Entropic flux balance analysis: trade off between entropy maximisation and additional information

Author(s): Samira Ranjbar, University of Galway

Reviewer(s): Ronan Fleming, University of Galway

In the context of FBA, an objective function is a mathematical representation of a cellular process that the model attempts to optimize. In entropic flux balance analysis [1], eFBA, the objective is to to maximize the Shannon entropy of the flux distribution, subject to constraints ensuring the feasibility and thermodynamic consistency of the fluxes. Shannon entropy measures the degree of disorder or randomness in a system. In entropic Flux Balance Analysis, maximizing the unormalised entropy of the (separate forward and reverse) flux distribution. This corresponds to the least biased distribution, given the constraints [1]. This tutorial builds on tutorial_entropicFluxBalanceAnalysis and examines the ways in which one can (a) trade off between maximisation of a linear combination of net fluxes with maximisation of unidirectional flux entropy, (b) how to set this trade off so that growth rate matches experimentally measured growth rate, and (c) how omics data, e.g., gene expression data, can be used to bias the magnitude of internal net flux.

Toy model

```
reactionFormulas = {'4 A  -> 4 B',...
    '2 A  -> 2 B',...
    'A  -> B',...
    'A  <=> ',...
    'B  <=> '};
reactionNames = {'R1', 'R2', 'R3', 'R4', 'R5'};
lowerBounds = [  -10,   -10,   -10, -10, 10];
upperBounds = [ 10,  10,  10, -10, 10];
model = createModel(reactionNames, reactionNames, reactionFormulas,...
                            'lowerBoundList', lowerBounds,
'upperBoundList', upperBounds);
```

```
Adding the following Metabolites to the model:
A[c]
B[c]
addMultipleReactions: Adding the following reactions to the model:
R1    4 A[c]     <=>    4 B[c]
R2    2 A[c]     <=>    2 B[c]
R3    A[c]     <=>    B[c]
R4    A[c]     <=>
R5    B[c]     ->
```

```
initCobraToolbox;
```

```
     _____   _____   _____   _____     _____      |
    /  ___| /  _  \ |  _  \ |  _  \   / ___ \     |   COnstraint-Based Reconstruction and Analysis
    | |     | | | | | |_| | | |_| |  | |___| |    |   The COBRA Toolbox - 2024
```

```
     | |     | | | | |  _  { |  _  /  |  ___  |   |
     | |___  | |_| | | |_| | | | \ \  | |   | |   |   Documentation:
     \_____| \_____/ |_____/ |_|  \_\ |_|   |_|   |   http://opencobra.github.io/cobratoolbox
                                                 |
 > Checking if git is installed ...  Done (version: 2.25.1).
 > Checking if the repository is tracked using git ...  Done.
 > Checking if curl is installed ...  Done.
 > Checking if remote can be reached ...  Done.
 > Initializing and updating submodules (this may take a while)... Done.
 > Adding all the files of The COBRA Toolbox ...  Done.
 > Define CB map output... set to svg.
 > TranslateSBML is installed and working properly.
 > Configuring solver environment variables ...
   - [*---] ILOG_CPLEX_PATH: /opt/ibm/ILOG/CPLEX_Studio1210/cplex/matlab/x86-64_linux
   - [*---] GUROBI_PATH: /opt/gurobi1003/linux64/matlab
   - [----] TOMLAB_PATH: --> set this path manually after installing the solver ( see instructions )
   - [*---] MOSEK_PATH: /opt/mosek/10.1/
   Done.
 > Checking available solvers and solver interfaces ...     0

     0

Check osense*c - A'*lam - w = 0 (stationarity):
     0
     0


 > [gurobi] Primal optimality condition in solveCobraLP satisfied.
 > [gurobi] Dual optimality condition in solveCobraLP satisfied.
Version identifier: 12.10.0.0 | 2019-11-26 | 843d4de
CPXPARAM_Output_WriteLevel                       3
CPXPARAM_Output_CloneLog                         -1
Found incumbent of value 0.000000 after 0.00 sec. (0.00 ticks)

Root node processing (before b&c):
  Real time             =    0.00 sec. (0.00 ticks)
Parallel b&c, 12 threads:
  Real time             =    0.00 sec. (0.00 ticks)
  Sync time (average)   =    0.00 sec.
  Wait time (average)   =    0.00 sec.
                          ------------
Total (root+branch&cut) =    0.00 sec. (0.00 ticks)
Could not find installation of tomlab_cplex, so it cannot be tested
GLPK Simplex Optimizer, v4.42
1 row, 2 columns, 1 non-zero
Preprocessing...
~     0: obj =   0.000000000e+00  infeas =  0.000e+00
OPTIMAL SOLUTION FOUND BY LP PREPROCESSOR

 > [glpk] Primal optimality condition in solveCobraLP satisfied.
 > [mosek] Primal optimality condition in solveCobraLP satisfied.
 > [mosek] Dual optimality condition in solveCobraLP satisfied.
Could not find installation of matlab, so it cannot be tested

   ------------------------------------------------------
   pdco.m                     Version pdco5 of 15 Jun 2018
   Primal-dual barrier method to minimize a convex function
   subject to linear constraints Ax + r = b,  bl <= x <= bu

   Michael Saunders       SOL and ICME, Stanford University
   Contributors:     Byunggyoo Kim (SOL), Chris Maes (ICME)
                     Santiago Akle (ICME), Matt Zahr (ICME)
                     Aekaansh Verma (ME)
```

```
   ---------------------------------------------------------

The objective is linear
The matrix A is an explicit sparse matrix

m        =        1     n        =        2      nnz(A)  =        1
max |b | =        0     max |x0| =  1.0e+00      xsize   =  1.0e+00
max |y0| =        1     max |z0| =  1.0e+00      zsize   =  1.0e+00

x0min    =        1     featol   =  1.0e-06      d1max   =  1.0e-04
z0min    =        1     opttol   =  1.0e-06      d2max   =  5.0e-04
mu0      =  1.0e-01     steptol  =     0.99     bigcenter=     1000

LSMR/MINRES:
atol1    =  1.0e-10     atol2    =  1.0e-15      btol    =  0.0e+00
conlim   =  1.0e+12     itnlim   =       10      show    =        0

Method   =        2     (1 or 11=chol  2 or 12=QR  3 or 13=LSMR  4 or 14=MINRES 21=SQD(LU)  22=SQD(MA57))
Eliminating dy before dx


Bounds:
  [0,inf]  [-inf,0]  Finite bl  Finite bu  Two bnds   Fixed    Free
        0         0          0          0         0       2       0
  [0, bu]  [bl,  0]  excluding fixed variables
        0         0

Itn   mu stepx stepz  Pinf  Dinf  Cinf   Objective    nf  center       QR
  0                  -6.6 -99.0  -Inf  1.2500000e-07        1.0
  1 -1.0 1.000 1.000 -99.0 -99.0  -Inf  0.0000000e+00  1     1.0        1
  2 -3.0 1.000 1.000 -99.0 -99.0  -Inf  0.0000000e+00  1     1.0
  3 -5.0 1.000 1.000 -99.0 -99.0  -Inf  0.0000000e+00  1     1.0
  4 -7.0 1.000 1.000 -99.0 -99.0  -Inf  0.0000000e+00  1     1.0
   Converged

max |x| =     0.000    max |y| =     0.000    max |z| =     0.000   scaled
max |x| =     0.000    max |y| =     0.000    max |z| =     0.000 unscaled
max |x| and max |z| exclude fixed variables
PDitns  =         4     QRitns =         0    cputime =       0.0

Distribution of vector     x         z
[      1,     10 )         0         2
[    0.1,      1 )         0         0
[   0.01,    0.1 )         0         0
[  0.001,   0.01 )         0         0
[ 0.0001,  0.001 )         0         0
[  1e-05, 0.0001 )         0         0
[  1e-06,  1e-05 )         0         0
[  1e-07,  1e-06 )         0         0
[  1e-08,  1e-07 )         0         0
[      0,  1e-08 )         2         0
Elapsed time is 0.021359 seconds.

 > [pdco] Primal optimality condition in solveCobraLP satisfied.
 > [pdco] Dual optimality condition in solveCobraLP satisfied.
Could not find installation of quadMinos, so it cannot be tested
Could not find installation of dqqMinos, so it cannot be tested
Could not find installation of cplex_direct, so it cannot be tested

 > [cplexlp] Primal optimality condition in solveCobraLP satisfied.
 > [cplexlp] Dual optimality condition in solveCobraLP satisfied.
Could not find installation of tomlab_snopt, so it cannot be tested
 Done.
 > Setting default solvers ...Could not find installation of matlab, so it cannot be tested
```

```
Done.
> Saving the MATLAB path ... Done.
  - The MATLAB path was saved as ~/pathdef.m.

> Summary of available solvers and solver interfaces

          Support        LP      MILP        QP      MIQP       NLP        EP
   ------------------------------------------------------------------------------
   gurobi           active               1         1         1         1         -         -
   ibm_cplex        active               1         1         1         1         -         -
   tomlab_cplex     active               0         0         0         0         -         -
   glpk             active               1         1         -         -         -         -
   mosek            active               1         -         1         -         -         1
   matlab           active               0         -         -         -         0         -
   pdco             active               1         -         1         -         -         1
   quadMinos        active               0         -         -         -         -         -
   dqqMinos         active               0         -         0         -         -         -
   cplex_direct     active               0         0         0         -         -         -
   cplexlp          active               1         -         -         -         -         -
   qpng             passive              -         -         1         -         -         -
   tomlab_snopt     passive              -         -         -         -         0         -
   lp_solve         legacy               1         -         -         -         -         -
   ------------------------------------------------------------------------------
   Total            -                    7         3         5         2         0         2

+ Legend: - = not applicable, 0 = solver not compatible or not installed, 1 = solver installed.


> You can solve LP problems using: 'gurobi' - 'ibm_cplex' - 'glpk' - 'mosek' - 'pdco' - 'cplexlp'
> You can solve MILP problems using: 'gurobi' - 'ibm_cplex' - 'glpk'
> You can solve QP problems using: 'gurobi' - 'ibm_cplex' - 'mosek' - 'pdco'
> You can solve MIQP problems using: 'gurobi' - 'ibm_cplex'
> You can solve NLP problems using:
> You can solve EP problems using: 'mosek' - 'pdco'

> Checking for available updates ...
> You cannot update your fork using updateCobraToolbox() because this is a development branch.
> The current branch is: master
> The last commit to the current branch is: 097dc1
> You can use MATLAB.devTools (https://github.com/opencobra/MATLAB.devTools) to update your fork.
removing: /home/samira/fork-cobratoolbox/src/analysis/thermo/componentContribution/new
removing: /home/samira/fork-cobratoolbox/src/analysis/thermo/groupContribution/new
removing: /home/samira/fork-cobratoolbox/src/analysis/thermo/inchi/new
removing: /home/samira/fork-cobratoolbox/src/analysis/thermo/molFiles/new
removing: /home/samira/fork-cobratoolbox/src/analysis/thermo/protons/new
removing: /home/samira/fork-cobratoolbox/src/analysis/thermo/trainingModel/new
```

```
FBAsolution = optimizeCbModel(model,'max');
```

The flux distribution is shown in figure 1:

Figure1: Flux distribution calculated using FBA in model1.

As you can see, reactions 'R1' and 'R2' or 'R1' and 'R3' have created a cycle within the network. A cycle means the system is not following the laws of thermodynamics and, therefore, is not occurring in reality. lets look at eFBA results:

```
param.solver = 'mosek';
param = mosekParamSetEFBA(param);
[solution,~] = entropicFluxBalanceAnalysis(model,param);
```

And illustration using escher is shown in figure2:

Figure2: Flux distribution calculated using eFBA in model1.

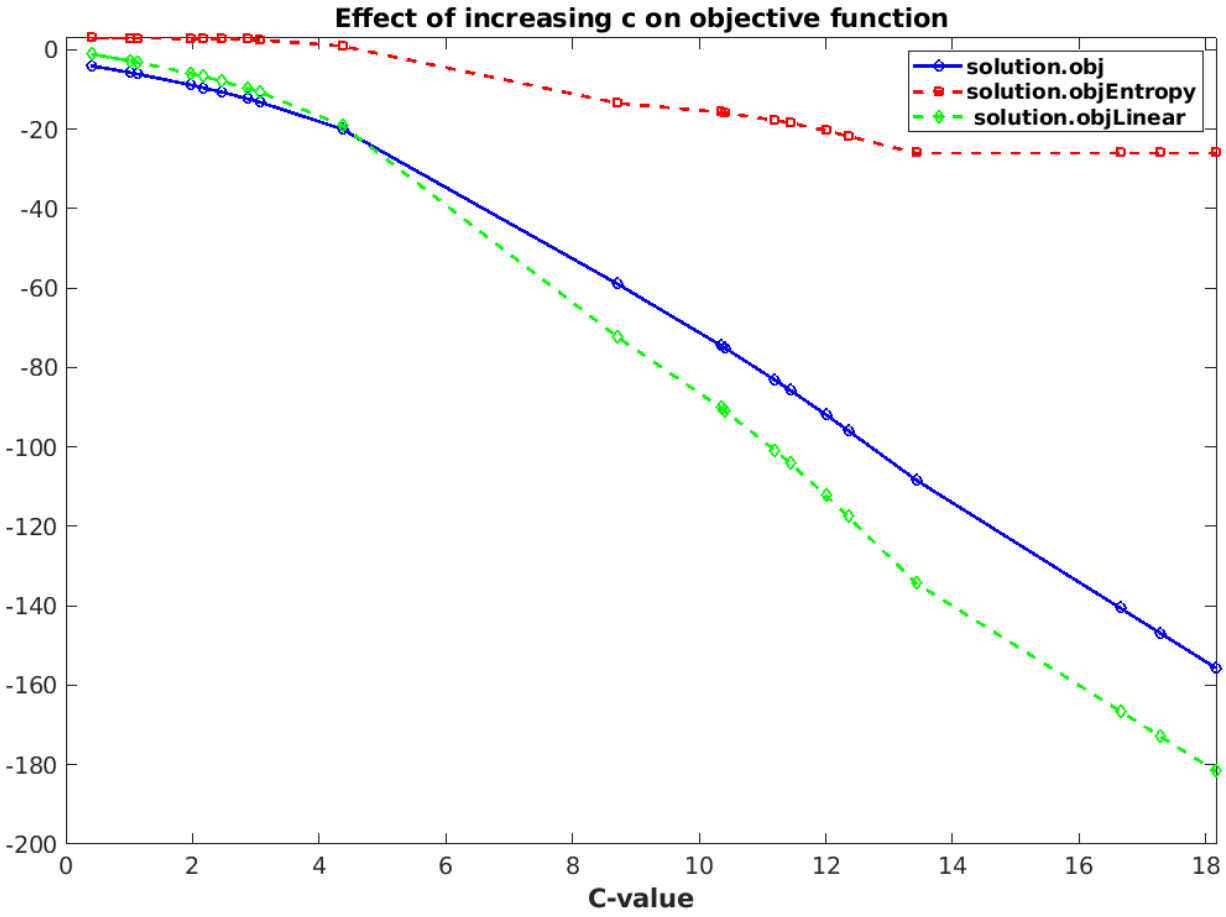
```
plotyy_eFBA(model,'R1',20,20,1);
```

