ICON-GEMs

Authors: Thummarat Paklao[1], Apichat Suratanee[2], and Kitiporn Plaimas[3]

ICONGEMs (Integration of CO-expression Network into GEnome scale Metabolic models) [1] - an approach to integrating gene co-expression networks into FBA models, allowing for more accurate determination of flux distribution and functional pathways. By constructing a comprehensive gene co-expression network, we obtained a global perspective on the cell's mechanism. Using quadratic programming, we optimized the alignment between pairs of reaction fluxes and the correlation of their associated genes within the co-expression network.

$$\max \sum_{(i,j) \in R} q_i q_j = q^T A q$$

Subject to $\sum_{j=1}^{n+p} \bar{S}_{i,j} \bar{v}_j = 0,$

$0 \le \bar{v}_j \le f(g_j)$ for all $j = 1, 2, 3, \ldots, n+p,$

$$\sum_{j=1}^{n+p} \bar{c}_j \bar{v}_j \ge \alpha z^*$$

$$\sum_{(i,j) \in \mathrm{Re}} v_i v_j = 0$$

$$q_j = 1 + \frac{v_j}{M_j} \quad \text{for all } j = 1, 2, 3, \ldots, n+p,$$

where the matrix $\bar{S} = [S_{\mathrm{irr}} \;\; S_{\mathrm{rev}} \;\; S_{\mathrm{rev}}]^T$ includes submatrices S_{irr} and S_{rev}, which correspond to the columns of matrix S that represent irreversible and reversible reaction fluxes, respectively. The vector $\bar{v} = [v_{\mathrm{irr}} \;\; v_{\mathrm{rev}} \;\; -v_{\mathrm{rev}}]^T$ contains components of irreversibly and reversibly oriented fluxes, where v_{rev} represents the reversible component. The function $f(g_i)$ converts a gene expression value g_j into a corresponding flux bound value.

The vector $\bar{c} = [c_{\mathrm{irr}} \;\; c_{\mathrm{rev}} \;\; c_{\mathrm{rev}}]^T$ encompasses components of irreversible and reversible reaction fluxes, where c_{irr} and c_{rev} denote the irreversible and reversible fluxes, respectively. The vector $\bar{c}$ is initialized as zeros, with a single one placed at the position of the reaction of interest (biomass flux). The symbol z^* represents the potential maximum biomass predicted using the E-flux method [2]. The parameter $\alpha \in (0,1]$ determines the proportion of biomass required to evaluate the organisms' vitality, and in this study, α is set to 1. The set consists of reaction pairs derived from the same reversible reaction flux. The term represents the maximum gene expression value for reaction flux .

In this model, q_i and q_j represent the transformed flux values for reactions i and j, respectively. The set includes reaction pairs with genes linked in the co-expression network. The objective function is a summation of the products, specifically for reactions i and j corresponding to genes connected within the co-expression network.

REQUIREMENT

- Matlab (version 2018a or better)

- Cobra Toolbox

- Gurobi solver (version 9.0.1 or better, free academic)

- Gene expression profile in .csv file

(Note that the first column of gene expression data should have gene symbols/names used in the GPR association of the genome scale metabolic model. First row of gene expression data should have condition names.)

```
initCobraToolbox(false) % false, as we don't want to update
```

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

 > Checking if git is installed ...  Done (version: 2.40.1).
 > 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: --> set this path manually after installing the solver ( see instructions )
   - [*---] GUROBI_PATH: C:\gurobi1002\win64\matlab
   - [----] TOMLAB_PATH: --> set this path manually after installing the solver ( see instructions )
   - [----] MOSEK_PATH: --> set this path manually after installing the solver ( see instructions )
   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.
Warning: Cplex is not on the MATLAB path. Complete the installation as specified here: https://
opencobra.github.io/cobratoolbox/stable/installation.html#ibm-ilog-cplex
changeCobraSolver: problem initialising CPLEX object: Undefined function 'Cplex' for input arguments of ty
Could not find installation of ibm_cplex, so it cannot be tested
Could not find installation of tomlab_cplex, so it cannot be tested
Original LP has 1 row, 2 columns, 1 non-zero
Objective value = 0
OPTIMAL SOLUTION FOUND BY LP PRESOLVER

