Verify the COBRA Toolbox

Authors: Ronan Fleming, Leiden University

Reviewers:

MATERIALS - EQUIPMENT SETUP

Please ensure that all the required dependencies (e.g. , `git` and `curl`) of The COBRA Toolbox have been properly installed by following the installation guide here.

PROCEDURE

Check available optimisation solvers

At initialisation, one from a set of available optimisation solvers will be selected as the default solver. If `Gurobi` is installed, it is used as the default solver for LP, QP and MILP problems. Otherwise, the `GLPK` solver is selected by for LP and MILP problems and QPNG is selected for QP problems. Check the currently selected solvers with:

```
changeCobraSolver
```

```
Defined solvers are:
    CBT_LP_SOLVER: glpk
    CBT_MILP_SOLVER: glpk
    CBT_QP_SOLVER: qpng
```

ANTICIPATED RESULTS

A list of solvers assigned to solve each class of optimisation solver is returned.

CRITICAL STEP

A dependency on at least one linear optimisation solver must be satisfied for flux balance analysis.

Verify a basic installation of the COBRA Toolbox

Test if flux balance analysis works

```
testFBA
```

```
   Testing flux balance analysis using glpk ...
>> Optimal minimum 1-norm solution
BiomassEcoli               0.9219
EX_co2(e)                   21.78
EX_glc(e)                     -10
EX_h2o(e)                   41.29
EX_h(e)                     8.428
EX_nh4(e)                  -9.851
EX_o2(e)                   -19.93
EX_pi(e)                  -0.8405
EX_so4(e)                 -0.2149

>> Optimal solution on fructose
```

```
BiomassEcoli               0.9219
EX_co2(e)                  21.78
EX_glc(e)                    -10
EX_h2o(e)                  41.29
EX_h(e)                    8.428
EX_nh4(e)                 -9.851
EX_o2(e)                  -19.93
EX_pi(e)                 -0.8405
EX_so4(e)                -0.2149

>> Optimal anaerobic solution

>> Optimal ethanol secretion rate solution
Done.
```