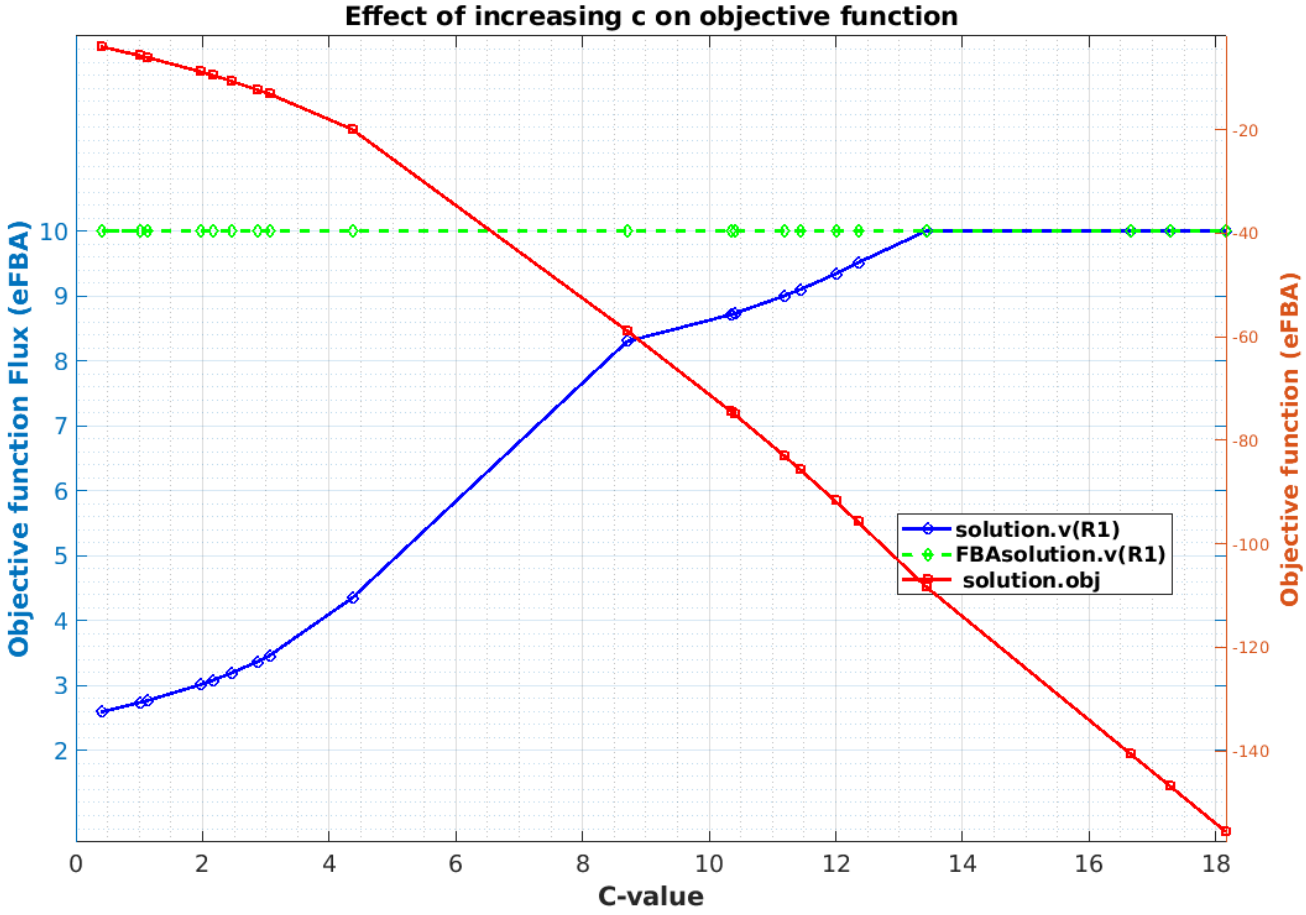
```
 _____   _____   _____   _____     _____
/  ___| /  _  \ |  _  \ |  _  \   / ___ \
| |     | | | | | |_| | | |_| |  | |___| |
| |     | | | | |  _  { |  _  /  |  ___  |
| |___  | |_| | | |_| | | | \ \  | |   | |
\_____| \_____/ |_____/ |_|  \_\ |_|   |_|

COnstraint-Based Reconstruction and Analysis
The COBRA Toolbox - 2024

Documentation:
http://opencobra.github.io/cobratoolbox

> Checking if git is installed ...  Done (version: 2.25.1).
> Checking if the repository is tracked using git ...  Done.
> Checking if curl is installed ...  Done.
> Checking if remote can be reached ...  Done.
> Initializing and updating submodules (this may take a while)... Done.
> Adding all the files of The COBRA Toolbox ...  Done.
> Define CB map output... set to svg.
```

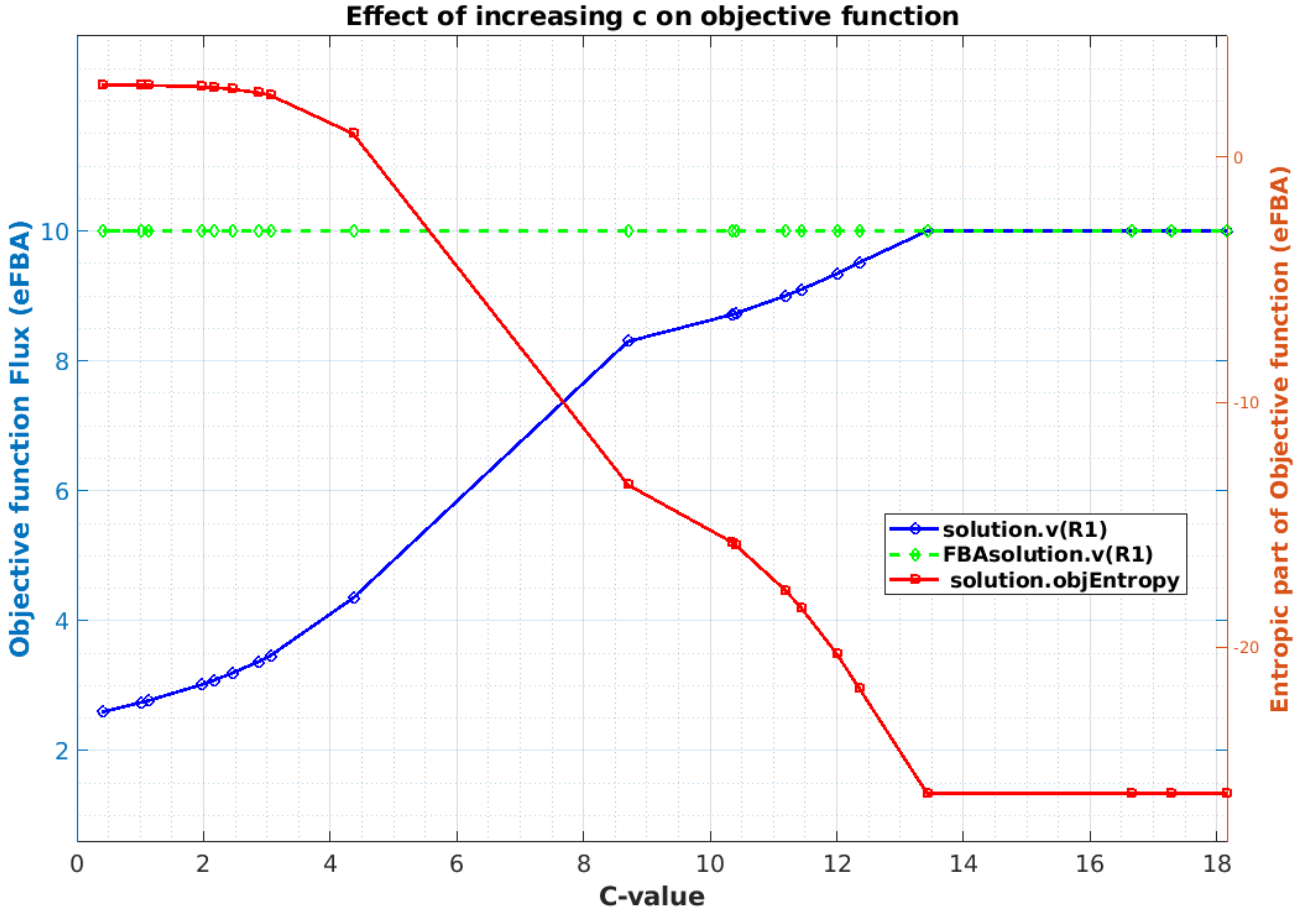
```
 > TranslateSBML is installed and working properly.
 > Configuring solver environment variables ...
   - [*---] ILOG_CPLEX_PATH: /opt/ibm/ILOG/CPLEX_Studio1210/cplex/matlab/x86-64_linux
   - [*---] GUROBI_PATH: /opt/gurobi1003/linux64/matlab
   - [----] TOMLAB_PATH: --> set this path manually after installing the solver ( see instructions )
   - [*---] MOSEK_PATH: /opt/mosek/10.1/
   Done.
 > Checking available solvers and solver interfaces ...     0

     0

Check osense*c - A'*lam - w = 0 (stationarity):
     0
     0


 > [gurobi] Primal optimality condition in solveCobraLP satisfied.
 > [gurobi] Dual optimality condition in solveCobraLP satisfied.
Version identifier: 12.10.0.0 | 2019-11-26 | 843d4de
CPXPARAM_Output_WriteLevel                       3
CPXPARAM_Output_CloneLog                         -1
Found incumbent of value 0.000000 after 0.00 sec. (0.00 ticks)

Root node processing (before b&c):
  Real time             =    0.00 sec. (0.00 ticks)
Parallel b&c, 12 threads:
  Real time             =    0.00 sec. (0.00 ticks)
  Sync time (average)   =    0.00 sec.
  Wait time (average)   =    0.00 sec.
                          ------------
Total (root+branch&cut) =    0.00 sec. (0.00 ticks)
Could not find installation of tomlab_cplex, so it cannot be tested
GLPK Simplex Optimizer, v4.42
1 row, 2 columns, 1 non-zero
Preprocessing...
~     0: obj =   0.000000000e+00  infeas =  0.000e+00
OPTIMAL SOLUTION FOUND BY LP PREPROCESSOR

 > [glpk] Primal optimality condition in solveCobraLP satisfied.
 > [mosek] Primal optimality condition in solveCobraLP satisfied.
 > [mosek] Dual optimality condition in solveCobraLP satisfied.
Could not find installation of matlab, so it cannot be tested

   ----------------------------------------------------------
   pdco.m                       Version pdco5 of 15 Jun 2018
   Primal-dual barrier method to minimize a convex function
   subject to linear constraints Ax + r = b,  bl <= x <= bu

   Michael Saunders       SOL and ICME, Stanford University
   Contributors:     Byunggyoo Kim (SOL), Chris Maes (ICME)
                     Santiago Akle (ICME), Matt Zahr (ICME)
                     Aekaansh Verma (ME)
   ----------------------------------------------------------

The objective is linear
The matrix A is an explicit sparse matrix

m        =        1     n        =        2      nnz(A)  =        1
max |b | =        0     max |x0| =  1.0e+00      xsize   =  1.0e+00
max |y0| =        1     max |z0| =  1.0e+00      zsize   =  1.0e+00

x0min    =        1     featol   =  1.0e-06      d1max   =  1.0e-04
z0min    =        1     opttol   =  1.0e-06      d2max   =  5.0e-04
mu0      =  1.0e-01     steptol  =     0.99     bigcenter=     1000
```

