_test_model("ecoli")

#Set constraints for aerobic growth in glucose minimal media
model.reactions.get_by_id("EX_glc_e").lower_bound= -10
model.reactions.get_by_id("EX_o2_e").lower_bound = -15
```

Add genes/reactions for lycopene production

Reactions are supplied from: Alper et. al. “Identifying gene targets for the metabolic engineering of lycopene biosynthesis in Escherichia coli” *Metabolic Engineering*, 2005.

```
#Add crtEBI pathway for lycopene production
#Hint: see Alper et al 2005 Met Eng, Supp Info for reactions
#New metabolites: ggpp_c, phyto_c, lyco_c
from cobra import Metabolite
coa_c = model.metabolites.get_by_id('coa_c')
ipdp_c = model.metabolites.get_by_id('ipdp_c')
frdp_c = model.metabolites.get_by_id('frdp_c')
ppi_c = model.metabolites.get_by_id('ppi_c')
nadp_c = model.metabolites.get_by_id('nadp_c')
nadph_c = model.metabolites.get_by_id('nadph_c')
#Create new metabolites
ggpp_c = Metabolite('ggpp_c', formula='C20H36O7P2', name='Geranylgeranyl
Pyrophospate', compartment='c')
phyto_c = Metabolite('phyto_c', formula='C40H64', name='Phytoene', compartment='c')
lyco_c = Metabolite('lyco_c', formula='C40H56', name='Lycopene', compartment='c')
```

Add genes/reactions for lycopene production

```python
#New reactions: CRTE, CRTB, CRTI, LYCO-dem
from cobra import Reaction
#add CRTE:
reaction1 = Reaction('CRTE')
reaction1.name = 'Geranylgeranyl diphosphate (GGPP) synthase'
reaction1.subsystem = 'Lycopene biosynthesis'
reaction1.lower_bound = 0
reaction1.upper_bound = 1000
reaction1.add_metabolites({ipdp_c: -1.0, frdp_c: -1.0, ggpp_c: 1.0, ppi_c: 1.0})
model.add_reaction(reaction1)
#add CRTB:
reaction2 = Reaction('CRTB')
reaction2.name = 'Phytoene synthase'
reaction2.subsystem = 'Lycopene biosynthesis'
reaction2.lower_bound = 0
reaction2.upper_bound = 1000
reaction2.add_metabolites({ggpp_c: -2.0, phyto_c: 1.0, ppi_c: 1.0})
model.add_reaction(reaction2)
#add CRTI:
reaction3 = Reaction('CRTI')
reaction3.name = 'Phytoene desaturase'
reaction3.subsystem = 'Lycopene biosynthesis'
reaction3.lower_bound = 0
reaction3.upper_bound = 1000
reaction3.add_metabolites({phyto_c: -1.0, nadp_c: -8.0, lyco_c: 1.0, nadph_c: 8.0})
model.add_reaction(reaction3)
#add LYCO-dem:
reaction4 = Reaction('LYCO-dem')
reaction4.name = 'Lycopene demand'
reaction4.subsystem = 'Lycopene biosynthesis'
reaction4.lower_bound = 0
reaction4.upper_bound = 1000
reaction4.add_metabolites({lyco_c: -1.0})
model.add_reaction(reaction4)
```

How much lycopene is produced?

```
#Set the objective to biomass
model.reactions.get_by_id('Ec_biomass_iJO1366_core_53p95M').objective_coefficient = 0
model.reactions.get_by_id('Ec_biomass_iJO1366_WT_53p95M').objective_coefficient = 1.0

#Solve
solution=model.optimize() #solution is stored at model.solution

#Output solution
print('Growth Rate (1/h): ' + str(solution.x_dict.get('Ec_biomass_iJO1366_WT_53p95M')))
print('Lycopene Production Rate (mmol/gdcw/h): ' + str(solution.x_dict.get('LYCO-dem')))
print('Lycopene Yield (mol/mol glucose): ' +
str(-solution.x_dict.get('LYCO-dem')/solution.x_dict.get('EX_glc_e')))
```