testSolveCobraLP

```
   Running dummyModel with solveCobraLP using glpk ...
 > [glpk] Optimality condition (1) in solveCobraLP satisfied.
 > [glpk] Optimality condition (2) in solveCobraLP satisfied.

 > [glpk] Optimality condition (1) in solveCobraLP satisfied.
 > [glpk] Optimality condition (2) in solveCobraLP satisfied.
Original LP has 2 rows, 2 columns, 4 non-zeros
Objective value = 600
OPTIMAL SOLUTION FOUND BY LP PRESOLVER

 > [glpk] Optimality condition (1) in solveCobraLP satisfied.
 > [glpk] Optimality condition (2) in solveCobraLP satisfied.
Done.
   Running ecoli with solveCobraLP using glpk ... Done.
   Running dummyModel with solveCobraLP using pdco ...
   ------------------------------------------------------------
   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 dense matrix

m        =        2     n        =        2      nnz(A)  =        4
max |b | =        1     max |x0| =  1.0e+00      xsize   =  1.0e+02
max |y0| =        0     max |z0| =  1.0e+00      zsize   =  1.0e+02

x0min    =        1     featol   =  1.0e-06      d1max   =  5.0e-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   =       20      show    =        0

Method   =        1     (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          2          2         2       0       0
  [0, bu]  [bl,  0]  excluding fixed variables
        2         0

Itn   mu stepx stepz  Pinf  Dinf  Cinf   Objective    nf  center     Chol
  0                    0.3   0.6   0.0 -5.9999997e+04        1.0
  1 -1.0 0.756 0.756  -0.3  -0.0  -0.4 -2.6962494e+04  1    30.9        3
  2 -1.0 0.078 0.078  -0.3  -0.0  -0.4 -2.5877439e+04  1  1164.7
  3 -1.0 0.147 0.147  -0.4  -0.1  -0.5 -2.2098922e+04  1   143.5
  4 -1.1 0.006 0.006  -0.4  -0.1  -0.5 -2.1984231e+04  1  6471.0
  5 -1.1 0.014 0.014  -0.4  -0.1  -0.5 -2.1261135e+04  1   434.8
  6 -1.1 0.019 0.019  -0.4  -0.1  -0.5 -1.7870918e+04  1  3621.6
  7 -1.1 1.000 1.000 -16.5  -8.5   0.4  3.7050583e+06  1   305.1
  8 -1.1 1.000 1.000 -17.8 -12.6  -1.0  3.6809979e+06  1     1.2
  9 -2.9 0.998 0.998 -17.8 -12.7  -2.7  3.6794412e+06  1     2.1
 10 -4.2 1.000 1.000 -17.8 -12.6  -4.2  3.6794068e+06  1     1.0
 11 -6.2 1.000 1.000 -18.3 -13.1  -6.2  3.6794056e+06  1     1.0
   Converged

max |x| =     0.010    max |y| = 38400.000    max |z| =  1728.333   scaled
max |x| =     1.000    max |y| =3840000.006    max |z| =172833.334 unscaled
PDitns  =        11   Cholitns =         0    cputime =       0.0

Distribution of vector     x         z
[  1e+05,  1e+06 )         0         2
[  1e+04,  1e+05 )         0         0
[  1e+03,  1e+04 )         0         0
[    100,  1e+03 )         0         0
[     10,    100 )         0         0
[      1,     10 )         0         0
[    0.1,      1 )         2         0
[   0.01,    0.1 )         0         0
[  0.001,   0.01 )         0         0
[      0,  0.001 )         0         0
Elapsed time is 0.018898 seconds.

 > [pdco] Optimality condition (1) in solveCobraLP satisfied.
 > [pdco] Optimality condition (2) in solveCobraLP satisfied.

   ------------------------------------------------------------
   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 dense matrix

m        =        2     n        =        2      nnz(A)  =        4
max |b | =        1     max |x0| =  1.0e+00      xsize   =  1.0e+02
max |y0| =        0     max |z0| =  1.0e+00      zsize   =  1.0e+02

x0min    =        1     featol   =  1.0e-06      d1max   =  5.0e-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   =       20      show    =        0
```

```
Method   =        1     (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          2          2         2       0       0
  [0, bu]  [bl,  0]  excluding fixed variables
        2         0

Itn   mu stepx stepz  Pinf  Dinf  Cinf   Objective    nf  center     Chol
  0                    0.3   0.6   0.0 -5.9999997e+04        1.0
  1 -1.0 0.756 0.756  -0.3  -0.0  -0.4 -2.6962494e+04  1    30.9        3
  2 -1.0 0.078 0.078  -0.3  -0.0  -0.4 -2.5877439e+04  1  1164.7
  3 -1.0 0.147 0.147  -0.4  -0.1  -0.5 -2.2098922e+04  1   143.5