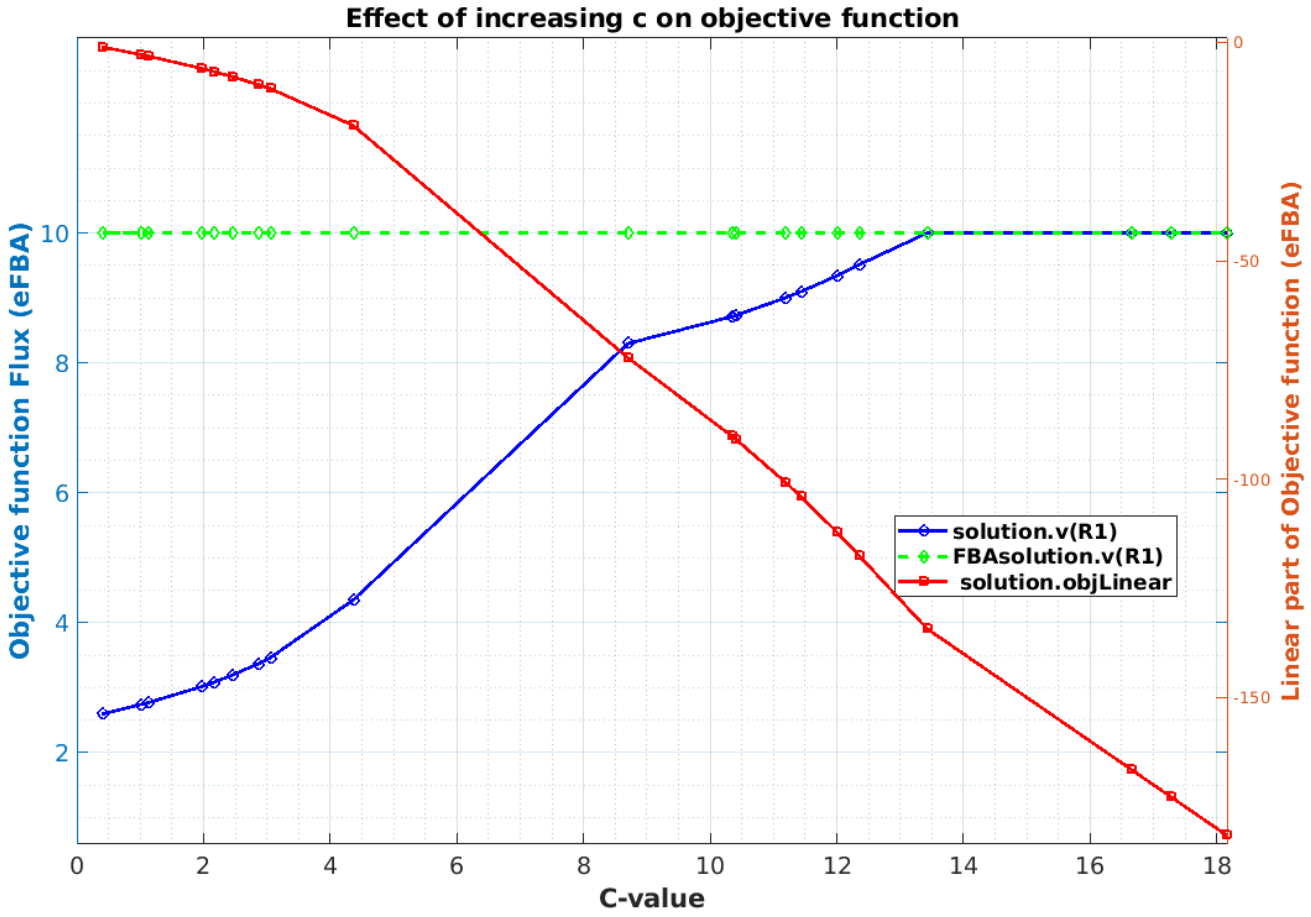
```
LSMR/MINRES:
atol1    =  1.0e-10     atol2    =  1.0e-15      btol    =  0.0e+00
conlim   =  1.0e+12     itnlim   =       10      show    =        0

Method   =        2     (1 or 11=chol  2 or 12=QR  3 or 13=LSMR  4 or 14=MINRES 21=SQD(LU)  22=SQD(MA57))
Eliminating dy before dx


Bounds:
  [0,inf]  [-inf,0]  Finite bl  Finite bu  Two bnds   Fixed    Free
        0         0          0          0         0       2       0
  [0, bu]  [bl,  0]  excluding fixed variables
        0         0

Itn   mu stepx stepz  Pinf  Dinf  Cinf   Objective    nf  center       QR
  0                   -6.6 -99.0  -Inf  1.2500000e-07        1.0
  1 -1.0 1.000 1.000 -99.0 -99.0  -Inf  0.0000000e+00  1     1.0        1
  2 -3.0 1.000 1.000 -99.0 -99.0  -Inf  0.0000000e+00  1     1.0
  3 -5.0 1.000 1.000 -99.0 -99.0  -Inf  0.0000000e+00  1     1.0
  4 -7.0 1.000 1.000 -99.0 -99.0  -Inf  0.0000000e+00  1     1.0
   Converged

max |x| =     0.000    max |y| =     0.000    max |z| =     0.000   scaled
max |x| =     0.000    max |y| =     0.000    max |z| =     0.000 unscaled
max |x| and max |z| exclude fixed variables
PDitns  =         4     QRitns =         0    cputime =       0.0

Distribution of vector     x         z
[      1,     10 )         0         2
[    0.1,      1 )         0         0
[   0.01,    0.1 )         0         0
[  0.001,   0.01 )         0         0
[ 0.0001,  0.001 )         0         0
[  1e-05, 0.0001 )         0         0
[  1e-06,  1e-05 )         0         0
[  1e-07,  1e-06 )         0         0
[  1e-08,  1e-07 )         0         0
[      0,  1e-08 )         2         0
Elapsed time is 0.009168 seconds.

 > [pdco] Primal optimality condition in solveCobraLP satisfied.
 > [pdco] Dual optimality condition in solveCobraLP satisfied.
Could not find installation of quadMinos, so it cannot be tested
Could not find installation of dqqMinos, so it cannot be tested
Could not find installation of cplex_direct, so it cannot be tested

 > [cplexlp] Primal optimality condition in solveCobraLP satisfied.
 > [cplexlp] Dual optimality condition in solveCobraLP satisfied.
Could not find installation of tomlab_snopt, so it cannot be tested
 Done.
 > Setting default solvers ...Could not find installation of matlab, so it cannot be tested
 Done.
 > Saving the MATLAB path ... Done.
   - The MATLAB path was saved as ~/pathdef.m.

 > Summary of available solvers and solver interfaces

            Support        LP      MILP        QP      MIQP       NLP        EP
    ------------------------------------------------------------------------------
    gurobi          active               1        1        1        1        -        -
    ibm_cplex       active               1        1        1        1        -        -
    tomlab_cplex    active               0        0        0        0        -        -
    glpk            active               1        1        -        -        -        -
```

```
    mosek           active              1        -        1        -        -        1
    matlab          active              0        -        -        -        0        -
    pdco            active              1        -        1        -        -        1
    quadMinos       active              0        -        -        -        -        -
    dqqMinos        active              0        -        0        -        -        -
    cplex_direct    active              0        0        0        -        -        -
    cplexlp         active              1        -        -        -        -        -
    qpng            passive             -        -        1        -        -        -
    tomlab_snopt    passive             -        -        -        -        0        -
    lp_solve        legacy              1        -        -        -        -        -
    ------------------------------------------------------------------------------
    Total           -                   7        3        5        2        0        2

 + Legend: - = not applicable, 0 = solver not compatible or not installed, 1 = solver installed.


 > You can solve LP problems using: 'gurobi' - 'ibm_cplex' - 'glpk' - 'mosek' - 'pdco' - 'cplexlp'
 > You can solve MILP problems using: 'gurobi' - 'ibm_cplex' - 'glpk'
 > You can solve QP problems using: 'gurobi' - 'ibm_cplex' - 'mosek' - 'pdco'
 > You can solve MIQP problems using: 'gurobi' - 'ibm_cplex'
 > You can solve NLP problems using:
 > You can solve EP problems using: 'mosek' - 'pdco'

 > Checking for available updates ...
> You cannot update your fork using updateCobraToolbox() because this is a development branch.
> The current branch is: master
> The last commit to the current branch is: 097dc1
> You can use MATLAB.devTools (https://github.com/opencobra/MATLAB.devTools) to update your fork.
removing: /home/samira/fork-cobratoolbox/src/analysis/thermo/componentContribution/new
removing: /home/samira/fork-cobratoolbox/src/analysis/thermo/groupContribution/new
removing: /home/samira/fork-cobratoolbox/src/analysis/thermo/inchi/new
removing: /home/samira/fork-cobratoolbox/src/analysis/thermo/molFiles/new
removing: /home/samira/fork-cobratoolbox/src/analysis/thermo/protons/new
removing: /home/samira/fork-cobratoolbox/src/analysis/thermo/trainingModel/new
```


Effect of increasing c on objective function
solution.obj
solution.objEntropy
solution.objLinear
0
-20
-40
-60
-80
-100
-120
-140
-160
-180
-200
0
2
4
6
8
10
12
14
16
18
C-value


Effect of increasing c on objective function
Objective function Flux (eFBA)
Objective function (eFBA)
C-value
solution.v(R1)
FBAsolution.v(R1)
solution.obj
0
2
4
6
8
10
12
14
16
18
2
3
4
5
6
7
8
9
10
-20
-40
-60
-80
-100
-120
-140


Effect of increasing c on objective function
Objective function Flux (eFBA)
Entropic part of Objective function (eFBA)
C-value
solution.v(R1)
FBAsolution.v(R1)
solution.objEntropy
0
2
4
6
8
10
12
14
16
18
2
4
6
8
10
0
-10
-20



Use **tmorous mesenchymal** as an example:

```
clear all
warning('off', 'all')
load('~/drive/glioblastoma/results/csf_media/defaultModels/absolute/-2/
TCmesnh/scRecon3D_mesnh.mat')
 model = scRecon3D_mesnh;
 param.solver = 'mosek';
param = mosekParamSetEFBA(param);
[solution,~] = entropicFluxBalanceAnalysis(model,param);
```

```
Using existing internal net flux bounds without modification.

MOSEK Version 10.1.15 (Build date: 2023-10-12 20:25:19)
Copyright (c) MOSEK ApS, Denmark WWW: mosek.com
Platform: Linux/64-X86

Problem
  Name                   :
  Objective sense        : minimize
  Type                   : CONIC (conic optimization problem)
  Constraints            : 3821
  Affine conic cons.     : 4532 (13596 rows)
  Disjunctive cons.      : 0
```

```
  Cones                  : 0
  Scalar variables       : 9329
  Matrix variables       : 0
  Integer variables      : 0

Optimizer started.
Presolve started.
Linear dependency checker started.
Linear dependency checker terminated.
Eliminator started.
Freed constraints in eliminator : 0
Eliminator terminated.
Eliminator - tries                  : 1                 time                   : 0.00
Lin. dep.  - tries                  : 1                 time                   : 0.00
Lin. dep.  - primal attempts        : 1                 successes              : 1
Lin. dep.  - dual attempts          : 0                 successes              : 0
Lin. dep.  - primal deps.           : 46                dual deps.             : 0
Presolve terminated. Time: 0.01
Optimizer  - threads                : 6
Optimizer  - solved problem         : the primal
Optimizer  - Constraints            : 3559
Optimizer  - Cones                  : 4532
Optimizer  - Scalar variables       : 15966             conic                  : 13596
Optimizer  - Semi-definite variables: 0                 scalarized             : 0
Factor     - setup time             : 0.02
Factor     - dense det. time        : 0.00              GP order time          : 0.00
Factor     - nonzeros before factor : 2.07e+04          after factor           : 3.29e+04
Factor     - dense dim.             : 0                 flops                  : 1.35e+06
ITE PFEAS    DFEAS    GFEAS    PRSTATUS   POBJ              DOBJ              MU       TIME
0   1.0e+04  1.0e+00  9.6e+03  0.00e+00   3.751763625e+03   -5.850484381e+03  1.0e+00  0.04
1   6.1e+03  6.1e-01  7.5e+03  -1.00e+00  8.313076808e+03   -1.287894680e+03  6.1e-01  0.05
2   5.0e+03  5.0e-01  6.8e+03  -1.00e+00  1.112763764e+04   1.527382355e+03   5.0e-01  0.06
3   1.7e+03  1.7e-01  4.0e+03  -1.00e+00  4.148339684e+04   3.189078369e+04   1.7e-01  0.07
4   6.2e+02  6.2e-02  2.4e+03  -9.99e-01  1.292870514e+05   1.197149432e+05   6.2e-02  0.07
5   2.4e+02  2.4e-02  1.5e+03  -9.97e-01  3.572449862e+05   3.477239896e+05   2.4e-02  0.08
6   1.5e+02  1.5e-02  1.2e+03  -9.91e-01  5.777499688e+05   5.682774441e+05   1.5e-02  0.09
7   5.7e+01  5.7e-03  7.1e+02  -9.86e-01  1.568819718e+06   1.559568322e+06   5.7e-03  0.09
8   2.4e+01  2.4e-03  4.5e+02  -9.59e-01  3.574612144e+06   3.565849568e+06   2.4e-03  0.11
9   1.5e+01  1.5e-03  3.3e+02  -8.84e-01  5.333036825e+06   5.324856390e+06   1.5e-03  0.12
10  9.7e+00  9.7e-04  2.5e+02  -7.64e-01  6.797962076e+06   6.790586654e+06   9.7e-04  0.13
11  6.1e+00  6.1e-04  1.6e+02  -5.53e-01  7.360832852e+06   7.354732475e+06   6.1e-04  0.14
12  5.7e+00  5.7e-04  1.5e+02  -1.74e-01  7.092266424e+06   7.086435891e+06   5.7e-04  0.15
13  3.4e+00  3.4e-04  8.1e+01  -9.66e-02  5.426427962e+06   5.422241669e+06   3.4e-04  0.16
14  1.9e+00  1.9e-04  3.5e+01  3.30e-01   3.319803309e+06   3.317220350e+06   1.9e-04  0.17
15  1.0e+00  1.0e-04  1.5e+01  6.45e-01   2.038763664e+06   2.037238923e+06   1.0e-04  0.17
16  6.1e-01  6.1e-05  7.1e+00  8.15e-01   1.298751555e+06   1.297829593e+06   6.1e-05  0.18
17  5.7e-01  5.7e-05  6.4e+00  9.01e-01   1.205250834e+06   1.204391739e+06   5.7e-05  0.19
18  3.4e-01  3.4e-05  3.0e+00  9.09e-01   7.451237743e+05   7.446090848e+05   3.4e-05  0.20
19  1.1e-01  1.1e-05  6.3e-01  9.52e-01   2.722482999e+05   2.720739261e+05   1.1e-05  0.20
20  7.1e-02  7.1e-06  3.2e-01  9.89e-01   1.739122982e+05   1.738033011e+05   7.1e-06  0.21
21  4.1e-02  4.1e-06  1.4e-01  9.94e-01   1.006840270e+05   1.006219591e+05   4.1e-06  0.22
22  2.0e-02  2.0e-06  4.7e-02  9.98e-01   4.701830842e+04   4.698871132e+04   2.0e-06  0.22
23  1.1e-02  1.1e-06  1.9e-02  1.00e+00   2.425817555e+04   2.424215085e+04   1.1e-06  0.23
24  4.6e-03  4.6e-07  5.6e-03  1.00e+00   8.973177465e+03   8.966243553e+03   4.6e-07  0.24
25  1.5e-03  1.5e-07  1.1e-03  1.00e+00   1.516039780e+03   1.513786029e+03   1.5e-07  0.24
26  5.7e-04  5.7e-08  2.4e-04  1.00e+00   -5.284016613e+02  -5.292396071e+02  5.7e-08  0.25
27  3.9e-04  3.9e-08  1.4e-04  1.00e+00   -8.805553505e+02  -8.811355183e+02  3.9e-08  0.26
28  1.9e-04  1.9e-08  4.6e-05  1.00e+00   -1.299360601e+03  -1.299635631e+03  1.9e-08  0.26
29  6.4e-05  6.4e-09  9.3e-06  1.00e+00   -1.544003826e+03  -1.544097855e+03  6.4e-09  0.27
30  2.1e-05  2.1e-09  1.7e-06  1.00e+00   -1.628423008e+03  -1.628453078e+03  2.1e-09  0.28
31  7.8e-06  7.8e-10  4.0e-07  1.00e+00   -1.652572166e+03  -1.652583634e+03  7.8e-10  0.29
32  5.5e-06  5.5e-10  2.3e-07  1.00e+00   -1.657062117e+03  -1.657070094e+03  5.5e-10  0.30
33  1.9e-06  1.9e-10  4.8e-08  1.00e+00   -1.663672571e+03  -1.663675361e+03  1.9e-10  0.31
34  1.3e-06  1.3e-10  2.7e-08  1.00e+00   -1.664786582e+03  -1.664788498e+03  1.3e-10  0.31
```

```
35  7.0e-07  7.0e-11  1.1e-08  1.00e+00   -1.665911928e+03  -1.665912949e+03  7.0e-11  0.32
36  2.1e-07  2.1e-11  1.8e-09  1.00e+00   -1.666811683e+03  -1.666811992e+03  2.1e-11  0.33
37  5.1e-08  5.1e-12  2.1e-10  1.00e+00   -1.667106905e+03  -1.667106980e+03  5.1e-12  0.34
38  1.7e-08  1.7e-12  4.0e-11  1.00e+00   -1.667171007e+03  -1.667171032e+03  1.7e-12  0.35
39  4.3e-09  4.3e-13  5.1e-12  1.00e+00   -1.667194344e+03  -1.667194350e+03  4.3e-13  0.35
40  9.2e-10  9.2e-14  5.1e-13  1.00e+00   -1.667200707e+03  -1.667200709e+03  9.2e-14  0.36
41  2.2e-10  2.2e-14  5.9e-14  1.00e+00   -1.667202042e+03  -1.667202042e+03  2.2e-14  0.37
42  2.2e-10  5.7e-15  6.4e-15  1.00e+00   -1.667202362e+03  -1.667202362e+03  5.0e-15  0.38
Optimizer terminated. Time: 0.38


Interior-point solution summary
  Problem status  : PRIMAL_AND_DUAL_FEASIBLE
  Solution status : OPTIMAL
  Primal.  obj: -1.6672023624e+03   nrm: 1e+00    Viol.  con: 7e-07    var: 0e+00    acc: 9e-08
  Dual.    obj: -1.6672023624e+03   nrm: 2e+03    Viol.  con: 8e-12    var: 1e-11    acc: 0e+00
Optimizer summary
  Optimizer                 -                        time: 0.38
    Interior-point          - iterations : 42        time: 0.38
      Basis identification  -                        time: 0.00
        Primal              - iterations : 0         time: 0.00
        Dual                - iterations : 0         time: 0.00
        Clean primal        - iterations : 0         time: 0.00
        Clean dual          - iterations : 0         time: 0.00
    Simplex                 -                        time: 0.00
      Primal simplex        - iterations : 0         time: 0.00
      Dual simplex          - iterations : 0         time: 0.00
    Mixed integer           - relaxations: 0         time: 0.00


 > [mosek] Primal optimality condition in solveCobraEP satisfied.
 > [mosek] Dual   optimality condition in solveCobraEP satisfied.

Optimality conditions (biochemistry)
 7.2e-07 || N*(vf - vr) + B*ve - b ||_inf
 7.6e-12 || cf + ci + N'*y_N + y_vi + Qv*vf + k_vf + z_vf ||_inf
 7.6e-12 || cr - ci - N'*y_N - y_vi + Qv*vf + k_vr + z_vr ||_inf
 1.1e-11 || ce + B'*y_N  + z_ve ||_inf
       0 || k_e_1 + z_e_1 ||_inf
 1.6e-12 || -g + k_e_vf + z_e_vf||_inf
 1.6e-12 || -g + k_e_vr + z_e_vr||_inf
 1.7e-07 || e_vf + vf*log(vf) ||_inf
 1.1e-07 || e_vr + vr*log(vr) ||_inf

Derived optimality conditions (biochemistry)
 2.3e-05 || g.*log(vf) + g - k_vf ||_inf
   3e-05 || g.*log(vr) + g - k_vr ||_inf
 2.3e-05 || cf + ci + N'*y_N + y_vi + Qv*vf + g.*log(vf) + g + z_vf ||_inf
   3e-05 || cr - ci - N'*y_N - y_vi + Qv*vf + g.*log(vr) + g + z_vr ||_inf

Thermo conditions
       2 || g.*log(vr/vf) - 2*N'*y_N ||_inf
 3.5e-05 || g.*log(vr/vf) + cr - cf - 2*ci - 2*N'*y_N - 2*y_vi ||_inf
 3.5e-05 || g.*log(vr/vf) + cr - cf - 2*ci - 2*N'*y_N - 2*y_vi - z_vr + z_vf ||_inf
 2.6e-11 min(slack)
 7.2e-07 max(slack)
```