```
Growth Rate (1/h): 0.90
Lycopene Production Rate (mmol/gdcw/h): 0.0
Lycopene Yield (mol/mol glucose): 0.0
```

Why do you think that no lycopene is produced??

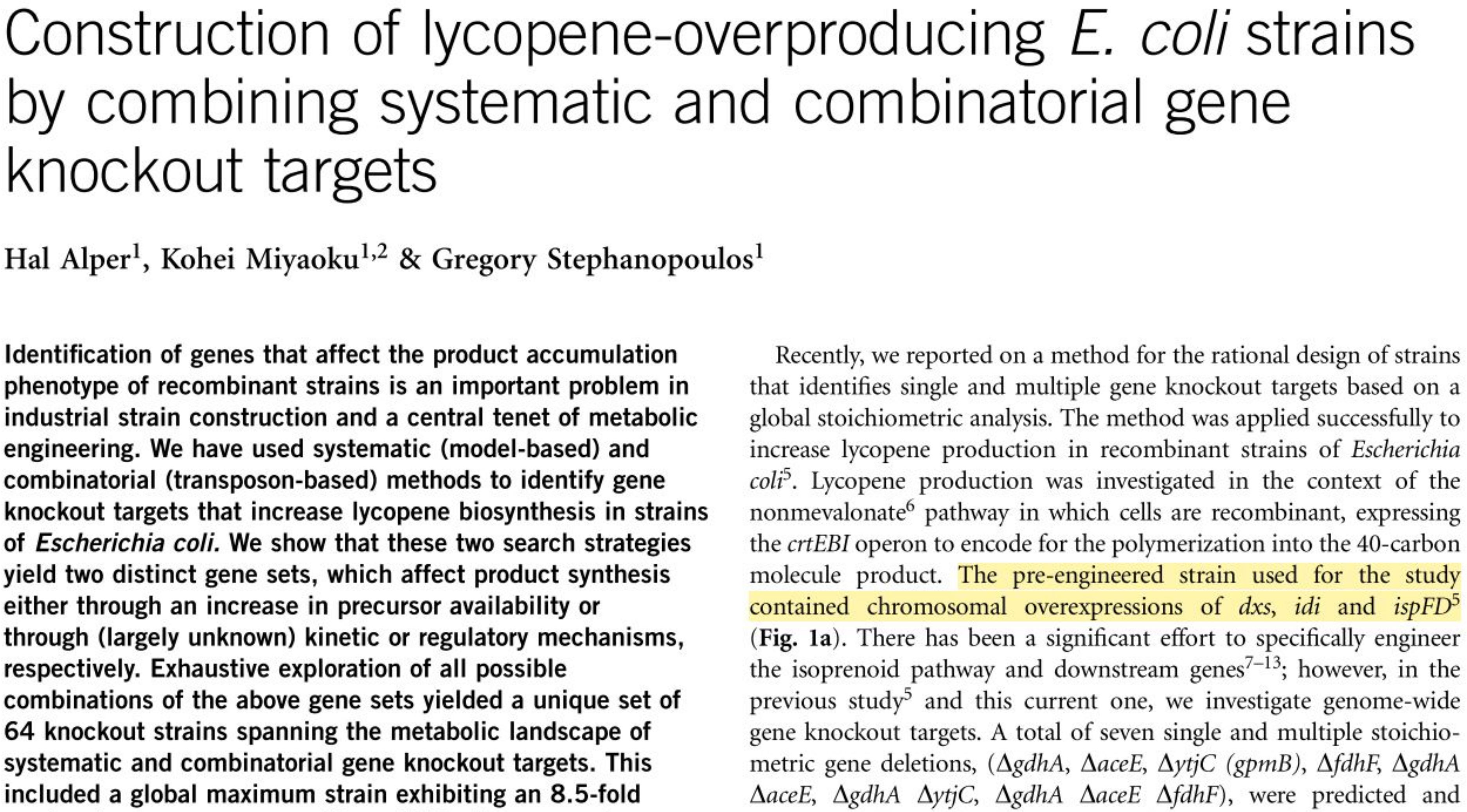
What is the theoretical maximum yield of lycopene in this system?

