

Proton shuttle testing with sparse flux balance analysis

Author: Ronan Fleming, Ines Thiele, University of Luxembourg.

Reviewer:

INTRODUCTION

We consider a biochemical network of m molecular species and n biochemical reactions. The biochemical network is mathematically represented by a stoichiometric matrix $S \in \mathbb{R}^{m \times n}$. In standard notation, flux balance analysis (FBA) is the linear optimisation problem

$$\begin{aligned} \text{min}_v \quad & p(v) = c^T v \\ \text{s.t.} \quad & Sv = \Delta, \\ & l \leq v \leq u, \end{aligned}$$

where $c \in \mathbb{R}^n$ is a parameter vector that linearly combines one or more reaction fluxes to form what is termed the objective function, and where a $b_i < 0$, or $b_i > 0$, represents some fixed output, or input, of the i th molecular species. A typical application of flux balance analysis is to predict an optimal non-equilibrium steady-state flux vector that optimises a linear objective function, such as biomass production rate, subject to bounds on certain reaction rates.

In this tutorial, we demonstrate how to predict the minimal number of active reactions that are still consistent with an optimal objective derived from the result of a standard flux balance analysis problem. In each case, the corresponding problem is a cardinality minimisation problem that we term sparse flux balance analysis

$$\begin{aligned} \text{min}_v \quad & \|v\|_0 \\ \text{s.t.} \quad & Sv = b \\ & l \leq v \leq u \\ & c^T v = p^* \end{aligned}$$

where the last constraint is optional and represents the requirement to satisfy an optimal objective value p^* derived from any solution to a flux balance analysis (FBA) problem. This approach is used to check for minimal sets of reactions that either should be active, or should not be active in a flux balance model that is representative of a biochemical network.

In particular, this tutorial illustrates the use of sparse flux balance analysis to compute the minimal set of reactions that must be active to produce ATP

TIMING

A minimal solution to sparse flux balance analysis problem can be obtained in <10 seconds. The time consuming part is comparing the predictions with the biochemical literature to assess whether the predictions are consistent with biochemical network function or not, as such, the process of refining a model to increase its biochemical fidelity can take days or weeks.

PROCEDURE

Loading and examining the properties of Recon3.0model

We are going to focus here on testing the biochemical fidelity of Recon3.0model, so load it, unless it is already loaded into the workspace

```
clear %model
if ~exist('modelOrig','var')
    filename='Recon3.0';
    filename='Recon2.0';
    filename='Recon2.0model';
    filename='Recon3.0model';
    filename='H990.0';
    filename='Recon2.2model';
    filename='Recon3.0';
    filename='Recon3.0model';
    directory='~/work/ibg/load/programReconstruction/projects/recon2models/data/recon3comparisonModels';
    model = loadIdentifiedModel(filename,directory);
    model.csense(1:size(model.S,1),1)='E';
    modelOrig = model;
else
    model=modelOrig;
end
```

Setting the numerical tolerance

Implementation of sparse flux balance analysis with any numerical optimisation solver, requires a tolerance to be set that distinguished between zero and non-zero flux, based on the numerical tolerance of the currently installed optimisation solver. Typically 1e-6 will suffice, except for multiscale models.

```
feasTol = getCobraSolverParams('LP', 'feasTol');
```

Testing for activity of ATP synthase with all exchanges closed

Detect the ATP synthase reaction and if there is none already, add one.

```
atpsynthaseIsOstrcrap(model.rxns,'ATPS4m') | strcrap(model.rxns,'ATPS4m');% | strcrap(model.rxns,'ATPM');
if ~any(atpsynthaseIsO)
    fprintf('Could not find ATP synthase reaction, adding one.')
    if ~strcrap(filename,'HMR2.0')
        %model = addReaction(model, 'ATPMnew', 'h2o[c] + atp[c] -> h[c] + adp[c] + pi[c]');
        model = addReaction(model, 'ATPS4m', '4.0 h[c] + adp[m] + pi[m] -> h2o[m] + 3.0 h[m] + atp[m]');
    else
        %model = addReaction(model, 'ATPMnew', 'm02040c + m01271c -> m02039c + m01205c + m02751c');
        model = addReaction(model, 'ATPS4m', '4.0 m02039c + m01205m + m02751m -> m02040m + 3.0 m02039m + m01271m');
    end
    atpsynthaseIsOstrcrap(model.rxns,'ATPS4m');
    fprintf('%s %s\n',model.rxns{atpsynthaseIsO}, ' is the ATP synthase reaction')
else
    fprintf('%s %s\n',model.rxns{atpsynthaseIsO}, ' is the ATP synthase reaction')
end
```

Display the size of the model

```
[nMet,nRxn] = size(model.S);
fprintf('%6s,%6s\n','nMet','nRxn'); fprintf('%6s,%6s,%6s\n',nMet,nRxn,' totals in ', model.modelID)
```