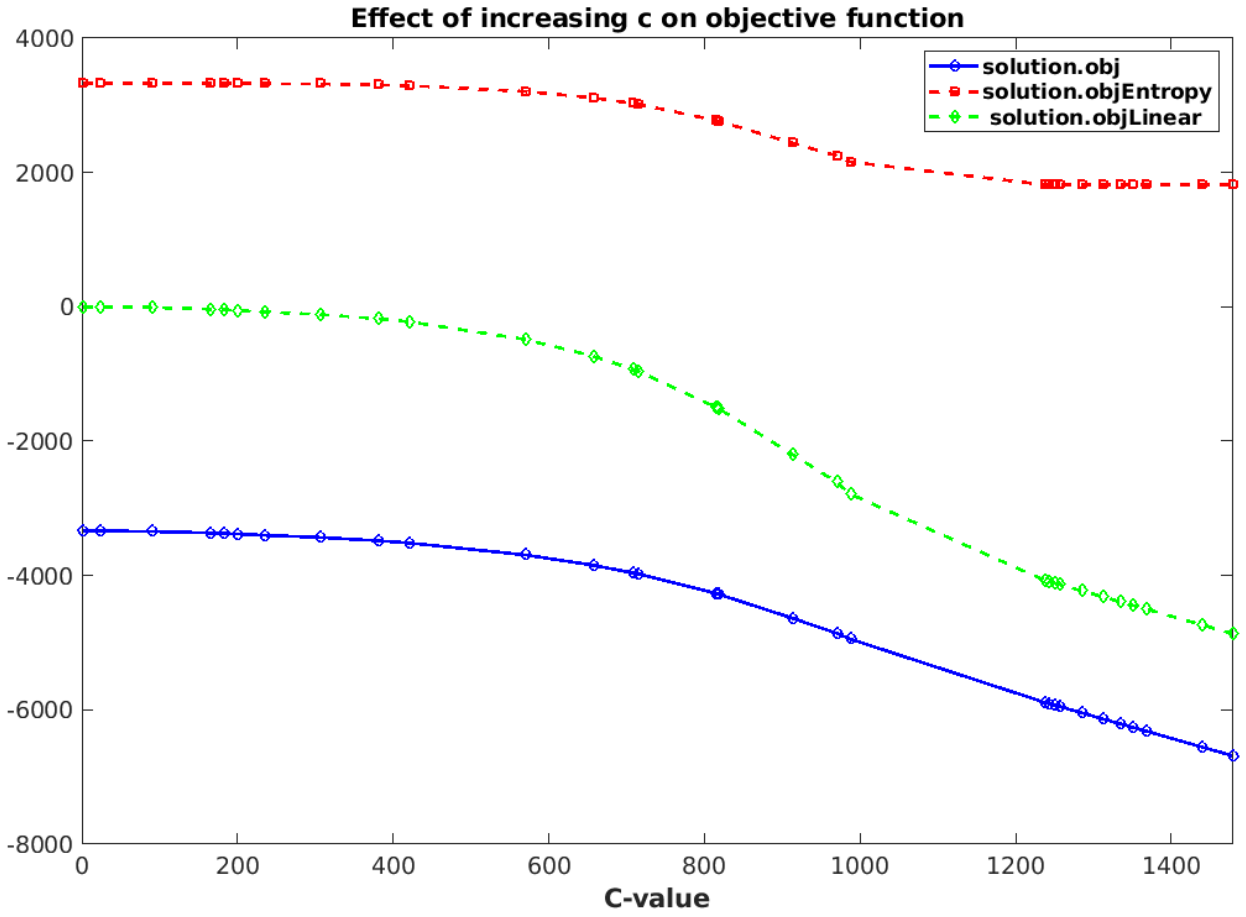
```
plotyy_eFBA(model,'biomass_reaction',1500,30,0);
```

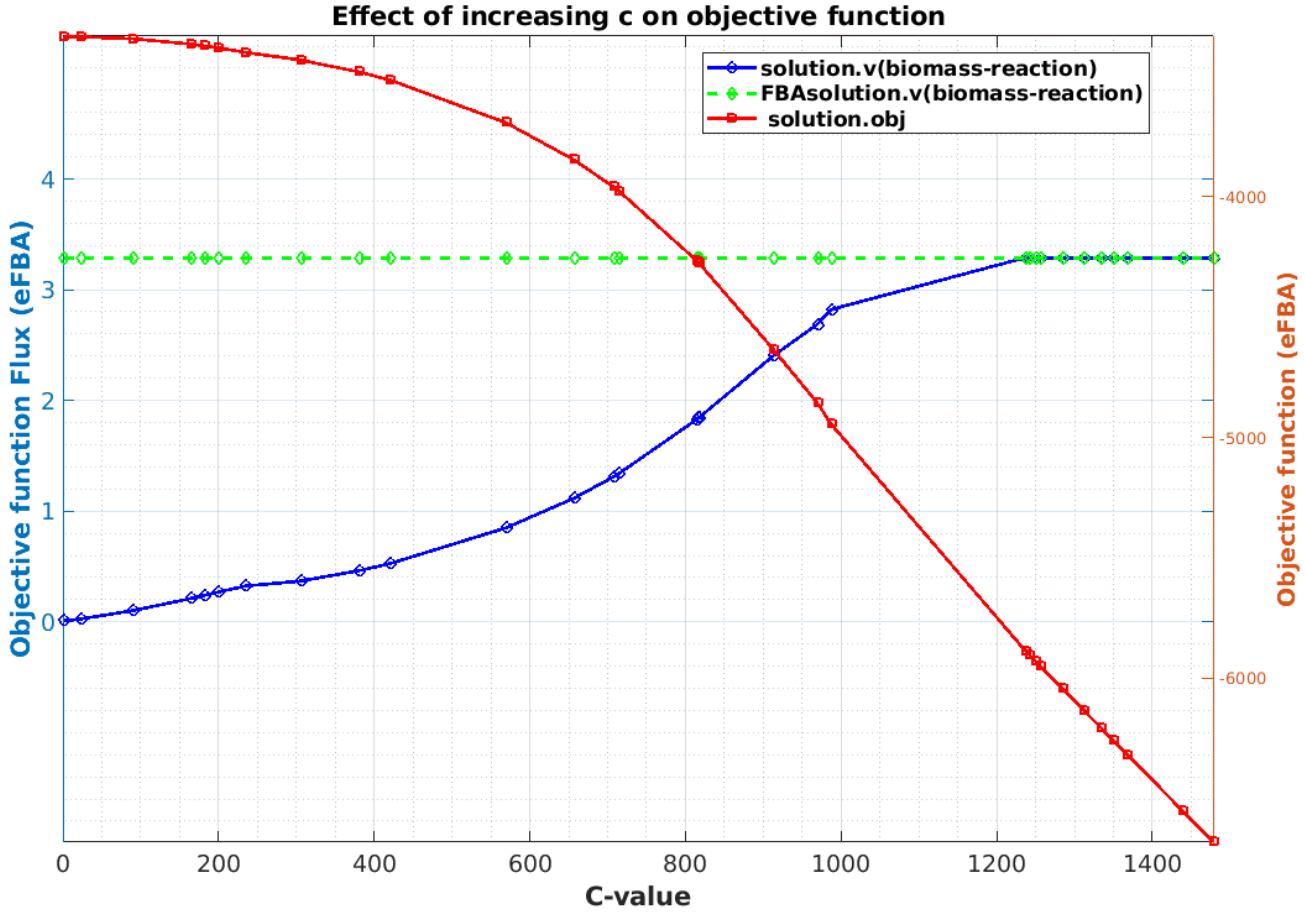
```
      _____   _____   _____   _____     _____      |
     /  ___| /  _  \ |  _  \ |  _  \   / ___ \     |   COnstraint-Based Reconstruction and Analysis
     | |     | | | | | |_| | | |_| |   | |___| |    |   The COBRA Toolbox - 2024
     | |     | | | | |  _  { |  _  /   |  ___  |   |
```

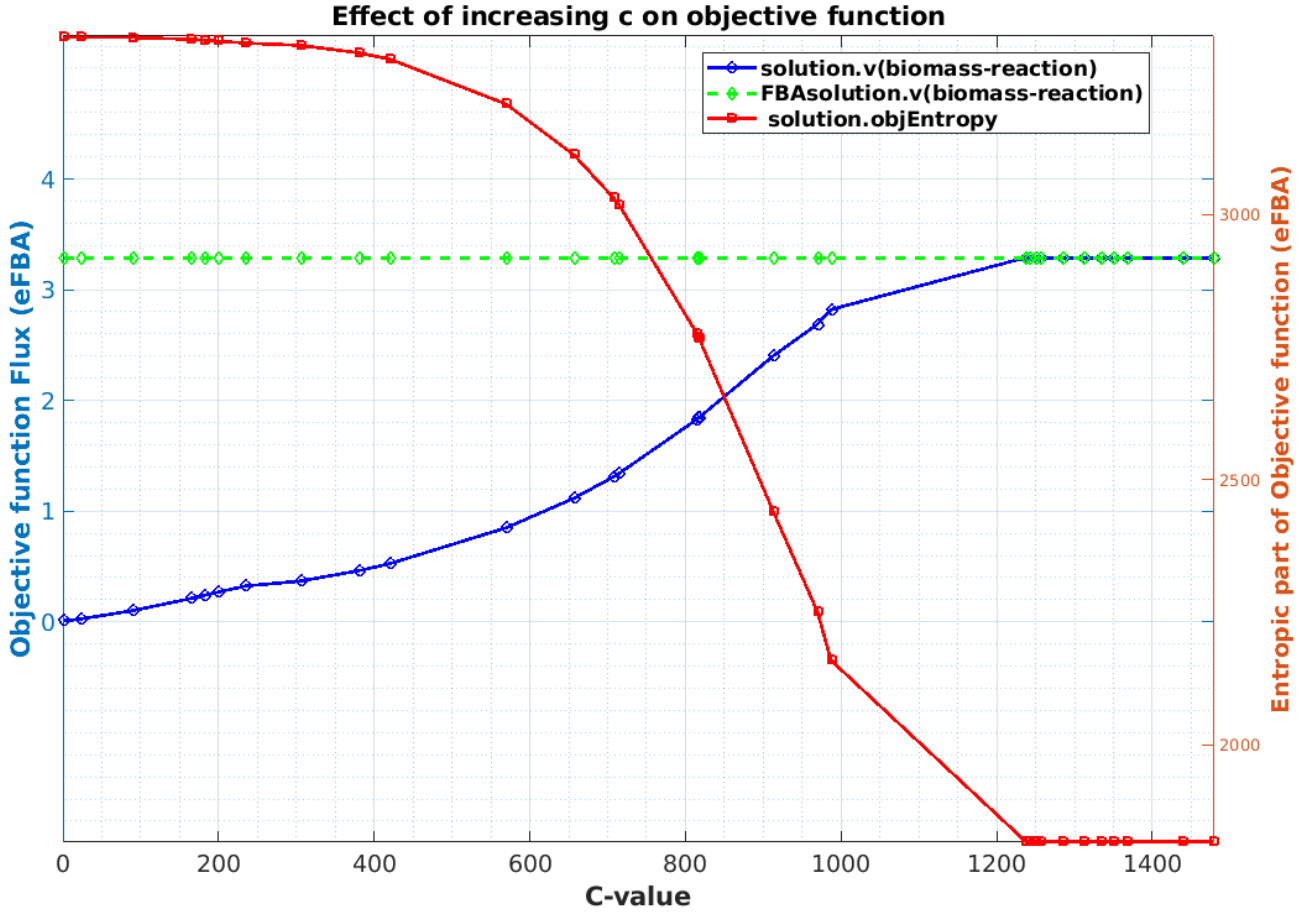
```
     | |___  | |_| | | |_| | | | \ \  | |   | |   |   Documentation:
     \_____| \_____/ |_____/ |_|  \_\ |_|   |_|   |   http://opencobra.github.io/cobratoolbox
                                                 |
 > Checking if git is installed ...  Done (version: 2.25.1).
 > Checking if the repository is tracked using git ...  Done.
 > Checking if curl is installed ...  Done.
 > Checking if remote can be reached ...  Done.
 > Initializing and updating submodules (this may take a while)... Done.
 > Adding all the files of The COBRA Toolbox ...  Done.
 > Define CB map output... set to svg.
 > TranslateSBML is installed and working properly.
 > Configuring solver environment variables ...
   - [*---] ILOG_CPLEX_PATH: /opt/ibm/ILOG/CPLEX_Studio1210/cplex/matlab/x86-64_linux
   - [*---] GUROBI_PATH: /opt/gurobi1003/linux64/matlab
   - [----] TOMLAB_PATH: --> set this path manually after installing the solver ( see instructions )
   - [*---] MOSEK_PATH: /opt/mosek/10.1/
   Done.
 > Checking available solvers and solver interfaces ...     0

     0

Check osense*c - A'*lam - w = 0 (stationarity):
     0
     0


 > [gurobi] Primal optimality condition in solveCobraLP satisfied.
 > [gurobi] Dual optimality condition in solveCobraLP satisfied.
Version identifier: 12.10.0.0 | 2019-11-26 | 843d4de
CPXPARAM_Output_WriteLevel                       3
CPXPARAM_Output_CloneLog                         -1
Found incumbent of value 0.000000 after 0.00 sec. (0.00 ticks)

Root node processing (before b&c):
  Real time             =    0.00 sec. (0.00 ticks)
Parallel b&c, 12 threads:
  Real time             =    0.00 sec. (0.00 ticks)
  Sync time (average)   =    0.00 sec.
  Wait time (average)   =    0.00 sec.
                          ------------
Total (root+branch&cut) =    0.00 sec. (0.00 ticks)
Could not find installation of tomlab_cplex, so it cannot be tested
GLPK Simplex Optimizer, v4.42
1 row, 2 columns, 1 non-zero
Preprocessing...
~     0: obj =   0.000000000e+00  infeas =  0.000e+00
OPTIMAL SOLUTION FOUND BY LP PREPROCESSOR

 > [glpk] Primal optimality condition in solveCobraLP satisfied.
 > [mosek] Primal optimality condition in solveCobraLP satisfied.
 > [mosek] Dual optimality condition in solveCobraLP satisfied.
Could not find installation of matlab, so it cannot be tested

   ------------------------------------------------------
   pdco.m                     Version pdco5 of 15 Jun 2018
   Primal-dual barrier method to minimize a convex function
   subject to linear constraints Ax + r = b,  bl <= x <= bu

   Michael Saunders       SOL and ICME, Stanford University
   Contributors:     Byunggyoo Kim (SOL), Chris Maes (ICME)
                     Santiago Akle (ICME), Matt Zahr (ICME)
                     Aekaansh Verma (ME)
   ------------------------------------------------------
```