```
#Set the objective to lycopene production
model.reactions.get_by_id('Ec_biomass_iJO1366_core_53p95M').objective_coefficient = 0
model.reactions.get_by_id('Ec_biomass_iJO1366_WT_53p95M').objective_coefficient = 0
model.reactions.get_by_id('LYCO-dem').objective_coefficient = 1.0

#Solve
solution = model.optimize()
```

```
Growth Rate (1/h): 0.0
Lycopene Production Rate (mmol/gdcw/h): 1.10
Lycopene Yield (mol/mol glucose): 0.11
```

Notice trade-offs between growth rate and lycopene yield

Must engineer the system in order to have lycopene production



Construction of lycopene-overproducing *E. coli* strains by combining systematic and combinatorial gene knockout targets

Hal Alper[1], Kohei Miyaoku[1,2] & Gregory Stephanopoulos[1]

Identification of genes that affect the product accumulation phenotype of recombinant strains is an important problem in industrial strain construction and a central tenet of metabolic engineering. We have used systematic (model-based) and combinatorial (transposon-based) methods to identify gene knockout targets that increase lycopene biosynthesis in strains of *Escherichia coli*. We show that these two search strategies yield two distinct gene sets, which affect product synthesis either through an increase in precursor availability or through (largely unknown) kinetic or regulatory mechanisms, respectively. Exhaustive exploration of all possible combinations of the above gene sets yielded a unique set of 64 knockout strains spanning the metabolic landscape of systematic and combinatorial gene knockout targets. This included a global maximum strain exhibiting an 8.5-fold

Recently, we reported on a method for the rational design of strains that identifies single and multiple gene knockout targets based on a global stoichiometric analysis. The method was applied successfully to increase lycopene production in recombinant strains of *Escherichia coli*[5]. Lycopene production was investigated in the context of the nonmevalonate[6] pathway in which cells are recombinant, expressing the *crtEBI* operon to encode for the polymerization into the 40-carbon molecule product. The pre-engineered strain used for the study contained chromosomal overexpressions of *dxs*, *idi* and *ispFD*[5] (**Fig. 1a**). There has been a significant effort to specifically engineer the isoprenoid pathway and downstream genes[7–13]; however, in the previous study[5] and this current one, we investigate genome-wide gene knockout targets. A total of seven single and multiple stoichiometric gene deletions, (Δ*gdhA*, Δ*aceE*, Δ*ytjC (gpmB)*, Δ*fdhF*, Δ*gdhA* Δ*aceE*, Δ*gdhA* Δ*ytjC*, Δ*gdhA* Δ*aceE* Δ*fdhF*), were predicted and

Overexpress genes as specified *dxs*, *idi*, & *ispFD*

Can look up each gene in the iJO1366 model Excel reference (download from supplement of Orth et. al. 2011 *MSB*) to figure out what the corresponding reaction is named

We can enforce overexpression by adding a constraint on the lower bound, but what should that constraint be?

Output values in current optimal solution with:

```
print 'Flux in original solution:'
print('DXPS: ', solution.x_dict.get('DXPS'))
```

Or perform FVA to see a range of possible values that optimize the objective function:

```
from cobra.flux_analysis import flux_variability_analysis

reactions OE = [model.reactions.DXPS, model.reactions.IPDDI, model.reactions.MECDPS,
model.reactions.MEPCT]
fva = flux variability analysis(model, reaction_list = reactions_OE,
fraction_of_optimum=0.9)