Display the constraints

```
minInf = -1000;
maxInf = 1000;
printConstraints(model, minInf, maxInf);
```

Identify the exchange reaction(s) heuristically

```
if ~isfield(model,'SIntRxnIsO')
    model = findSIntRxnInd(model,size(model.S,1),1);
end
```

Maximise the atp synthase reaction

```
model.c{:}=@;
model.c{atpsynthaseIsO}=1;
osenseStr='max';
```

Choose to minimize the zero norm of the optimal flux vector

```
isfWorm='zero';
```

Allow thermodynamically infeasible fluxes

```
allowLoops=1;
```

Select the approximate step functions when minimizing the zero norm of the flux vector

```
% zeroNormApprox='cappedL1';% Capped-L1 norm
% zeroNormApprox='exp';%Exponential function
% zeroNormApprox='log';%Logarithmic function
% zeroNormApprox='SCAD';%Smoothly clipped absolute deviation function
% zeroNormApprox='lp-';%L_p norm with p=0
% zeroNormApprox='lp-';%L_p norm with 0<p<1
zeroNormApprox='all';% test all approximations available and return the best one
```

Close all external reactions

```
model.lb{~model.SIntRxnIsO}=0;
model.ub{~model.SIntRxnIsO}=0;
```

Run sparse flux balance analysis on the model with all exchanges closed

```
sparseFBAbsoluteBounded = optimizeCSModel(model, osenseStr, minfWorm, allowLoops, zeroNormApprox);
```

Check to see if there is a non-zero flux through the ATP synthase reaction

```
fprintf('%g\n',sparseFBAbsoluteBounded.v{atpsynthaseIsO}, ' flux through the ATP synthase reaction')
```

Display the sparse flux solution, but only the non-zero fluxes, above a specified cutoff.

```
cutoffFBAvecSel;
for m=1:nRxn
    if abs(sparseFBAbsoluteBounded.v{m})>cutoff
```

```

        formulaPrintKofFormula(model, model.rxns{n}, 0);
        fprintf('%10g%15s%10g\n', sparseFBA_solutionUnbounded.v(n), model.rxns{n}, formula{1});
    end
end

```

ANTICIPATED RESULTS

In a model for flux balance analysis, there should be zero flux through the ATP synthase reaction if all external reaction bounds are zero.

TROUBLESHOOTING

If there is non-zero flux through the ATP synthase reaction, with all external reaction bounds zero, then the bounds on one of the reactions in each of the minimal sets needs to be set to eliminate flux in one direction. Each of the minimal sets corresponds to net flux around a stoichiometrically balanced cycle, which is thermodynamically infeasible ([Bramson, et al. 2012](#)). Steady-state mass balance constraints do not enforce thermodynamic constraints. In lieu of such constraints, the bounds on reactions can be set based on the biochemical literature to eliminate net flux around a stoichiometrically balanced cycle. In a model, with all external reactions blocked (bounds are set to zero), maximising the ATP synthase reaction while minimising the cardinality of all internal reactions, using sparse flux balance analysis can be used to find any such cycle of minimal cardinality (minimal number of active reactions). By further constraining the bounds to convert one reversible reaction in each such cycle to an irreversible reaction, thermodynamically infeasible flux around cycles, such as those involving the ATP synthase reaction, can be eliminated. The following sections of this tutorial illustrate how to test different parts of the model for thermodynamically infeasible flux through the ATP synthase reaction.

Testing for activity of ATP synthase with all exchanges closed and all internal reactions reversible

Fully open all internal reactions

```

model.lb(model.SIntRxnIdx)=1000;
model.ub(model.SIntRxnIdx)=1000;

```

Run sparse flux balance analysis on the model with all exchanges closed and all internal reactions reversible

```

sparseFBA_solutionUnbounded = optimizeCbModel(model, osenseStr, minNorm, allowLoops, zeroNormApprox);

```

Check to see if there is a non-zero flux through the ATP synthase reaction

```

fprintf('%10g\n', sparseFBA_solutionUnbounded.v(atpSynthaseIdx), ' flux through the ATP synthase reaction')

```

Display the sparse flux solution, but only the non-zero fluxes, above a specified cutoff.

```

cutoff=fevalTel;
for n=1:nRxn
    if abs(sparseFBA_solutionUnbounded.v(n))>cutoff
        formulaPrintKofFormula(model, model.rxns{n}, 0);
        fprintf('%10g%15s%10g\n', sparseFBA_solutionUnbounded.v(n), model.rxns{n}, formula{1});
    end
end

```

ANTICIPATED RESULTS

In a model for flux balance analysis, there might be non-zero flux through the ATP synthase reaction if all external reaction bounds are zero and all internal reactions reversible. This indicates the importance of appropriately constrained internal reaction bounds.

