Quadruple precision Flux Balance Analysis

Author(s): Ronan M.T. Fleming, University of Galway

Reviewer(s):

INTRODUCTION

In this tutorial, Flux Balance Analysis (FBA) is introduced using the E. coli core model, with functions in the COBRA Toolbox v3.0 [2].

Flux balance analysis is a solution to the optimisation problem

$$\begin{aligned} \max \quad & c^T v \\ \text{s.t.} \quad & Sv = b \\ & l \leq v \leq u \end{aligned}$$

where c is a vector of linear objective coefficients, S is an m times n matrix of stoichiometric coefficients for m molecular species involved in n reactions. l and u are n times 1 vectors that are the lower and upper bounds on the n times 1 variable vector v of reaction rates (fluxes). The optimal objective value is $c^T v^\star$ is always unique, but the optimal vector $v^\star$ is usually not unique.

In summary, the data is {c,S,l,u} and the variable being optimised is v.

TIMING

< 1 hrs

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. Please ensure that the COBRA Toolbox has been initialised (tutorial_initialize.mlx) and verify that the pre-packaged LP and QP solvers are functional (tutorial_verify.mlx).

PROCEDURE

Load a multiscale model

To load a model from a MAT-file, you can simply use the filename (with or without file extension).

```
if ~exist('model','var')
    load ME_matrix_GlcAer_WT.mat
    model = modelGlcOAer_WT;
    clear modelGlcOAer_WT;
end
```

Check the scaling properties of a stoichiometric matrix

The scaling properties of the stoichiometric matrix using:

```
[precisionEstimate, solverRecommendation] = checkScaling(model);
```

```
----------------------- Scaling summary report -----------------------

Estimation level:                           fine (scltol = 1.00)
Name of matrix:                             S
Size of matrix:
       * metabolites:                       68298
       * reactions:                         76664
Stoichiometric coefficients:
       * Minimum (absolute non-zero value):   5.50e-05
       * Maximum (absolute non-zero value):   8.01e+05
Lower bound coefficients:
       * Minimum (absolute non-zero value):   1.00e+06
       * Maximum (absolute non-zero value):   1.00e+09
Upper bound coefficients:
       * Minimum (absolute non-zero value):   5.54e+00
       * Maximum (absolute non-zero value):   1.00e+09
Row scaling coefficients:
       * Minimum:                           1.67e-03 (row #: 24548)
       * Maximum:                           7.90e+04 (row #: 66206)
Column scaling coefficients:
       * Minimum:                           3.47e-04 (column #: 130)
       * Maximum:                           1.13e+04 (column #: 24943)

--------------------------------- Ratios -------------------------------

Ratio of stoichiometric coefficients:         1.46e+10
Order of magnitude diff. (stoich. coeff.):    10

Ratio of lower bounds:                        1.00e+03
Order of magnitude diff. (lower bounds):      3

Ratio of upper bounds:                        1.80e+08
Order of magnitude diff. (upper bounds):      8

Ratio of row scaling coefficients:            4.72e+07
Order of magnitude diff. (row scaling):       7

Ratio of column scaling coefficients:         3.27e+07
Order of magnitude diff. (column scaling):    7

----------------------------------------------------------------------

-> The model has badly scaled rows and columns. Quad precision is strongly recommended.

   Set the Quad MINOS solver with: >> changeCobraSolver('quadMinos', 'LP')
```

If `precisionEstimate=='quad'` then the "model has badly scaled rows and colums" then a quad precision solver is required.

```
precisionEstimate
```

```
precisionEstimate =
'quad'
```

Quad precision solvers come installed by default in COBRA Toolbox v3 so solverRecommendation should include {'dqqMinos'} and {'quadMinos'}

```
solverRecommendation
```

```
solverRecommendation = 1×2 cell
'dqqMinos'  'quadMinos'
```

Change to a solution approach that combines double and quad precision to try to solve the LP problem. If they are properly installed then `solverOK == 1`

```
[solverOK, solverInstalled] = changeCobraSolver('dqqMinos','LP')
```

```
 > Solver dqqMinos is installed but not working properly.
solverOK = logical
   0
solverInstalled = logical
   1
```

```
%[solverOK, solverInstalled] = changeCobraSolver('quadMinos','LP')
```

Further information on numerical characterisation of COBRA models can be found in `tutorial_numCharact.mlx`

Solve a quad FBA problem

Solve a FBA problem in quad precision using `optimizeCbModel`

```
if solverOK
    FBAsolution = optimizeCbModel(model,'max');
end
```

TROUBLESHOOTING

Always check the value of `solution.stat`, which returns the status of the solution, even when using quad precision.