 > [glpk] Primal optimality condition in solveCobraLP satisfied.Could not find installation of mosek, so i
```

```
 > [matlab] Primal optimality condition 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 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 0.049409 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
Warning: Cplex is not on the MATLAB path. Complete the installation as specified here: https://
opencobra.github.io/cobratoolbox/stable/installation.html#ibm-ilog-cplex
changeCobraSolver: problem initialising CPLEX object: Undefined function 'Cplex' for input arguments of ty
Could not find installation of cplexlp, so it cannot be tested
Could not find installation of tomlab_snopt, so it cannot be tested
 Done.
 > Setting default solvers ...Could not find installation of mosek, so it cannot be tested
 Done.
 > Saving the MATLAB path ... Done.
   - The MATLAB path was saved in the default location.

 > Summary of available solvers and solver interfaces

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

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


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

> Checking for available updates ... skipped
removing: C:\Users\hp\cobratoolbox\src\analysis\thermo\componentContribution\new
removing: C:\Users\hp\cobratoolbox\src\analysis\thermo\groupContribution\new
removing: C:\Users\hp\cobratoolbox\src\analysis\thermo\inchi\new
removing: C:\Users\hp\cobratoolbox\src\analysis\thermo\molFiles\new
removing: C:\Users\hp\cobratoolbox\src\analysis\thermo\protons\new
removing: C:\Users\hp\cobratoolbox\src\analysis\thermo\trainingModel\new
```

Load the expression data that will be used for the simulation. For this tutorial, we have choosen to use E. coli Microarray-based gene expression data (downloaded from http://systemsbiology.ucsd.edu/InSilicoOrganisms/Ecoli/EcoliExpression2)

```
modelFileName = 'ecoli_core_model.mat';
```

```
modelDirectory = getDistributedModelFolder(modelFileName); %Look up the
folder for the distributed Models.
modelFileName= [modelDirectory filesep modelFileName]; % Get the full path.
Necessary to be sure, that the right model is loaded
model = readCbModel(modelFileName);
```

Load the expression data that will be used for the simulation. For this tutorial, we have chosen to use E. coli microarray-based gene expression data

```
fileGeneName = 'gene_exp.csv';
fileDir = fileparts(which(fileGeneName));
cd(fileDir);
[exp, genetxt] = xlsread([fileDir filesep fileGeneName]);
```

The integration of the co-expression network and metabolic model is completed using the function ICONGEMs. The inputs are: a loaded model file, an expression array (exp), genetxt, a row vector of conditions for calculating flux distribution (with the default set to all conditions), and a threshold for constructing the co-expression network (default value is 0.9). The alpha value represents the proportion of biomass (a value in the range (0,1], with the default set to 1).

```
solution = ICONGEMs(model, exp, genetxt);
```

Using optional inputs:

```
% set parameter
condition = 1:size(exp,2);
threashold = 0.9;
alpha = 0.99;
```

Call ICONGEMs function:

```
solution1 = ICONGEMs(model, exp, genetxt, condition, threshold,alpha);
```

After the algorithm is finished, the solution for the predicted metabolic fluxes will be added to the workspace. Numerical flux values can be examined in more detail by double-clicking the solution. Moreover, the output of this algorithm is reported in the `result.csv` file.

REFERENCES

1. Paklao, T., Suratanee, A. & Plaimas, K. ICON-GEMs: integration of co-expression network in genome-scale metabolic models, shedding light through systems biology. BMC Bioinformatics 24, 492 (2023). https://doi.org/10.1186/s12859-023-05599-0.

2. Colijn C, Brandes A, Zucker J, Lun DS, Weiner B, Farhat MR, Cheng T-Y, Moody DB, Murray M, Galagan JE. Interpreting expression data with metabolic flux models: predicting mycobacterium tuberculosis mycolic acid production. PLoS Comput Biol. 2009;5(8):e1000489.