

Subset of a model that admits a thermodynamically consistent flux

```
[solverOK,solverInstalled]=changeCobraSolver('ibm_cplex','all');
```

```
> changeCobraSolver: IBM ILOG CPLEX interface added to MATLAB path.  
> ibm_cplex (version 1210) is compatible and fully tested with MATLAB R2019a on your operating system.  
> changeCobraSolver: Solver for LP problems has been set to ibm_cplex.  
  
> changeCobraSolver: IBM ILOG CPLEX interface added to MATLAB path.  
> ibm_cplex (version 1210) is compatible and fully tested with MATLAB R2019a on your operating system.  
> changeCobraSolver: Solver for MILP problems has been set to ibm_cplex.  
  
> changeCobraSolver: IBM ILOG CPLEX interface added to MATLAB path.  
> ibm_cplex (version 1210) is compatible and fully tested with MATLAB R2019a on your operating system.  
> changeCobraSolver: Solver for QP problems has been set to ibm_cplex.  
  
> changeCobraSolver: IBM ILOG CPLEX interface added to MATLAB path.  
> ibm_cplex (version 1210) is compatible and fully tested with MATLAB R2019a on your operating system.  
> changeCobraSolver: Solver for MIQP problems has been set to ibm_cplex.  
> changeCobraSolver: Solver ibm_cplex not supported for problems of type NLP. Currently used: matlab
```

```
%[solverOK,solverInstalled]=changeCobraSolver('gurobi','all');  
%[solverOK,solverInstalled]=changeCobraSolver('ibm_cplex','QP');
```

Load model

```
modelToLoad='circularToy';  
modelToLoad='ecoli_core';  
modelToLoad='modelRecon3MitoOpen';  
modelToLoad='Recon3DModel';  
%modelToLoad='iDopa';
```

Load a model

```
driver_thermoModelLoad
```

```
Model loaded: Recon3DModel  
lower bounds greater than zero  
Internal stoichiometric nullspace computed in 0.65154 seconds.
```

Stoichiometric consistency

```
if ~isfield(model,'SConsistentRxnBool') ||  
~isfield(model,'SConsistentMetBool')  
    massBalanceCheck=0;  
    %massBalanceCheck=1;  
    printLevel=2;  
    [SConsistentMetBool, SConsistentRxnBool,  
SInConsistentMetBool, SInConsistentRxnBool, unknownSConsistencyMetBool,  
unknownSConsistencyRxnBool, model,stoichConsistModel]...  
    = findStoichConsistentSubset(model, massBalanceCheck, printLevel);  
else  
    %Extract stoich consistent submodel  
    if any(~model.SConsistentMetBool)
```

```

rxnRemoveMethod='inclusive';%maintains stoichiometric consistency
[stoichConsistModel, rxnRemoveList] = removeMetabolites(model,
model.mets(~model.SConsistentMetBool),rxnRemoveMethod);
SConsistentRxnBool2=~ismember(model.rxns,rxnRemoveList);
if ~all(model.SConsistentRxnBool==SConsistentRxnBool2)
    error('inconsistent reaction removal')
end
try
    stoichConsistModel = removeUnusedGenes(stoichConsistModel);
catch ME
    disp(ME.message)
end
else
    stoichConsistModel = model;
end
end

[nMet,nRxn]=size(stoichConsistModel.S)

```

```

nMet =
    5835
nRxn =
    10600

```

Flux consistency

```

fluxConsistentParam.method='fastcc';%can handle additional constraints
fluxConsistentParam.printLevel=1;
[~,~,~,~,stoichConsistModel]=
findFluxConsistentSubset(stoichConsistModel,fluxConsistentParam);

```

Extract flux consistent submodel

```

if any(~stoichConsistModel.fluxConsistentRxnBool)
    rxnRemoveList =
stoichConsistModel.rxns(~stoichConsistModel.fluxConsistentRxnBool);
    stoichFluxConsistModel = removeRxns(stoichConsistModel,
rxnRemoveList,'metRemoveMethod','exclusive','ctrsRemoveMethod','inclusive');
    try
        stoichFluxConsistModel = removeUnusedGenes(stoichFluxConsistModel);
    catch ME
        disp(ME.message)
    end
else
    stoichFluxConsistModel = stoichConsistModel;
end
[nMet,nRxn]=size(stoichFluxConsistModel.S)

```

```

nMet =
    5835
nRxn =
    10600

```

Forced reactions

```
forcedRxnBool = model.lb>0 | model.ub<0;  
nForcedRxn = nnz(forcedRxnBool)
```

```
nForcedRxn =  
    0
```

```
printConstraints(model,[],[],forcedRxnBool)  
model.lb(strcmp(model.rxns,'biomass_reaction'))=0;  
return
```

Thermodynamic consistency

```
%save('debug_prior_to_findThermoConsistentFluxSubset.mat')  
%return  
param.printLevel = 1;  
param.acceptRepairedFlux=1;  
param.relaxBounds=1;  
[thermoFluxConsistentMetBool,thermoFluxConsistentRxnBool,stoichFluxConsistModel,stoichFluxThermoConsistModel] =  
findThermoConsistentFluxSubset(stoichFluxConsistModel,param);
```

Size of the largest flux, stoich and thermo consistent submodel

```
[nMet,nRxn]=size(stoichFluxThermoConsistModel.S)
```

Nullspace

Nullspace is necessary for backup check of thermodynamic consistency using thermoFlux2QNty

```
[stoichFluxThermoConsistModel,rankK,nnzK,timeTaken] =  
internalNullspace(stoichFluxThermoConsistModel);  
rankK
```

Minimal thermodynamically consistent submodel

Compute the minimal thermodynamically consistent submodel

```
[minimalModel, modelThermoMetBool, modelThermoRxnBool] =  
thermoKernel(stoichFluxThermoConsistModel);  
[nMet,nRxn]=size(minimalModel.S)
```

Data to define a thermodynamically consistent subnetwork

Setup random data to select a random subset

```
param.n=200;  
[rankMetConnectivity,rankMetInd,rankConnectivity] =  
rankMetabolicConnectivity(stoichFluxThermoConsistModel,param);
```

```

[nMet,nRxn]=size(stoichFluxThermoConsistModel.S);
rxnWeights=rand(nRxn,1)-0.5;
rxnWeights(stoichFluxThermoConsistModel.SConsistentRxnBool)=0;

coreRxnBool=rxnWeights<0.45;
removeRxnBool=rxnWeights>0.48;
rxnWeights(rxnWeights>0.4)=1;
rxnWeights(rxnWeights<-0.4)=-1;
rxnWeights(rxnWeights>=-0.4 & rxnWeights<=0.4)=0;
hist(rxnWeights)
metWeights=rand(nMet,1)-0.5;
metWeights(rankMetInd(1:200))=0;
coreMetBool=metWeights<0.45;
removeMetBool=metWeights>0.5;
metWeights(metWeights>0.4)=1;
metWeights(metWeights<-0.4)=-1;
metWeights(metWeights>=-0.4 & metWeights<=0.4)=0;
hist(metWeights)

```

Remove inactive reactions and absent metabolites

```

param.printLevel = 1;
[solverOK,solverInstalled]=changeCobraSolver('gurobi','QP');
[thermoFluxConsistentMetBool,thermoFluxConsistentRxnBool,stoichFluxThermoConsistModel,stoichFluxThermoConsistModelRed] =
findThermoConsistentFluxSubset(stoichFluxThermoConsistModel, param,
removeMetBool, removeRxnBool);
[nMet,nRxn]=size(stoichFluxThermoConsistModelRed.S)

```