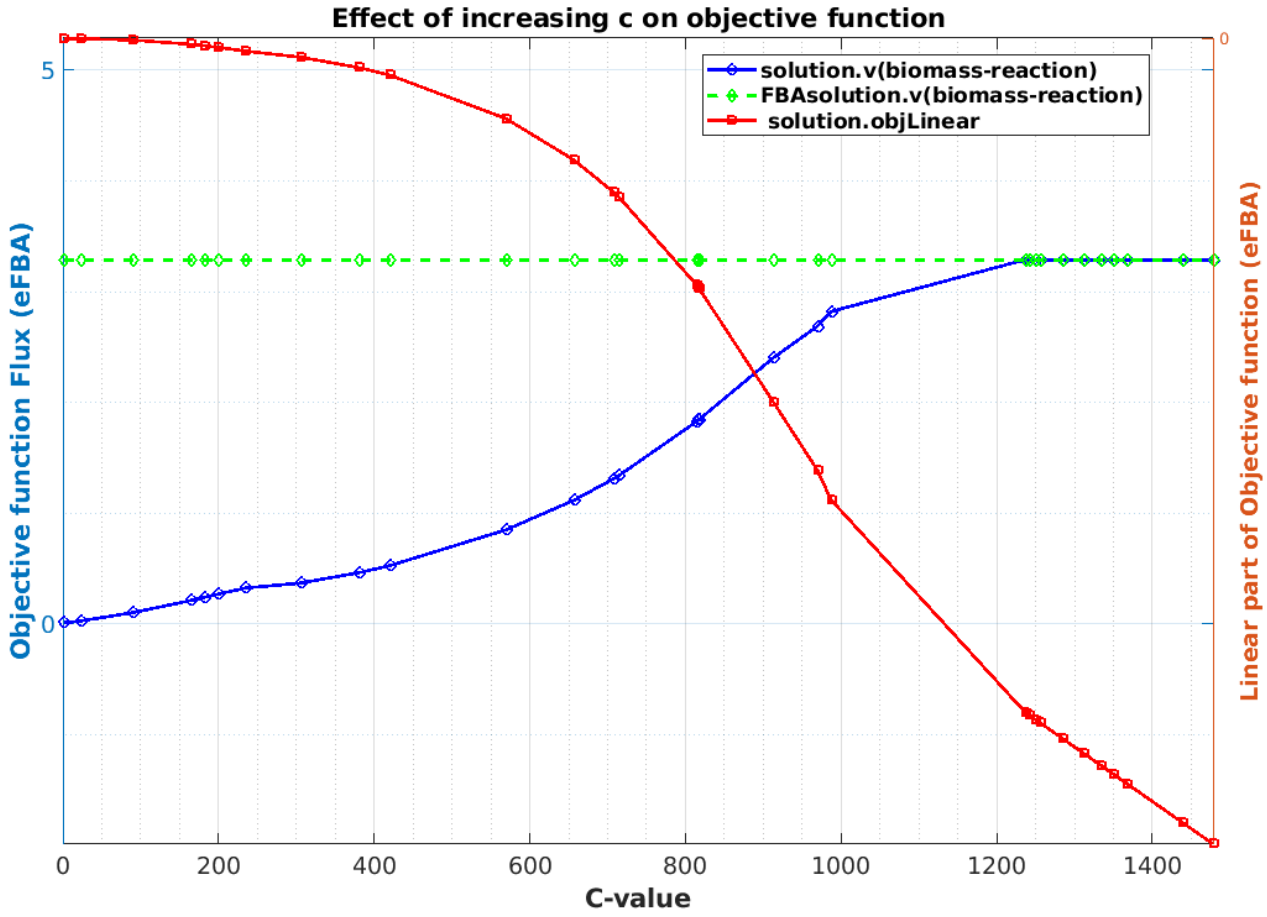
```
The objective is linear
The matrix A is an explicit sparse matrix

m        =        1     n        =        2      nnz(A)  =        1
max |b | =        0     max |x0| =  1.0e+00      xsize   =  1.0e+00
max |y0| =        1     max |z0| =  1.0e+00      zsize   =  1.0e+00

x0min    =        1     featol   =  1.0e-06      d1max   =  1.0e-04
z0min    =        1     opttol   =  1.0e-06      d2max   =  5.0e-04
mu0      =  1.0e-01     steptol  =     0.99     bigcenter=     1000

LSMR/MINRES:
atol1    =  1.0e-10     atol2    =  1.0e-15      btol    =  0.0e+00
conlim   =  1.0e+12     itnlim   =       10      show    =        0

Method   =        2     (1 or 11=chol  2 or 12=QR  3 or 13=LSMR  4 or 14=MINRES 21=SQD(LU)  22=SQD(MA57))
Eliminating dy before dx


Bounds:
  [0,inf]  [-inf,0]  Finite bl  Finite bu  Two bnds   Fixed    Free
        0         0          0          0         0       2       0
  [0, bu]  [bl,  0]  excluding fixed variables
        0         0

Itn   mu stepx stepz  Pinf  Dinf  Cinf   Objective    nf  center       QR
  0                  -6.6 -99.0  -Inf  1.2500000e-07        1.0
  1 -1.0 1.000 1.000 -99.0 -99.0  -Inf  0.0000000e+00  1     1.0        1
  2 -3.0 1.000 1.000 -99.0 -99.0  -Inf  0.0000000e+00  1     1.0
  3 -5.0 1.000 1.000 -99.0 -99.0  -Inf  0.0000000e+00  1     1.0
  4 -7.0 1.000 1.000 -99.0 -99.0  -Inf  0.0000000e+00  1     1.0
   Converged

max |x| =     0.000    max |y| =     0.000    max |z| =     0.000   scaled
max |x| =     0.000    max |y| =     0.000    max |z| =     0.000 unscaled
max |x| and max |z| exclude fixed variables
PDitns  =         4     QRitns =         0    cputime =       0.0

Distribution of vector     x         z
[      1,     10 )         0         2
[    0.1,      1 )         0         0
[   0.01,    0.1 )         0         0
[  0.001,   0.01 )         0         0
[ 0.0001,  0.001 )         0         0
[  1e-05, 0.0001 )         0         0
[  1e-06,  1e-05 )         0         0
[  1e-07,  1e-06 )         0         0
[  1e-08,  1e-07 )         0         0
[      0,  1e-08 )         2         0
Elapsed time is 0.026625 seconds.

 > [pdco] Primal optimality condition in solveCobraLP satisfied.
 > [pdco] Dual optimality condition in solveCobraLP satisfied.
Could not find installation of quadMinos, so it cannot be tested
Could not find installation of dqqMinos, so it cannot be tested
Could not find installation of cplex_direct, so it cannot be tested

 > [cplexlp] Primal optimality condition in solveCobraLP satisfied.
 > [cplexlp] Dual optimality condition in solveCobraLP satisfied.
Could not find installation of tomlab_snopt, so it cannot be tested
 Done.
 > Setting default solvers ...Could not find installation of matlab, so it cannot be tested
 Done.
```

```
 > Saving the MATLAB path ... Done.
   - The MATLAB path was saved as ~/pathdef.m.

 > Summary of available solvers and solver interfaces

          Support        LP      MILP        QP      MIQP       NLP        EP
    ------------------------------------------------------------------------------
    gurobi           active               1         1         1         1         -         -
    ibm_cplex        active               1         1         1         1         -         -
    tomlab_cplex     active               0         0         0         0         -         -
    glpk             active               1         1         -         -         -         -
    mosek            active               1         -         1         -         -         1
    matlab           active               0         -         -         -         0         -
    pdco             active               1         -         1         -         -         1
    quadMinos        active               0         -         -         -         -         -
    dqqMinos         active               0         -         0         -         -         -
    cplex_direct     active               0         0         0         -         -         -
    cplexlp          active               1         -         -         -         -         -
    qpng             passive              -         -         1         -         -         -
    tomlab_snopt     passive              -         -         -         -         0         -
    lp_solve         legacy               1         -         -         -         -         -
    ------------------------------------------------------------------------------
    Total            -                    7         3         5         2         0         2

 + Legend: - = not applicable, 0 = solver not compatible or not installed, 1 = solver installed.


 > You can solve LP problems using: 'gurobi' - 'ibm_cplex' - 'glpk' - 'mosek' - 'pdco' - 'cplexlp'
 > You can solve MILP problems using: 'gurobi' - 'ibm_cplex' - 'glpk'
 > You can solve QP problems using: 'gurobi' - 'ibm_cplex' - 'mosek' - 'pdco'
 > You can solve MIQP problems using: 'gurobi' - 'ibm_cplex'
 > You can solve NLP problems using:
 > You can solve EP problems using: 'mosek' - 'pdco'

 > Checking for available updates ...
> You cannot update your fork using updateCobraToolbox() because this is a development branch.
> The current branch is: master
> The last commit to the current branch is: 097dc1
> You can use MATLAB.devTools (https://github.com/opencobra/MATLAB.devTools) to update your fork.
removing: /home/samira/fork-cobratoolbox/src/analysis/thermo/componentContribution/new
removing: /home/samira/fork-cobratoolbox/src/analysis/thermo/groupContribution/new
removing: /home/samira/fork-cobratoolbox/src/analysis/thermo/inchi/new
removing: /home/samira/fork-cobratoolbox/src/analysis/thermo/molFiles/new
removing: /home/samira/fork-cobratoolbox/src/analysis/thermo/protons/new
removing: /home/samira/fork-cobratoolbox/src/analysis/thermo/trainingModel/new
```


Effect of increasing c on objective function
solution.obj
solution.objEntropy
solution.objLinear
4000
2000
0
-2000
-4000
-6000
-8000
0
200
400
600
800
1000
1200
1400
C-value


Effect of increasing c on objective function
solution.v(biomass-reaction)
FBAsolution.v(biomass-reaction)
solution.obj
Objective function Flux (eFBA)
Objective function (eFBA)
C-value
0
200
400
600
800
1000
1200
1400
0
1
2
3
4
-4000
-5000
-6000


Effect of increasing c on objective function
solution.v(biomass-reaction)
FBAsolution.v(biomass-reaction)
solution.objEntropy
Objective function Flux (eFBA)
Entropic part of Objective function (eFBA)
C-value
0
200
400
600
800
1000
1200
1400
0
1
2
3
4
2000
2500
3000



Estimation of C (coefficient of objective function)