Testing for activity of ATP synthase with all exchanges closed and all transport reactions reversible

Identify all of the transport reactions

```

allTransportRxnIdx=transportReactionIdx(model);
fprintf('%10g\n', nnz(allTransportRxnIdx), ' transport reactions in total. ');

```

Revert to original Recon3D model reaction bounds

```

model.lb=modelOrig.lb;
model.ub=modelOrig.ub;

```

Open all transport reactions (which might include an external reaction, e.g., a biomass reaction)

```

model.lb(allTransportRxnIdx)=1000;
model.ub(allTransportRxnIdx)=1000;

```

Close all external reactions

```

model.lb(~model.SIntRxnIdx)=0;
model.ub(~model.SIntRxnIdx)=0;

```

Run sparse flux balance analysis on the model with all exchanges closed

```

sparseFBA_solutionUnbounded = optimizeCbModel(model, osenseStr, minNorm, allowLoops, zeroNormApprox);

```

Check to see if there is a non-zero flux through the ATP synthase reaction

```
fprintf(' %g\n', sparseFBAsolutionBounded.v(atpsynthaseRxn), ' flux through the ATP synthase reaction')
```

Display the sparse flux solution, but only the non-zero fluxes, above a specified cutoff.

```
cutoff=feaxTol;  
for m=1:nRun  
    if abs(sparseFBAsolutionBounded.v(n))>cutoff  
        formula=printRxnFormula(model, model.rxn{n}, 0);  
        fprintf(' %18g %15g %15g\n', sparseFBAsolutionBounded.v(n), model.rxn{n}, formula{1});  
    end  
end
```

Testing for activity of ATP synthase with all exchanges closed and all mitochondrial transport reactions reversible

Identify all of the transport reactions involving the cytoplasm and mitochondrial matrix

```
originCompartments='c';  
destinationCompartments='m';  
unidirectionalRxn=0;  
cATransportRxn=0; for i=1:length(model.transportReactions)  
    if strcmp(model.transportReactions{i}.compartment,originCompartments) && strcmp(model.transportReactions{i}.compartment,destinationCompartments)  
        unidirectionalRxn=i;  
    end  
end  
fprintf(' %18g\n', sum(cATransportRxn), ' transport reactions involving the cytoplasm and mitochondrial matrix.');
```

Revert to original Recon3D model reaction bounds

```
model.lb=modelOrig.lb;  
model.ub=modelOrig.ub;
```

Open all transport reactions (which might include an external reaction, e.g., a biomass reaction)

```
model.lb(cATransportRxn)=1000;  
model.ub(cATransportRxn)=1000;
```

Close all external reactions

```
model.lb(-model.SIntRxn)=0;  
model.ub(-model.SIntRxn)=0;
```

Run sparse flux balance analysis on the model

```
sparseFBAsolutionBounded = optimizeCbModel(model, osenseStr, minfuns, allowLoops, zeroFunsApprox);
```

Check to see if there is a non-zero flux through the ATP synthase reaction

```
fprintf(' %g\n', sparseFBAsolutionBounded.v(atpsynthaseRxn), ' flux through the ATP synthase reaction')
```

Display the sparse flux solution, but only the non-zero fluxes, above a specified cutoff.

```
cutoff=feaxTol;  
for m=1:nRun  
    if abs(sparseFBAsolutionBounded.v(n))>cutoff  
        formula=printRxnFormula(model, model.rxn{n}, 0);  
        fprintf(' %18g %15g %15g\n', sparseFBAsolutionBounded.v(n), model.rxn{n}, formula{1});  
    end  
end
```

Testing for activity of ATP synthase with all exchanges closed and all plasma membrane transport reactions reversible

Identify all of the transport reactions across the plasma membrane

```
originCompartments='e';  
destinationCompartments='c';  
unidirectionalRxn=0;  
ecTransportRxn=0; for i=1:length(model.transportReactions)  
    if strcmp(model.transportReactions{i}.compartment,originCompartments) && strcmp(model.transportReactions{i}.compartment,destinationCompartments)  
        unidirectionalRxn=i;  
    end  
end  
fprintf(' %18g\n', sum(ecTransportRxn), ' transport reactions across the plasma membrane.');
```

Revert to original Recon3D model reaction bounds

```
model.lb=modelOrig.lb;  
model.ub=modelOrig.ub;
```

Open all transport reactions (which might include an external reaction, e.g., a biomass reaction)

```
model.lb(ecTransportRxn)=1000;  
model.ub(ecTransportRxn)=1000;
```

Close all external reactions

```
model.lb(-model.SIntKxnKool)n#;
model.ub(-model.SIntKxnKool)n#;
```