print fva
```

FVA results for genes to be overexpressed

```
         minimum   maximum
DXPS    0.005734  1.588167
IPDDI  -1.191429  0.396376
MECDPS  0.005373  1.587805
MEPCT   0.005373  1.587805
```

Overexpress genes as specified *dxs*, *idi*, & *ispFD*

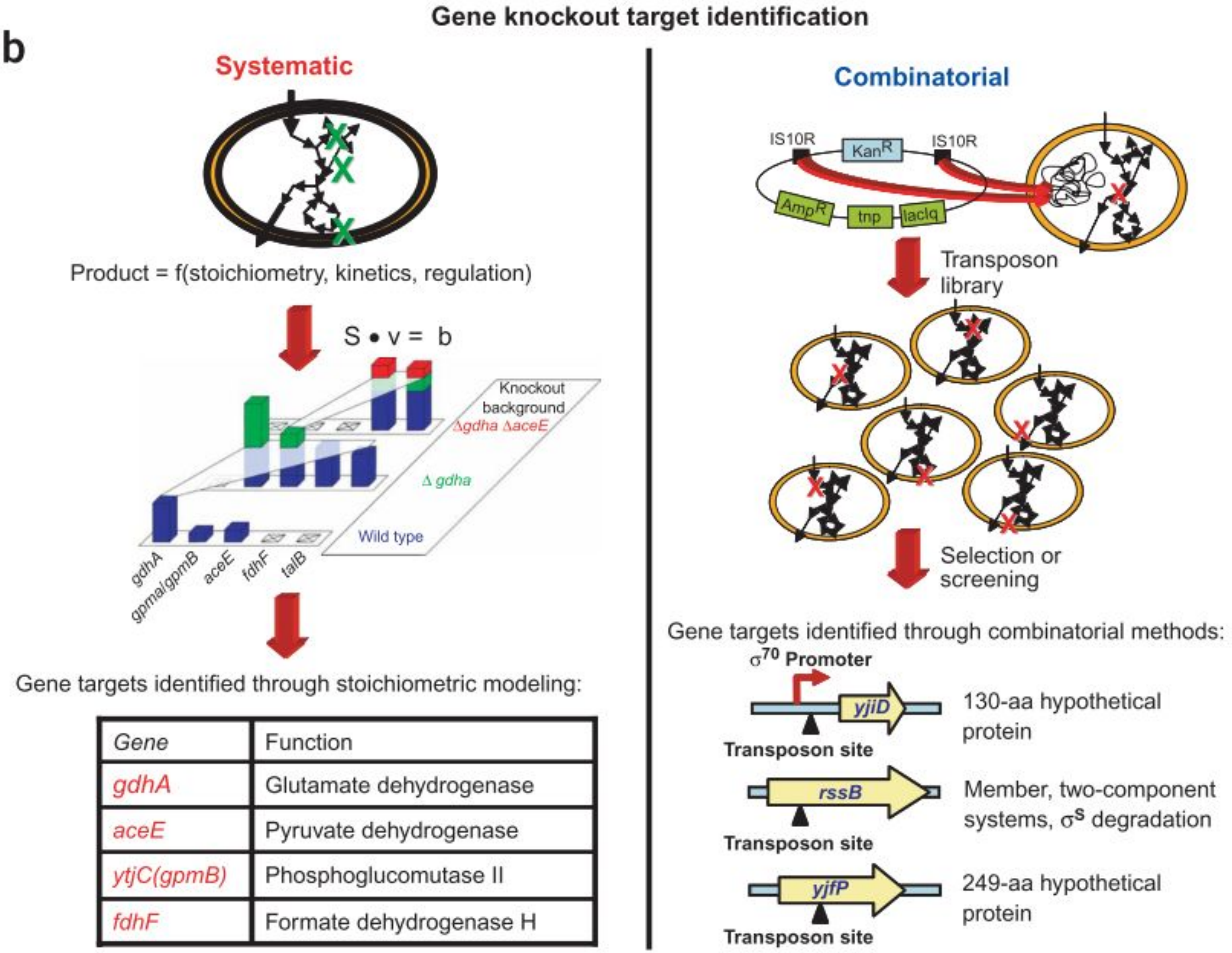
Can look up each gene in the iJO1366 model Excel reference (download from supplement of Orth et. al. 2011 *MSB*) to figure out what the corresponding reaction is named