`solution.stat == 1` means the FBA problem is solved successfully. Anything else and there is a problem.

Although it does not happen often, there are many reasons why an FBA problem might not solve, so they are divided into three categories.

```
%      solution.stat - Solver status in standardized form:
%                      * `-1` - No solution reported (timelimit, numerical
problem etc)
%                      * `1` - Optimal solution
%                      * `2` - Unbounded solution
%                      * `0` - Infeasible
```

`solution.stat == 0` means that the problem is overconstrainted and no feasible flux vector v exists. The constraints need to be relaxed before the problem will solve. See tutorial_relaxedFBA.mlx

`solution.stat = 2` means that the problem is underconstrained to the extent that the possible optimal value of the objective is unbounded, that is infinity, or minus infinity. This means that extra constraints need to be added, e.g., lower and upper bounds on the reaction rates.

`solution.stat = 1` means that the problem is more complicated than either of the above. It could be that the problem does, in principle, have a solution, but that the current solver cannot find one, so an industrial quality solver should be tested, e.g., gurobi. It could also mean that the FBA problem is poorly scaled so there are numerical problems solving it, or it could also be just slightly infeasible, in which case a higher precision solver will be required to solve the problem, e.g., a quadruple precision solver. The way each solver reports the nature of the problem varies between solvers, so checking `solution.origStat` against the documentation that comes with each solver is necessary to figure out what the potential solution is.

```
%    solution.stat - Solver status in standardized form:
%                       * `-1` - No solution reported (timelimit, numerical
problem etc)
%                       * `1` - Optimal solution
%                       * `2` - Unbounded solution
%                       * `0` - Infeasible
```


Acknowledgments

Prof. Michael A. Saunders at Stanford University is responsible for the quad precision solvers that are included in the COBRA toolbox.

REFERENCES

[1] P. E. Gill, W. Murray, M. A. Saunders and M. H. Wright (1987). Maintaining LU factors of a general sparse matrix, Linear Algebra and its Applications 88/89, 239-270.

[2] Multiscale modeling of metabolism and macromolecular synthesis in E. coli and its application to the evolution of codon usage, Thiele et al., PLoS One, 7(9):e45635 (2012).

[3] D. Ma, L. Yang, R. M. T. Fleming, I. Thiele, B. O. Palsson and M. A. Saunders, Reliable and efficient solution of genome-scale models of Metabolism and macromolecular Expression, Scientific Reports 7, 40863; doi: \url{10.1038/srep40863} (2017). http://rdcu.be/oCpn.

[4]. Laurent Heirendt & Sylvain Arreckx, Thomas Pfau, Sebastian N. Mendoza, Anne Richelle, Almut Heinken, Hulda S. Haraldsdottir, Jacek Wachowiak, Sarah M. Keating, Vanja Vlasov, Stefania Magnusdottir, Chiam Yu Ng, German Preciat, Alise Zagare, Siu H.J. Chan, Maike K. Aurich, Catherine M. Clancy, Jennifer Modamio, John T. Sauls, Alberto Noronha, Aarash Bordbar, Benjamin Cousins, Diana C. El Assal, Luis V. Valcarcel, Inigo Apaolaza, Susan Ghaderi, Masoud Ahookhosh, Marouen Ben Guebila, Andrejs Kostromins, Nicolas Sompairac, Hoai M. Le, Ding Ma, Yuekai Sun, Lin Wang, James T. Yurkovich, Miguel A.P. Oliveira, Phan T. Vuong, Lemmer P. El Assal, Inna Kuperstein, Andrei Zinovyev, H. Scott Hinton, William A. Bryant, Francisco J. Aragon Artacho, Francisco J. Planes, Egils Stalidzans, Alejandro Maass, Santosh Vempala, Michael Hucka, Michael A. Saunders, Costas D. Maranas, Nathan E. Lewis, Thomas Sauter, Bernhard Ø. Palsson, Ines Thiele, Ronan M.T. Fleming, **Creation and analysis of biochemical constraint-based models: the COBRA Toolbox v3.0**, Nature Protocols, volume 14, pages 639–702, 2019 doi.org/10.1038/s41596-018-0098-2.