Run sparse flux balance analysis on the model

```
sparseFBAsoLutIionBounded = optimizeCModel(model, osenseStr, minNorm, allowLoops, zeroNormApprox);
```

Check to see if there is a non-zero flux through the ATP synthase reaction

```
fprintff("%g\n", sparseFBAsoLutIionBounded.v(atpSynthaseIool), " flux through the ATP synthase reaction")
```

Display the sparse flux solution, but only the non-zero fluxes, above a specified cutoff.

```
cutoffFeasTol;
for nI:nKxn
    if abs(sparseFBAsoLutIionBounded.v(n))>cutoff
        formulaPrintKnfFormula(model, model.rxn{n}, 0);
        fprintff("%10g %15s %10g\n", sparseFBAsoLutIionBounded.v(n), model.rxn{n}, formula{1});
    end
end
```

Testing for activity of ATP synthase with all exchanges closed and peroxisomal transport reactions reversible

Identify all of the transport reactions across the plasma membrane

```
originCompartment='c';
destinationCompartment='x';
unidirectionalIool;
cxTransportKxnIool=transportReactionIool(model, originCompartment, destinationCompartment, unidirectionalIool);
fprintff("%s\n", nnc(cxTransportKxnIool), ' transport reactions across the peroxisome membrane.');
```

Revert to original Raco0.0 model reaction bounds

```
model.lbmodelOrig.lb;
model.ubmodelOrig.ub;
```

Open all transport reactions (which might include an external reaction, e.g., a biomass reaction)

```
model.lb(cxTransportKxnIool)n-I000;
model.ub(cxTransportKxnIool)nI000;
```

Close all external reactions

```
model.lb(-model.SIntKxnKool)n#;
model.ub(-model.SIntKxnKool)n#;
```

Run sparse flux balance analysis on the model

```
sparseFBAsoLutIionBounded = optimizeCModel(model, osenseStr, minNorm, allowLoops, zeroNormApprox);
```

Check to see if there is a non-zero flux through the ATP synthase reaction

```
fprintff("%g\n", sparseFBAsoLutIionBounded.v(atpSynthaseIool), " flux through the ATP synthase reaction")
```

Display the sparse flux solution, but only the non-zero fluxes, above a specified cutoff.

```
cutoffFeasTol;
for nI:nKxn
    if abs(sparseFBAsoLutIionBounded.v(n))>cutoff
        formulaPrintKnfFormula(model, model.rxn{n}, 0);
        fprintff("%10g %15s %10g\n", sparseFBAsoLutIionBounded.v(n), model.rxn{n}, formula{1});
    end
end
```

Testing for activity of ATP synthase with all exchanges closed and lysosomal transport reactions reversible

Identify all of the transport reactions across the plasma membrane

```
originCompartment='c';
destinationCompartment='l';
unidirectionalIool;
clTransportKxnIool=transportReactionIool(model, originCompartment, destinationCompartment, unidirectionalIool);
fprintff("%s\n", nnc(clTransportKxnIool), ' transport reactions across the lysosomal membrane.');
```

Revert to original Raco0.0 model reaction bounds

```
model.lbmodelOrig.lb;
model.ubmodelOrig.ub;
```

Open all transport reactions (which might include an external reaction, e.g., a biomass reaction)

```
model.lb(cLTtransportRxnIdx)=1000;  
model.ub(cLTtransportRxnIdx)=1000;
```

Close all external reactions

```
model.lb(-model.SIntRxnIdx)=0;  
model.ub(-model.SIntRxnIdx)=0;
```

Run sparse flux balance analysis on the model

```
sparseFBAbsoluteBounded = optimizeCbModel(model, osenseStr, minfuns, allowLoops, zeroRxnApprox);
```

Check to see if there is a non-zero flux through the ATP synthase reaction

```
fprintf(' %g\n', sparseFBAbsoluteBounded.v(atpsynthaseIdx), ' flux through the ATP synthase reaction')
```

Display the sparse flux solution, but only the non-zero fluxes, above a specified cutoff.

```
cutoff=feval(f1;  
for n=1:nRxn  
    if abs(sparseFBAbsoluteBounded.v(n))>cutoff  
        formula=printRxnFormula(model, model.rxn{n}, 0);  
        fprintf(' %18g %15s %14-6s\n', sparseFBAbsoluteBounded.v(n), model.rxn{n}, formula{1});  
    end  
end
```

REFERENCES

[Fleming_cardinality_node] Fleming, R.M.T., et al., Cardinality optimisation in constraint-based modelling: illustration with Recon 3D (submitted), 2017.

[Le et al 2015] Le Thi, H.A., Pham Dinh, T., Le, H.M., and Vo, X.T. (2015). DC approximation approaches for sparse optimization. European Journal of Operational Research 244, 26–46.