After understanding the behavior of flux in the objective function as a function of C, the question becomes: which value of C is the best for modeling? The value of flux changes between zero and the maximum value calculated in FBA. The C at the maximum value is not the best, since our model will never meet that value. It is important to know what is expected from the model in reality. The answer to this question comes from experiments. For example, in the case of biomass, we know that the growth rate is related to doubling time according to the equation below. If we know the doubling time of the cell, we can find the C value corresponding to that value in modeling.

```
clear all
 warning('off', 'all')
load('~/drive/glioblastoma/results/csf_media/defaultModels/absolute/-2/
TCmesnh/scRecon3D_mesnh.mat');
model = scRecon3D_mesnh;
```

```
solution_vals(:,1) = 0:50:1400;
param.printLevel = 0;
```

```
param.solver = 'mosek';
%model
initCobraToolbox
```

```
                                                                |
     _____   _____   _____   _____     _____                    |
    /  ___| /  _  \ |  _  \ |  _  \   / ___ \                   |   COnstraint-Based Reconstruction and Analysis
    | |     | | | | | |_| | | |_| |   | |___| |                  |   The COBRA Toolbox - 2024
    | |     | | | | |  _  { |  _  /   |  ___  |                |
    | |___  | |_| | | |_| | | | \ \   | |   | |                |   Documentation:
    \_____| \_____/ |_____/ |_|  \_\ |_|   |_|                  |   http://opencobra.github.io/cobratoolbox
                                                                |

 > Checking if git is installed ...  Done (version: 2.25.1).
 > Checking if the repository is tracked using git ...  Done.
 > Checking if curl is installed ...  Done.
 > Checking if remote can be reached ...  Done.
 > Initializing and updating submodules (this may take a while)... Done.
 > Adding all the files of The COBRA Toolbox ...  Done.
 > Define CB map output... set to svg.
 > TranslateSBML is installed and working properly.
 > Configuring solver environment variables ...
   - [*---] ILOG_CPLEX_PATH: /opt/ibm/ILOG/CPLEX_Studio1210/cplex/matlab/x86-64_linux
   - [*---] GUROBI_PATH: /opt/gurobi1003/linux64/matlab
   - [----] TOMLAB_PATH: --> set this path manually after installing the solver ( see instructions )
   - [*---] MOSEK_PATH: /opt/mosek/10.1/
   Done.
 > Checking available solvers and solver interfaces ...     0

     0

Check osense*c - A'*lam - w = 0 (stationarity):
     0
     0


 > [gurobi] Primal optimality condition in solveCobraLP satisfied.
 > [gurobi] Dual optimality condition in solveCobraLP satisfied.
Version identifier: 12.10.0.0 | 2019-11-26 | 843d4de
CPXPARAM_Output_WriteLevel                       3
CPXPARAM_Output_CloneLog                         -1
Found incumbent of value 0.000000 after 0.00 sec. (0.00 ticks)

Root node processing (before b&c):
  Real time             =    0.00 sec. (0.00 ticks)
Parallel b&c, 12 threads:
  Real time             =    0.00 sec. (0.00 ticks)
  Sync time (average)   =    0.00 sec.
  Wait time (average)   =    0.00 sec.
                          ------------
Total (root+branch&cut) =    0.00 sec. (0.00 ticks)
Could not find installation of tomlab_cplex, so it cannot be tested
GLPK Simplex Optimizer, v4.42
1 row, 2 columns, 1 non-zero
Preprocessing...
~     0: obj =   0.000000000e+00  infeas =  0.000e+00
OPTIMAL SOLUTION FOUND BY LP PREPROCESSOR

 > [glpk] Primal optimality condition in solveCobraLP satisfied.
 > [mosek] Primal optimality condition in solveCobraLP satisfied.
 > [mosek] Dual optimality condition in solveCobraLP satisfied.
Could not find installation of matlab, so it cannot be tested

   ------------------------------------------------------------
   pdco.m                     Version pdco5 of 15 Jun 2018
```

```
   Primal-dual barrier method to minimize a convex function
   subject to linear constraints Ax + r = b,  bl <= x <= bu

   Michael Saunders       SOL and ICME, Stanford University
   Contributors:     Byunggyoo Kim (SOL), Chris Maes (ICME)
                     Santiago Akle (ICME), Matt Zahr (ICME)
                     Aekaansh Verma (ME)
   ---------------------------------------------------------

The objective is linear
The matrix A is an explicit sparse matrix

m        =        1     n        =        2      nnz(A)  =        1
max |b | =        0     max |x0| =  1.0e+00      xsize   =  1.0e+00
max |y0| =        1     max |z0| =  1.0e+00      zsize   =  1.0e+00

x0min    =        1     featol   =  1.0e-06      d1max   =  1.0e-04
z0min    =        1     opttol   =  1.0e-06      d2max   =  5.0e-04
mu0      =  1.0e-01     steptol  =     0.99     bigcenter=     1000

LSMR/MINRES:
atol1    =  1.0e-10     atol2    =  1.0e-15      btol    =  0.0e+00
conlim   =  1.0e+12     itnlim   =       10      show    =        0

Method   =        2     (1 or 11=chol  2 or 12=QR  3 or 13=LSMR  4 or 14=MINRES 21=SQD(LU)  22=SQD(MA57))
Eliminating dy before dx


Bounds:
  [0,inf]  [-inf,0]  Finite bl  Finite bu  Two bnds   Fixed    Free
        0         0          0          0         0       2       0
  [0, bu]  [bl,  0]  excluding fixed variables
        0         0

Itn   mu stepx stepz  Pinf  Dinf  Cinf   Objective    nf  center       QR
  0                  -6.6 -99.0  -Inf  1.2500000e-07        1.0
  1 -1.0 1.000 1.000 -99.0 -99.0  -Inf  0.0000000e+00  1     1.0        1
  2 -3.0 1.000 1.000 -99.0 -99.0  -Inf  0.0000000e+00  1     1.0
  3 -5.0 1.000 1.000 -99.0 -99.0  -Inf  0.0000000e+00  1     1.0
  4 -7.0 1.000 1.000 -99.0 -99.0  -Inf  0.0000000e+00  1     1.0
   Converged

max |x| =     0.000    max |y| =     0.000    max |z| =     0.000   scaled
max |x| =     0.000    max |y| =     0.000    max |z| =     0.000 unscaled
max |x| and max |z| exclude fixed variables
PDitns  =         4     QRitns =         0    cputime =       0.1

Distribution of vector     x         z
[      1,     10 )         0         2
[    0.1,      1 )         0         0
[   0.01,    0.1 )         0         0
[  0.001,   0.01 )         0         0
[ 0.0001,  0.001 )         0         0
[  1e-05, 0.0001 )         0         0
[  1e-06,  1e-05 )         0         0
[  1e-07,  1e-06 )         0         0
[  1e-08,  1e-07 )         0         0
[      0,  1e-08 )         2         0
Elapsed time is 0.025806 seconds.

 > [pdco] Primal optimality condition in solveCobraLP satisfied.
 > [pdco] Dual optimality condition in solveCobraLP satisfied.
Could not find installation of quadMinos, so it cannot be tested
Could not find installation of dqqMinos, so it cannot be tested
```

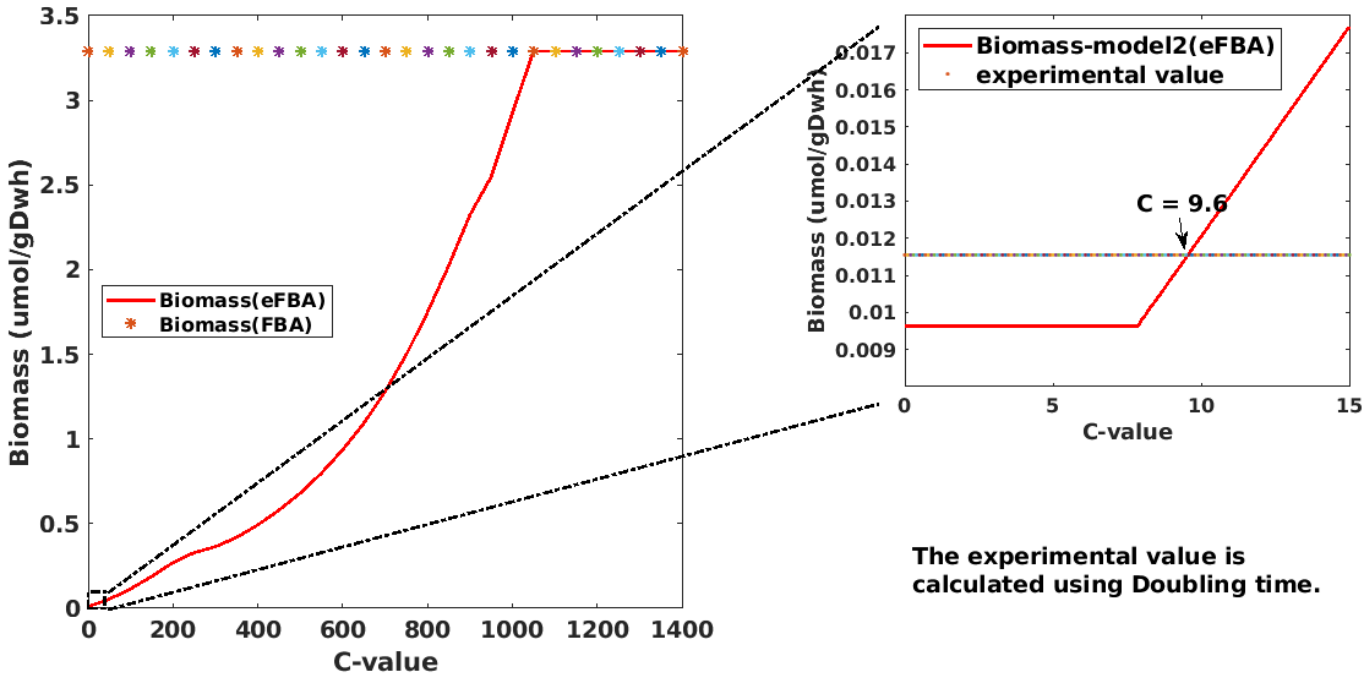
```
Could not find installation of cplex_direct, so it cannot be tested

 > [cplexlp] Primal optimality condition in solveCobraLP satisfied.
 > [cplexlp] Dual optimality condition in solveCobraLP satisfied.
Could not find installation of tomlab_snopt, so it cannot be tested
 Done.
 > Setting default solvers ...Could not find installation of matlab, so it cannot be tested
 Done.
 > Saving the MATLAB path ... Done.
   - The MATLAB path was saved as ~/pathdef.m.

 > Summary of available solvers and solver interfaces

          Support        LP      MILP        QP      MIQP       NLP        EP
    ------------------------------------------------------------------------
    gurobi          active               1         1         1         1         -         -
    ibm_cplex       active               1         1         1         1         -         -
    tomlab_cplex    active               0         0         0         0         -         -
    glpk            active               1         1         -         -         -         -
    mosek           active               1         -         1         -         -         1
    matlab          active               0         -         -         -         0         -
    pdco            active               1         -         1         -         -         1
    quadMinos       active               0         -         -         -         -         -
    dqqMinos        active               0         -         0         -         -         -
    cplex_direct    active               0         0         0         -         -         -
    cplexlp         active               1         -         -         -         -         -
    qpng            passive              -         -         1         -         -         -
    tomlab_snopt    passive              -         -         -         -         0         -
    lp_solve        legacy               1         -         -         -         -         -
    ------------------------------------------------------------------------
    Total           -                    7         3         5         2         0         2

 + Legend: - = not applicable, 0 = solver not compatible or not installed, 1 = solver installed.


 > You can solve LP problems using: 'gurobi' - 'ibm_cplex' - 'glpk' - 'mosek' - 'pdco' - 'cplexlp'
 > You can solve MILP problems using: 'gurobi' - 'ibm_cplex' - 'glpk'
 > You can solve QP problems using: 'gurobi' - 'ibm_cplex' - 'mosek' - 'pdco'
 > You can solve MIQP problems using: 'gurobi' - 'ibm_cplex'
 > You can solve NLP problems using:
 > You can solve EP problems using: 'mosek' - 'pdco'

 > Checking for available updates ...
> You cannot update your fork using updateCobraToolbox() because this is a development branch.
> The current branch is: master
> The last commit to the current branch is: 097dc1
> You can use MATLAB.devTools (https://github.com/opencobra/MATLAB.devTools) to update your fork.
removing: /home/samira/fork-cobratoolbox/src/analysis/thermo/componentContribution/new
removing: /home/samira/fork-cobratoolbox/src/analysis/thermo/groupContribution/new
removing: /home/samira/fork-cobratoolbox/src/analysis/thermo/inchi/new
removing: /home/samira/fork-cobratoolbox/src/analysis/thermo/molFiles/new
removing: /home/samira/fork-cobratoolbox/src/analysis/thermo/protons/new
removing: /home/samira/fork-cobratoolbox/src/analysis/thermo/trainingModel/new
```

```
FBAsolution = optimizeCbModel(model,'max');
for nr = 1:length(solution_vals)
    model.c(ismember(model.rxns, 'biomass_reaction')) = solution_vals(nr,1);
    [solution,~] = entropicFluxBalanceAnalysis(model,param);
    solution_vals(nr,2) = solution.v(ismember(model.rxns,
'biomass_reaction'));
end
```

```
sub_solution_vals(:,1) = 0:0.1:15;
param.printLevel = 0;
for nr = 1:length(sub_solution_vals)
    model.c(ismember(model.rxns, 'biomass_reaction')) =
sub_solution_vals(nr,1);
    [solution,~] = entropicFluxBalanceAnalysis(model,param);
    sub_solution_vals(nr,2) = solution.v(ismember(model.rxns,
'biomass_reaction'));
end
```

```
figure('Renderer', 'painters', 'Position', [10 10 1600 800]);
% Define subplot positions
pos1 = [0.1, 0.1, 0.4, 0.8];  % [left, bottom, width, height]
pos2 = [0.65, 0.4, 0.30, 0.5];  % [left, bottom, width, height]
experimental_value = log(2)/60; % it is an example in case that doubling
time is considered to be 600 hourse
% Subplot 1
subplot('Position', pos1);
plot(solution_vals(:,1), solution_vals(:,2), 'red', ...
solution_vals(:,1), FBAsolution.f, '*', 'LineWidth', 2);

legend('Biomass(eFBA)','Biomass(FBA)',"FontSize",12, 'Location', 'West')
ylabel('Biomass (umol/gDwh)', "FontSize",14, "FontWeight","bold");
xlabel('C-value', "FontSize",14, "FontWeight","bold");
xlim([0 max(solution_vals(:,1))])
rectangle('Position', [0, 0, 40, 0.1], 'EdgeColor', 'black', 'LineWidth',
2,'LineStyle','-.');

% Draw lines from rectangle to the next subplot
x1 = pos1(1) + pos1(3);
y1 = pos1(2) + pos1(4) / 2;
x2 = pos2(1);
y2 = pos2(2) + pos2(4) / 2;
set(gca, 'XTickLabel', get(gca, 'XTickLabel'), 'FontWeight', 'bold',
'FontSize', 14);

% Subplot 2
subplot('Position', pos2);
plot(sub_solution_vals(:,1), sub_solution_vals(:,2),'red',...
     sub_solution_vals(:,1),  experimental_value,'.', 'LineWidth',2)
legend('Biomass-model2(eFBA)','experimental value',"FontSize",14,
'Location',"NorthWest")
ylabel('Biomass (umol/gDwh)', "FontSize",14, "FontWeight","bold");
xlabel('C-value', "FontSize",14, "FontWeight","bold");
xlim([0 max(sub_solution_vals(:,1))])
% ylim([0.009 0.02])
% xticks(0:1:15);

%add crossing point box
```

```
[~,index] = min(abs(sub_solution_vals(:,2) - experimental_value));
% the closest intersection is:
[sub_solution_vals(index,1) experimental_value];
annotation("textarrow", [0.835 0.8385], [0.6263 0.5854], "String", ['C =
',num2str(sub_solution_vals(index,1))], 'FontWeight', 'bold', 'FontSize', 14)

% text
annotation('textbox', [pos2(1), 0.1, pos2(3), 0.1], 'String', ['The
experimental value is calculated using Doubling time.'],...
    'HorizontalAlignment', 'left', 'VerticalAlignment','middle',
'FontWeight', 'bold', 'FontSize', 14,'EdgeColor', 'none');
set(gca, 'XTickLabel', get(gca, 'XTickLabel'), 'FontWeight', 'bold',
'FontSize', 12);
annotation("line", [0.113 0.6328], [0.1211 0.8845], 'Color', 'black',
'LineWidth', 2,'LineStyle','-.')
annotation("line", [0.113 0.6328], [0.09871 0.3762],'Color', 'black',
'LineWidth', 2,'LineStyle','-.')
% axes 1
% Property editing
positionPropObjs = findobj(gcf, "-property", "Position");
positionPropObjs(1).Position = [0 0 1212.0000, 606.0000];
```