  4 -1.1 0.006 0.006  -0.4  -0.1  -0.5 -2.1984231e+04  1  6471.0
  5 -1.1 0.014 0.014  -0.4  -0.1  -0.5 -2.1261135e+04  1   434.8
  6 -1.1 0.019 0.019  -0.4  -0.1  -0.5 -1.7870918e+04  1  3621.6
  7 -1.1 1.000 1.000 -16.5  -8.5   0.4  3.7050583e+06  1   305.1
  8 -1.1 1.000 1.000 -17.8 -12.6  -1.0  3.6809979e+06  1     1.2
  9 -2.9 0.998 0.998 -17.8 -12.7  -2.7  3.6794412e+06  1     2.1
 10 -4.2 1.000 1.000 -17.8 -12.6  -4.2  3.6794068e+06  1     1.0
 11 -6.2 1.000 1.000 -18.3 -13.1  -6.2  3.6794056e+06  1     1.0
   Converged

max |x| =     0.010    max |y| = 38400.000    max |z| =  1728.333   scaled
max |x| =     1.000    max |y| =3840000.006    max |z| =172833.334 unscaled
PDitns  =        11   Cholitns =         0    cputime =       0.0

Distribution of vector     x         z
[  1e+05,  1e+06 )         0         2
[  1e+04,  1e+05 )         0         0
[  1e+03,  1e+04 )         0         0
[    100,  1e+03 )         0         0
[     10,    100 )         0         0
[      1,     10 )         0         0
[    0.1,      1 )         2         0
[   0.01,    0.1 )         0         0
[  0.001,   0.01 )         0         0
[      0,  0.001 )         0         0
Elapsed time is 0.010743 seconds.

 > [pdco] Optimality condition (1) in solveCobraLP satisfied.
 > [pdco] Optimality condition (2) in solveCobraLP satisfied.

   ------------------------------------------------------------
   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 dense matrix

m        =        2     n        =        2      nnz(A)  =        4
max |b | =        1     max |x0| =  1.0e+00      xsize   =  1.0e+02
max |y0| =        0     max |z0| =  1.0e+00      zsize   =  1.0e+02

x0min    =        1     featol   =  1.0e-06      d1max   =  5.0e-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   =       20      show    =        0

Method   =        1     (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          2          2         2       0       0
  [0, bu]  [bl,  0]  excluding fixed variables
        2         0

Itn   mu stepx stepz  Pinf  Dinf  Cinf   Objective    nf  center     Chol
  0                    0.3   0.6   0.0 -5.9999997e+04        1.0
  1 -1.0 0.756 0.756  -0.3  -0.0  -0.4 -2.6962494e+04  1    30.9        3
  2 -1.0 0.078 0.078  -0.3  -0.0  -0.4 -2.5877439e+04  1  1164.7
  3 -1.0 0.147 0.147  -0.4  -0.1  -0.5 -2.2098922e+04  1   143.5
  4 -1.1 0.006 0.006  -0.4  -0.1  -0.5 -2.1984231e+04  1  6471.0
  5 -1.1 0.014 0.014  -0.4  -0.1  -0.5 -2.1261135e+04  1   434.8
  6 -1.1 0.019 0.019  -0.4  -0.1  -0.5 -1.7870918e+04  1  3621.6
  7 -1.1 1.000 1.000 -16.5  -8.5   0.4  3.7050583e+06  1   305.1
  8 -1.1 1.000 1.000 -17.8 -12.6  -1.0  3.6809979e+06  1     1.2
  9 -2.9 0.998 0.998 -17.8 -12.7  -2.7  3.6794412e+06  1     2.1
 10 -4.2 1.000 1.000 -17.8 -12.6  -4.2  3.6794068e+06  1     1.0
 11 -6.2 1.000 1.000 -18.3 -13.1  -6.2  3.6794056e+06  1     1.0
   Converged

max |x| =     0.010    max |y| = 38400.000    max |z| =  1728.333   scaled
max |x| =     1.000    max |y| =3840000.006    max |z| =172833.334 unscaled
PDitns  =        11   Cholitns =         0    cputime =       0.0

Distribution of vector     x         z
[  1e+05,  1e+06 )         0         2
[  1e+04,  1e+05 )         0         0
[  1e+03,  1e+04 )         0         0
[    100,  1e+03 )         0         0
[     10,    100 )         0         0
[      1,     10 )         0         0
[    0.1,      1 )         2         0
[   0.01,    0.1 )         0         0
[  0.001,   0.01 )         0         0
[      0,  0.001 )         0         0
Elapsed time is 0.019409 seconds.

 > [pdco] Optimality condition (1) in solveCobraLP satisfied.
 > [pdco] Optimality condition (2) in solveCobraLP satisfied.
Done.
   Running ecoli with solveCobraLP using pdco ... Done.

Testing model with linear constraint matrix that has 72 rows and 95 columns...
   Testing testDifferentLPSolvers using cplex_direct ... Done.
   Testing testDifferentLPSolvers using glpk ... Done.
   Testing testDifferentLPSolvers using gurobi ... Done.
   Testing testDifferentLPSolvers using ibm_cplex ... Done.
   Testing testDifferentLPSolvers using matlab ... Done.