```
#Overexpress dxs, idi, ispFD
model.reactions.get_by_id('DXPS').lower_bound = 2
model.reactions.get_by_id('IPDDI').lower_bound = 0.5
model.reactions.get_by_id('MECDPS').lower_bound = 2
model.reactions.get_by_id('MEPCT').lower_bound = 2
```

Even with biomass as the objective, we now see lycopene produced:

Growth Rate (1/h): 0.76
Lycopene Production Rate (mmol/gdcw/h): 0.2496
Lycopene Yield (mol/mol glucose): 0.02496

Introduce gene knockouts



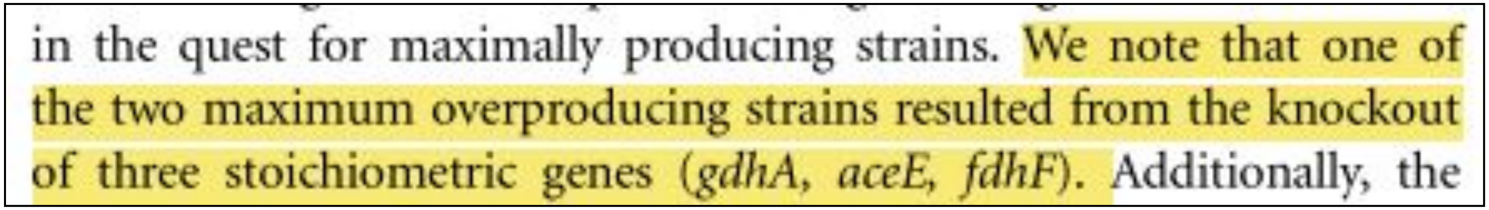
Gene targets identified through stoichiometric modeling:

Gene	Function
gdhA	Glutamate dehydrogenase
aceE	Pyruvate dehydrogenase
ytjC(gpmB)	Phosphoglucomutase II
fdhF	Formate dehydrogenase H

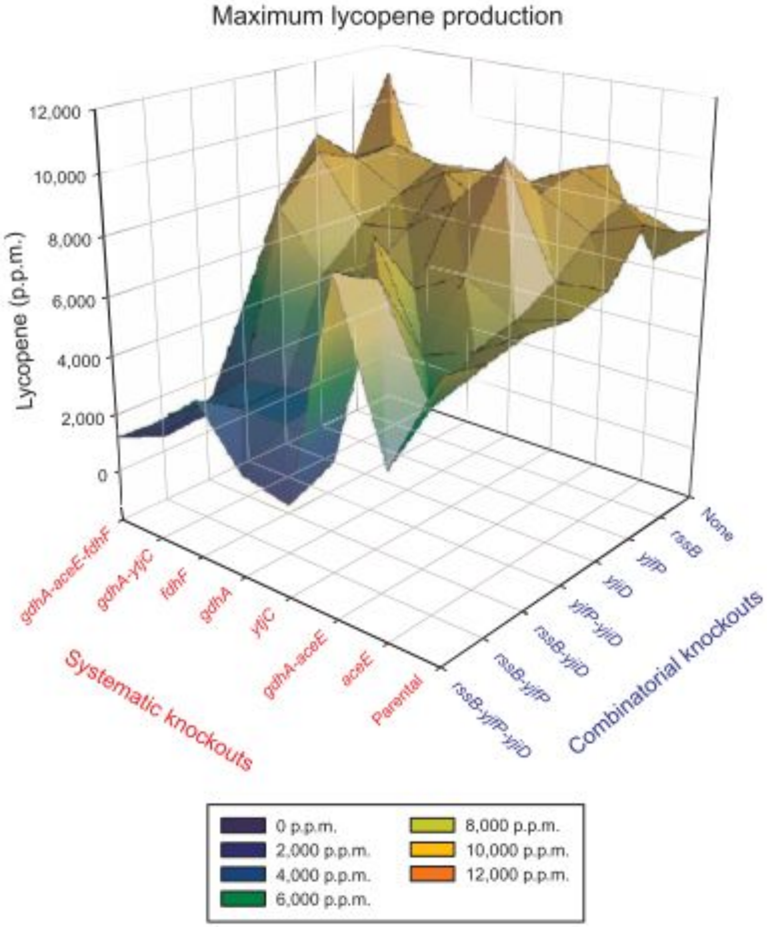
Gene targets identified through combinatorial methods:

Gene	Function
yjiD	130-aa hypothetical protein
rssB	Member, two-component systems, σ^S degradation
yjfP	249-aa hypothetical protein

Compare to experimental findings



in the quest for maximally producing strains. We note that one of the two maximum overproducing strains resulted from the knockout of three stoichiometric genes (*gdhA*, *aceE*, *fdhF*). Additionally, the



Introduce gene knockouts

```
#Knockout genes gdhA, aceE,
ytjC(gpmB), fdhF (yjjD, rssB, yjfP
aren't in model)
model.genes.b1761.knock out()  # gdhA
model.genes.b0114.knock out()  # aceA
model.genes.b4395.knock out()  # ytjC
model.genes.b4079.knock_out()  # fdhF
```