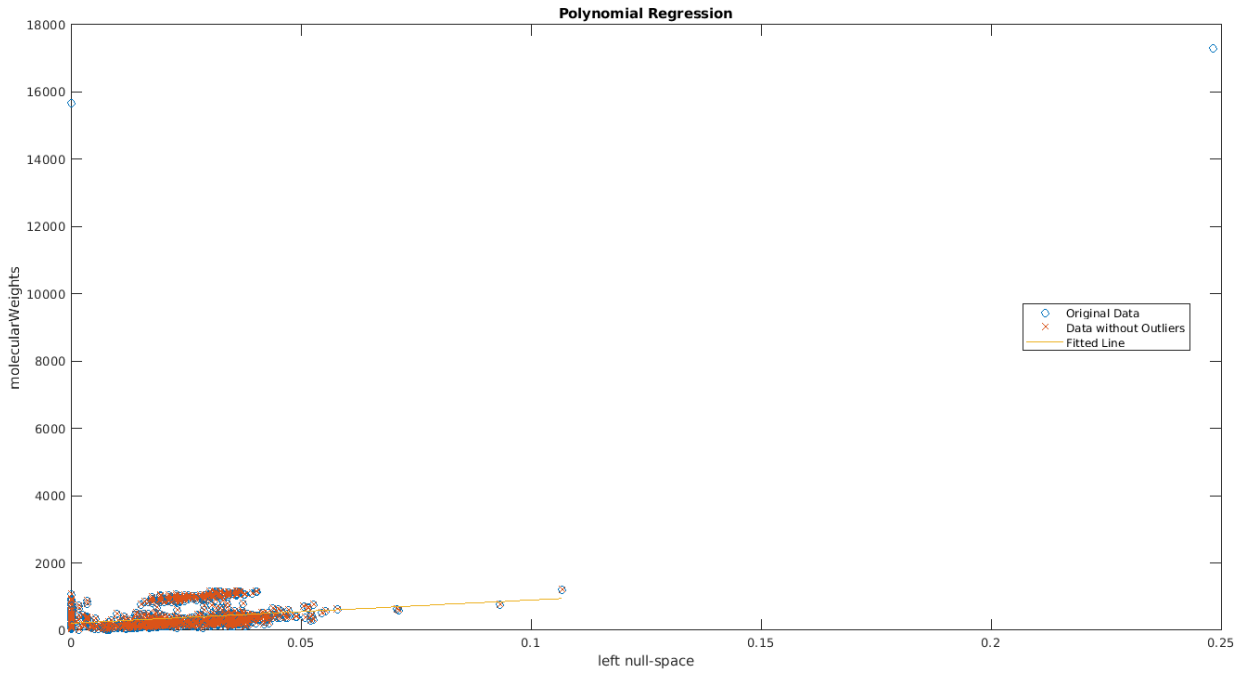


Putting weight on cr and cf to unbias transpot of massive molecules

1. **Using Mass**

```
[substratesMass, productsMass] = calculateReactionMasses(model);
```

```
All reactions that do not include an R-group are mass-balanced.
LeftNullSpace_nonzero = 1498
```




```
Fitted Polynomial Equation:
y = 6897.7526 * x^1 + 208.665
```

```
ave_mass = (substratesMass + productsMass) /2;
c_mass = ave_mass/ sum(ave_mass);% normalize to avoid too agressive results
```

2. Using Gene expression

```
T = readtable('~/drive/glioblastoma/data/input/csf_media/
scRef_celltypeAvg.csv');
geneWeight=calculateGeneWeight(model,T(:,[2,51]), 0.1);
```

```
% check to put weight only on internal reactions
j =1;
for i = 1: numel(model.rxns)
    if model.SConsistentRxnBool(i)
       SCReactionWeight(i) = geneWeight(j);
       SCReactionMass(i) = c_mass(j);
       j = j+1;
    elseif ~model.SConsistentRxnBool(i)
        SCReactionWeight(i)= 0;
        SCReactionMass(i) = 0;
    end
end
c_gene = -log(SCReactionWeight + 1);
c_Mass = SCReactionMass ;
```

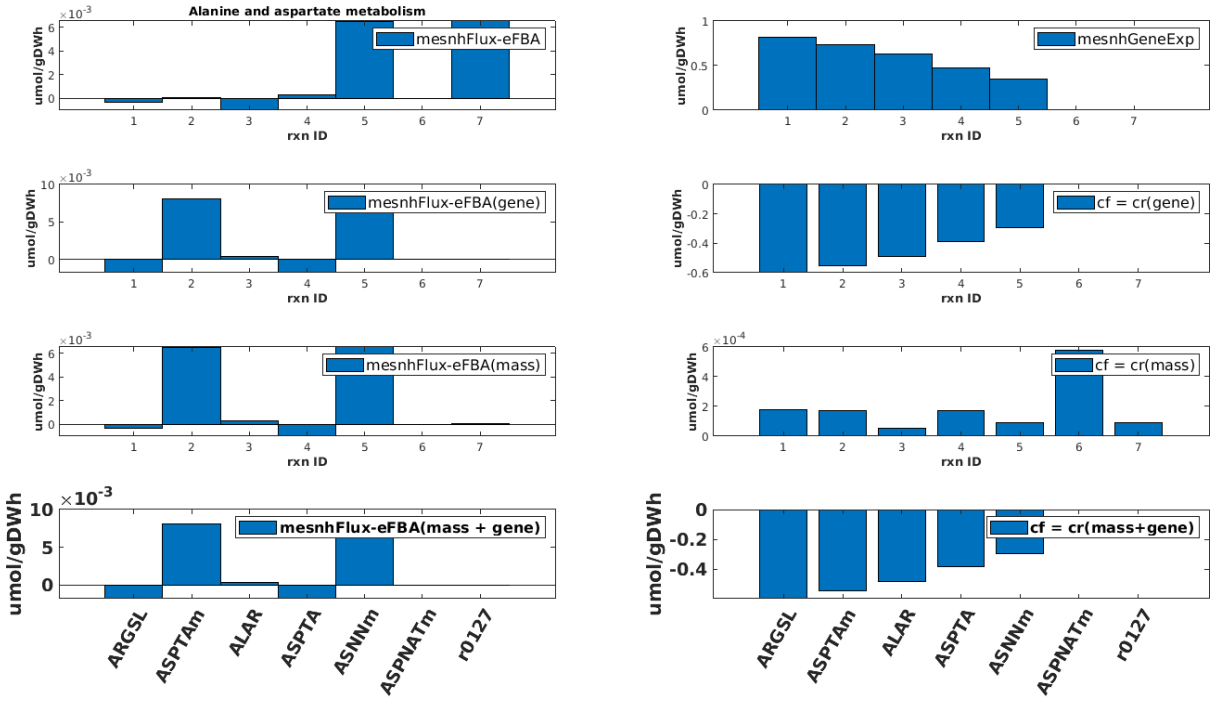
```
% weight on objective function
model.c(findRxnIDs(model, 'biomass_reaction')) = 9.6;%rec1
[solution,~] = entropicFluxBalanceAnalysis(model,param);
% changing cr and cf according to geneweight
model1 = model;
model1.cr = -log(SCReactionWeight + 1);%- min(-log(SCReactionWeight)));
model1.cf = -log(SCReactionWeight + 1); %- min(-log(SCReactionWeight)));

[solution1,~] = entropicFluxBalanceAnalysis(model1,param);
% changing cr and cf according to mass
model2 = model;
model2.cr = SCReactionMass;
model2.cf = SCReactionMass;

[solution2,~] = entropicFluxBalanceAnalysis(model2,param);

model3 = model;
model3.cr = model1.cr + model2.cr;
model3.cf = model1.cf + model2.cf;

[solution3,~] = entropicFluxBalanceAnalysis(model3,param);
% initCobraToolbox
FBAsolution = optimizeCbModel(model,'max');
FBAsolution1 = optimizeCbModel(model1,'max');
FBAsolution2 = optimizeCbModel(model2,'max');
FBAsolution3 = optimizeCbModel(model3,'max');
```

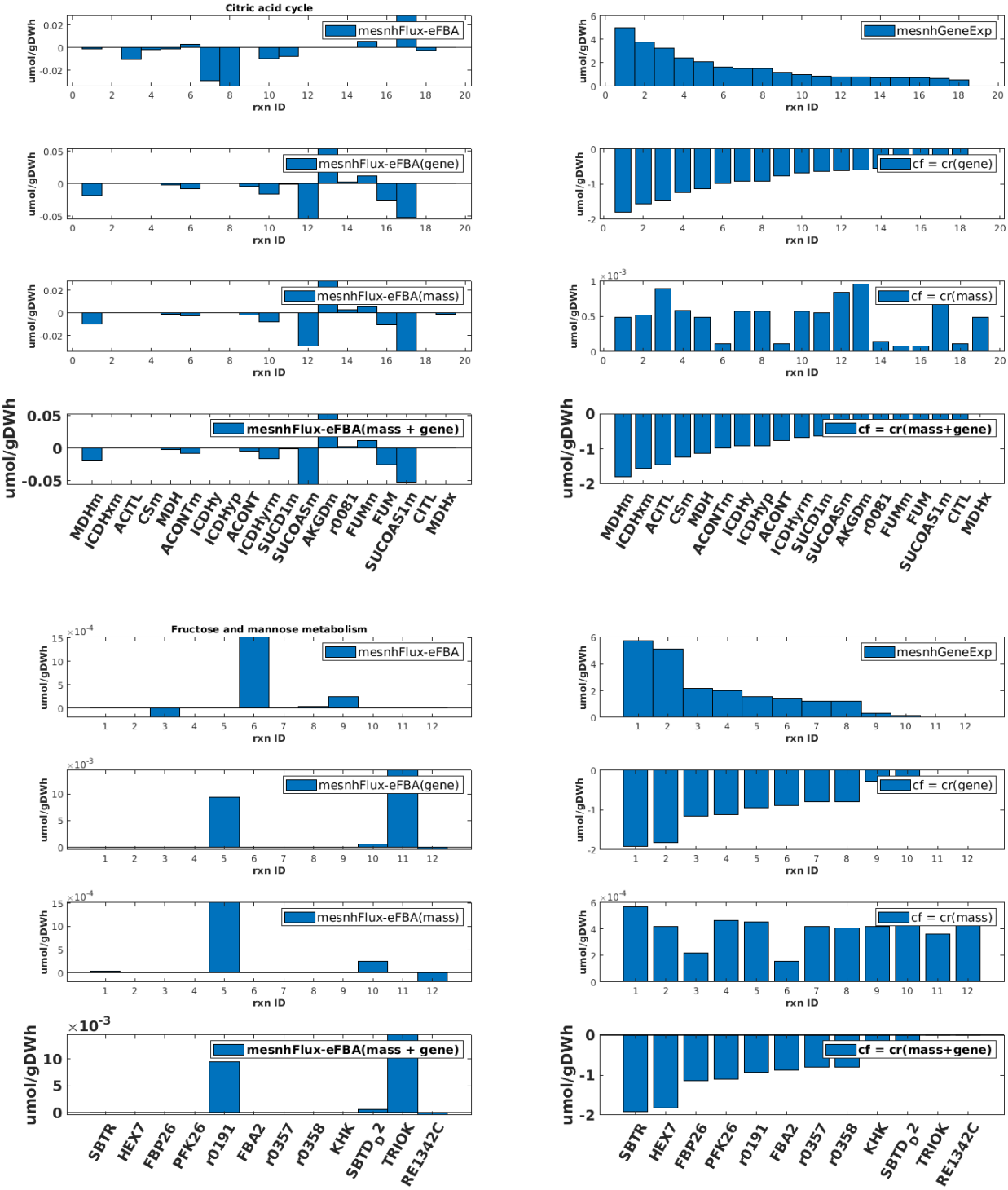
see the effect of different weights on net flux value in some subsystems:

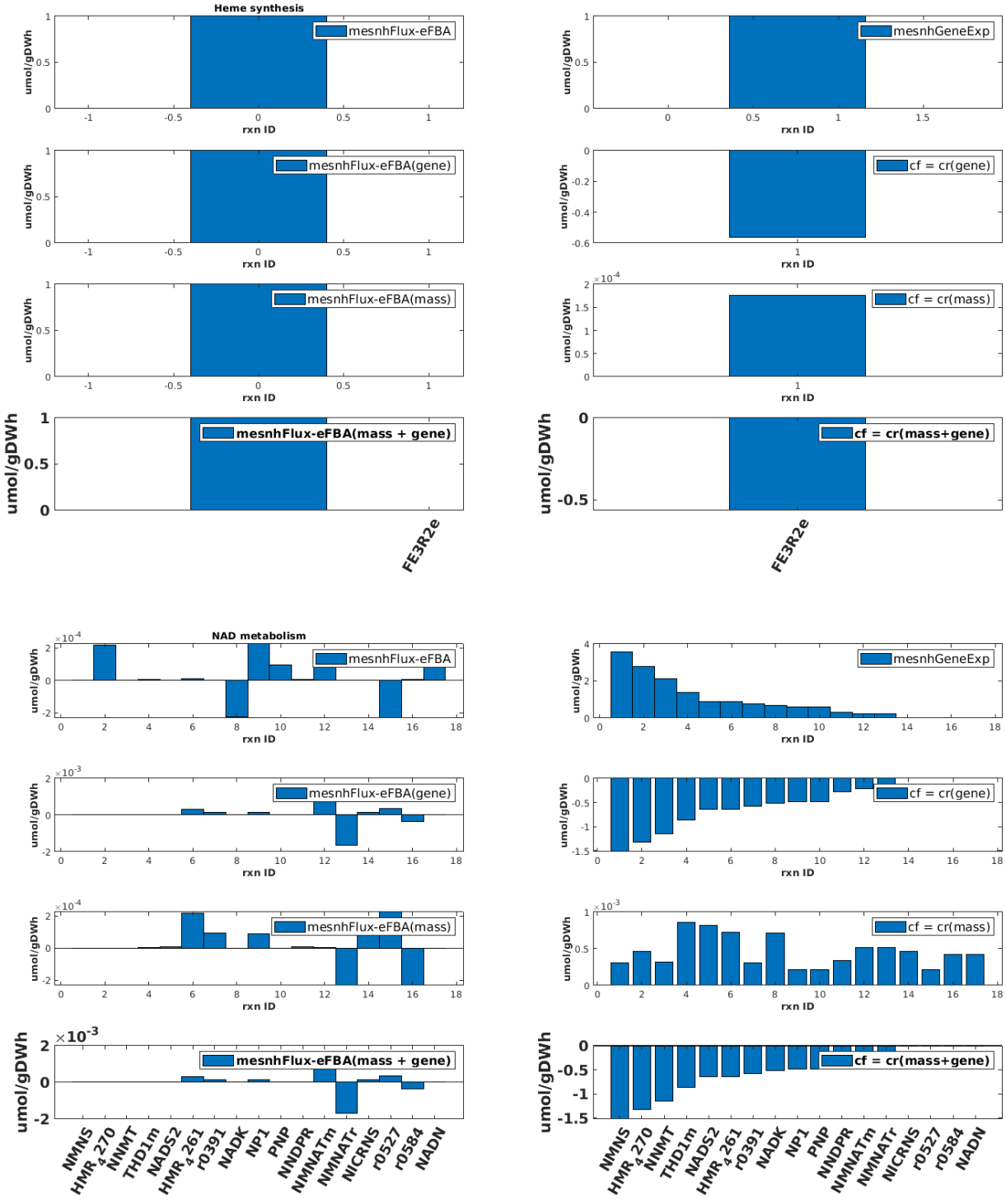
```
subSystem = unique(model.subSystems);

for i = 1:10:length(subSystem)

    figure('Renderer', 'painters', 'Position', [10 10 1600 800])
    ID = findRxnIDs(model,findRxnsFromSubSystem(model,subSystem(i)));
    [sortGeneWeight, idxgeneWeight] = sort(ID, 'descend');

    % findRxnsFromSubSystem(model,subSystem(i));
    %  printRxnFormula(model,model.rxns(ID))


    subplot(4,2,1)
    bar(solution.v(ID(idxgeneWeight)),1)
    legend('mesnhFlux-eFBA','FontSize',12)
    % set(gca,'XTick',
[1:length(model.rxns(ID(contains(findRxnsFromSubSystem(model,subSystem(i)),'r
ec1'))))],'xticklabel',regexprep(model.rxns(ID(contains(findRxnsFromSubSyste
m(model,subSystem(i)),'rec2'))),'\[[^\]]*\]',''))
    % set(gca,'XTickLabelRotation',60, 'FontSize',12)
```