   Testing testDifferentLPSolvers using mosek ... Done.
   Testing testDifferentLPSolvers using pdco ... Done.
   Testing testDifferentLPSolvers using quadMinos ... Done.
   Testing testDifferentLPSolvers using tomlab_cplex ... Done.
   Testing testDifferentLPSolvers using mosek_linprog ... Done.
```

```
   Testing testDifferentLPSolvers using dqqMinos ... Done.

 Summary:
              time            obj        y(rand)        w(rand)              solver                      a
  1       0.009595       0.873922       0.113308       0.091665                glpk
  2       0.031681       0.873922       0.113047       0.235520                pdco

Testing model with linear constraint matrix that has 2 rows and 2 columns...
   Testing testDifferentLPSolvers using cplex_direct ... Done.
   Testing testDifferentLPSolvers using glpk ... Done.
   Testing testDifferentLPSolvers using gurobi ... Done.
   Testing testDifferentLPSolvers using ibm_cplex ... Done.
   Testing testDifferentLPSolvers using matlab ... Done.
   Testing testDifferentLPSolvers using mosek ... Done.
   Testing testDifferentLPSolvers using pdco ...
Step lengths too smallDone.
   Testing testDifferentLPSolvers using quadMinos ... Done.
   Testing testDifferentLPSolvers using tomlab_cplex ... Done.
   Testing testDifferentLPSolvers using mosek_linprog ... Done.
   Testing testDifferentLPSolvers using dqqMinos ... Done.

 Summary:
              time            obj        y(rand)        w(rand)              solver                      a
  1       0.011557     600.000000      -0.000000    -200.000000                glpk
  2       0.009231     600.000000       0.000000    -200.000000                pdco

Testing model with linear constraint matrix that has 1 rows and 1 columns...
   Testing testDifferentLPSolvers using cplex_direct ... Done.
   Testing testDifferentLPSolvers using glpk ... Done.
   Testing testDifferentLPSolvers using gurobi ... Done.
   Testing testDifferentLPSolvers using ibm_cplex ... Done.
   Testing testDifferentLPSolvers using matlab ... Done.
   Testing testDifferentLPSolvers using mosek ... Done.
   Testing testDifferentLPSolvers using pdco ... Done.
   Testing testDifferentLPSolvers using quadMinos ... Done.
   Testing testDifferentLPSolvers using tomlab_cplex ... Done.
   Testing testDifferentLPSolvers using mosek_linprog ... Done.
   Testing testDifferentLPSolvers using dqqMinos ... Done.

 Summary:
              time            obj        y(rand)        w(rand)              solver                      a
  1       0.013073       1.000000       1.000000      -0.000000                glpk
  2       0.062806       1.000000       1.000000       0.000000                pdco
   Running optimalityConditions tests in solveCobraLP using pdco ...  Done.
   Running optimalityConditions tests in solveCobraLP using glpk ...  Done.
```

(Optional) Verify and test the entire COBRA Toolbox

TIMING ~ 30 min

Optionally test the functionality of The COBRA Toolbox locally, especially if one encounters an error running a function. The test suite runs tailored tests that verify the output and proper execution of core functions on the locally configured system. The full test suite can be invoked by typing:

```
testCOBRAToolbox=0;
if testCOBRAToolbox
    testAll
end
```

ANTICIPATED RESULTS

The test suite starts by initialising The COBRA Toolbox and thereafter, all of the tests are run. At the end of the test run, a comprehensive summary table is presented in which the respective tests and their test outcome is shown. On a properly configured system that is compatible with the most recent version of The COBRA Toolbox, all tests should pass.

TROUBLESHOOTING

If some third party dependencies are not properly installed, some tests may fail. The test suite, despite some tests failing, is not interrupted. The tests that fail are listed with a false status in the column Passed. The specific test can then be run individually to determine the exact cause of the error. If the error can be fixed, follow the tutorial on how to contribute to The COBRA Toolbox and contribute a fix.