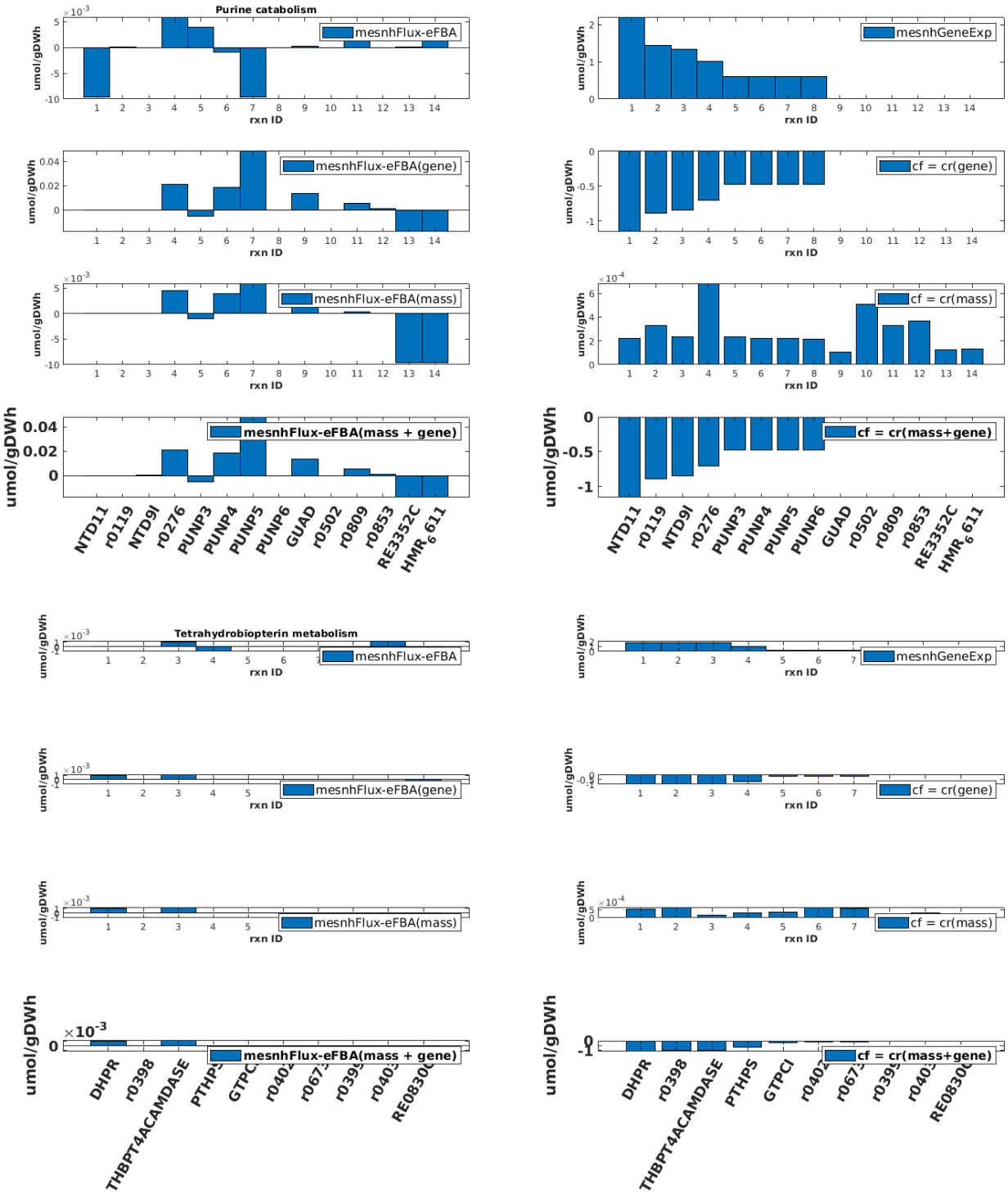
```
    title(char(subSystem(i)))
    xlabel('rxn ID', 'FontSize',14,'FontWeight','bold');
    ylabel('umol/gDWh', 'FontSize',14,'FontWeight','bold');

    subplot(4,2,2)
    [sortgenWeight, idxgeneWeight] = sort(SCReactionWeight(ID),'descend');
    bar(sortgenWeight,1)
    legend('mesnhGeneExp', 'FontSize',12)
    % set(gca,'XTick',
[1:length(model.rxns(ID(contains(findRxnsFromSubSystem(model,subSystem(i)),'r
ec1')))))],'xticklabel',regexprep(model.rxns(ID(contains(findRxnsFromSubSyste
m(model,subSystem(i)),'rec2'))),'\[[^\]]*\]',''))
    % set(gca,'XTickLabelRotation',60, 'FontSize',12)
    xlabel('rxn ID', 'FontSize',14,'FontWeight','bold');
    ylabel('umol/gDWh', 'FontSize',14,'FontWeight','bold');

    subplot(4,2,3)
    bar(solution1.v(ID(idxgeneWeight)),1)
    legend('mesnhFlux-eFBA(gene)','FontSize',12)
    xlabel('rxn ID', 'FontSize',14,'FontWeight','bold');
    ylabel('umol/gDWh', 'FontSize',14,'FontWeight','bold');

    subplot(4,2,4)
    bar(model1.cf(ID(idxgeneWeight)))
    legend('cf = cr(gene)','FontSize',12)
    % set(gca,'XTick',
[1:length(model.rxns(ID(contains(findRxnsFromSubSystem(model,subSystem(i)),'r
ec1')))))],'xticklabel',regexprep(model.rxns(ID(contains(findRxnsFromSubSyste
m(model,subSystem(i)),'rec2'))),'\[[^\]]*\]',''))
    % set(gca,'XTickLabelRotation',60, 'FontSize',12)
    % title('Gene expression in', char(subSystem(i)))
    xlabel('rxn ID', 'FontSize',14,'FontWeight','bold');
    ylabel('umol/gDWh', 'FontSize',14,'FontWeight','bold');

    subplot(4,2,5)
    bar(solution2.v(ID(idxgeneWeight)),1)
    legend('mesnhFlux-eFBA(mass)','FontSize',12)
    xlabel('rxn ID', 'FontSize',14,'FontWeight','bold');
    ylabel('umol/gDWh', 'FontSize',14,'FontWeight','bold');

    subplot(4,2,6)
    bar(model2.cf(ID(idxgeneWeight)))
    legend('cf = cr(mass)','FontSize',12)
    xlabel('rxn ID', 'FontSize',14,'FontWeight','bold');
    ylabel('umol/gDWh', 'FontSize',14,'FontWeight','bold');

    subplot(4,2,7)
    bar(solution3.v(ID(idxgeneWeight)),1)
    legend('mesnhFlux-eFBA(mass + gene)','FontSize',12)
```

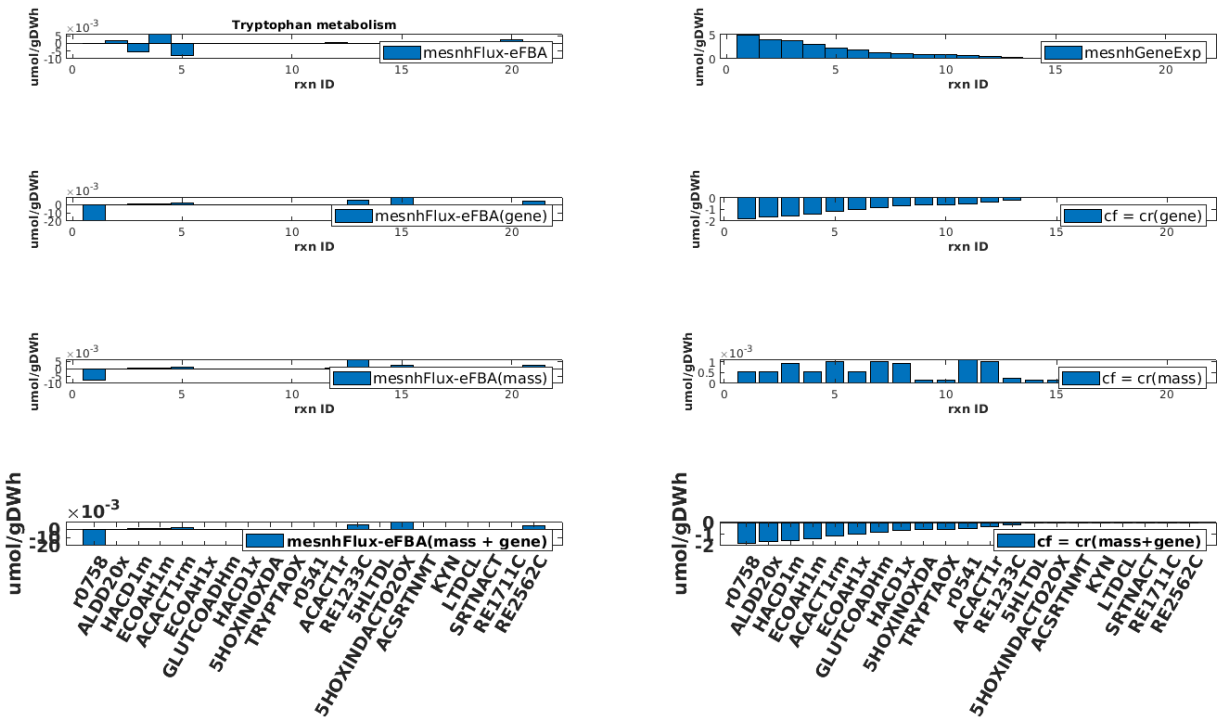
```
    set(gca,'XTick',
[1:length(model1.rxns(ID))],'xticklabel',regexprep(model1.rxns(ID(idxgeneWeig
ht)),'\[[^\]]*\]',''))
    set(gca,'XTickLabelRotation',60, 'FontSize',14,'FontWeight','bold')
    ylabel('umol/gDWh', 'FontSize',14,'FontWeight','bold');

    subplot(4,2,8)
    bar(model3.cf(ID(idxgeneWeight)))
    legend('cf = cr(mass+gene)','FontSize',12)
    set(gca,'XTick',
[1:length(model1.rxns(ID))],'xticklabel',regexprep(model1.rxns(ID(idxgeneWeig
ht)),'\[[^\]]*\]',''))
    set(gca,'XTickLabelRotation',60, 'FontSize',14,'FontWeight','bold')
    ylabel('umol/gDWh', 'FontSize',14,'FontWeight','bold');
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

end
```




Citric acid cycle
mesnhFlux-eFBA
mesnhGeneExp
mesnhFlux-eFBA(gene)
cf = cr(gene)
mesnhFlux-eFBA(mass)
cf = cr(mass)
mesnhFlux-eFBA(mass + gene)
cf = cr(mass+gene)
umol/gDWh
rxn ID
MDHm
ICDHxm
ACITL
CSm
MDH
ACONTm
ICDHy
ICDHyp
ACONT
ICDHyrm
SUCD1m
SUCOASm
AKGDm
r0081
FUMm
FUM
SUCOAS1m
CITL
MDHx
Fructose and mannose metabolism
mesnhFlux-eFBA
mesnhGeneExp
mesnhFlux-eFBA(gene)
cf = cr(gene)
mesnhFlux-eFBA(mass)
cf = cr(mass)
mesnhFlux-eFBA(mass + gene)
cf = cr(mass+gene)
umol/gDWh
rxn ID
SBTR
HEX7
FBP26
PFK26
r0191
FBA2
r0357
r0358
KHK
SBTD_2
TRIOK
RE1342C


Heme synthesis
mesnhFlux-eFBA
mesnhGeneExp
mesnhFlux-eFBA(gene)
cf = cr(gene)
mesnhFlux-eFBA(mass)
cf = cr(mass)
mesnhFlux-eFBA(mass + gene)
cf = cr(mass+gene)
umol/gDWh
rxn ID
FE3R2e
NAD metabolism
NMNS
HMR_4270
NNMT
THD1m
NADS2
HMR_4261
r0391
NADK
NP1
PNP
NNDPR
NMNATm
NMNATr
NICRNS
r0527
r0584
NADN


Purine catabolism
mesnhFlux-eFBA
mesnhGeneExp
mesnhFlux-eFBA(gene)
cf = cr(gene)
mesnhFlux-eFBA(mass)
cf = cr(mass)
mesnhFlux-eFBA(mass + gene)
cf = cr(mass+gene)
umol/gDWh
rxn ID
NTD11
r0119
NTD9l
r0276
PUNP3
PUNP4
PUNP5
PUNP6
GUAD
r0502
r0809
r0853
RE3352C
HMR_6611
Tetrahydrobiopterin metabolism
DHPR
r0398
THBPT4ACAMDASE
PTHPS
GTPCI



Acknowledgments

Co-funded by the European Union's Horizon Europe Framework Programme (101080997)

REFERENCES

[1] Fleming, R. M. T., Maes, C. M., Saunders, M. A., Ye, Y., and Palsson, B. O., "A variational principle for computing nonequilibrium fluxes and potentials in genome-scale biochemical networks", *Journal of Theoretical Biology* 292 (2012), pp. 71--77.