Build Atom Transition Multigraph

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

Reviewer(s):

INTRODUCTION

Given a metabolic model, this tutorial builds an atom transition multgraph as described elsewhere [1,2].

These tutorials should generally be used in the following order:

1. Initialise and set the paths to inputs and outputs

COBRA.tutorials/driver_initConservedMoietyPaths.mlx

2. Build an atom transition graph

tutorial_buildAtomTransitionMultigraph.mlx

3. Identify conserved moieties, given an atom transition graph

tutorial_identifyConservedMoieties.mlx

4. Analyse the output of #3

tutorial_analyseConservedMoieties.mlx

5. Prepare for visualisation of individual conserved moieties (beta)

tutorial_visualiseConservedMoieties.mlx

Define the model that will be used for conserved moiety decomposition

```
if ~exist('modelName','var')
    %modelName = 'centralMetabolism_fastCore';
    %modelName = 'iDopaNeuroC';
    modelName = 'DAS'
end
```

Set the paths

```
driver_initConservedMoietyPaths
```

```
projectDir =
'/home/rfleming/work/sbgCloud/code/fork-COBRA.tutorials/analysis/conservedMoieties'
dataDir =
'/home/rfleming/work/sbgCloud/code/fork-COBRA.tutorials/analysis/conservedMoieties/data'
softwareDir =
'/home/rfleming/work/sbgCloud/code/fork-COBRA.tutorials/analysis/conservedMoieties/software'
visDataDir =
```

```
'/home/rfleming/work/sbgCloud/code/fork-COBRA.tutorials/analysis/conservedMoieties/data/visualisation'
resultsDir =
'/home/rfleming/work/sbgCloud/code/fork-COBRA.tutorials/analysis/conservedMoieties/results/DAS'
ctfRxnfileDir =
'/home/rfleming/work/sbgCloud/code/fork-COBRA.tutorials/analysis/conservedMoieties/data/mini-ctf/rxns/atom
```

```
if ~recompute
    load([resultsDir modelName '_dATM.mat'])
    return
end
```

Load the model and input data

```
switch modelName
    case 'DAS'
        load([dataDir filesep 'models' filesep modelName '.mat'])
        %load('DAS.mat')
    case 'iDopaNeuro1'
        load([dataDir filesep 'models' filesep modelName '.mat'])
        %load('~/work/sbgCloud/programReconstruction/projects/exoMetDN/
results/codeResults/iDN1/iDopaNeuro1/iDopaNeuro1.mat');
        model = iDopaNeuro1;
    case
{'centralMetabolism','centralMetabolism_fastCore','centralMetabolism_thermoKe
rnel'}
        load([resultsDir filesep modelName '.mat'])
        model.SConsistentRxnBool(matches(model.rxns,'DM_fad'))=1;
        %ctfRxnfileDir ='~/work/sbgCloud/code/fork-ctf/rxns/atomMapped'
    otherwise
        load([dataDir 'models' filesep modelName '.mat'])
end
```

Remove any flux inconsistent reactions

Identify the flux consistent set

```
paramConsistency.epsilon = 1e-5;
paramConsistency.method = 'fastcc';
[~,~,~,~, model] = findFluxConsistentSubset(model, paramConsistency);
```

Optionally remove any flux inconsistent reactions and the corresponding metabolites and coupling constraints if necessary

```
if 0
    model = removeRxns(model,
model.rxns(~fluxConsistentRxnBool),'metRemoveMethod','exclusive','ctrsRemoveM
ethod','infeasible');
end
```

Check to see if all internal reactions are already atom mapped within the Chemical Table File repository (https://github.com/opencobra/ctf). Clone this repository locally to ctfRxnfileDir then specify the path below.

```
[metRXNBool,rxnRXNBool,internalRxnBool]  = findRXNFiles(model,ctfRxnfileDir);
```

```
RXN files available for 4 reactions.
RXN files found for all 4 internal reactions.
```

If some internal reactions are missing atom mappings, then atomically resolve the metabolic model including atom mapping of as many internal reactions as possible, without mapping hydrogens, and then calculate the atom transition multigraph.

```
if any(~rxnRXNBool(internalRxnBool))
    param.debug=1;
    [model, arm, report] = atomicallyResolveModel(model,param);
end
```

Atom transition multigraph

If there are atom mappings for all reactions already, just calculate the atom transition multigraph in order to follow the path of all the atoms in the network (this may take some time).

```
if exist('arm','var')
    dATM = arm.dATM;
else
    cd(ctfRxnfileDir)
    [dATM, metAtomMappedBool, rxnAtomMappedBool, M2Ai, Ti2R] =
buildAtomTransitionMultigraph(model, ctfRxnfileDir);
end
```

```
cd(resultsDir)
save([resultsDir filesep modelName '_dATM.mat'],'dATM')
```

Acknowledgments

Co-funded by the European Union's Horizon Europe Framework Programme (101080997)

REFERENCES

1. Ghaderi, S., Haraldsdóttir, H. S., Ahookhosh, M., Arreckx, S., & Fleming, R. M. T. (2020). Structural conserved moiety splitting of a stoichiometric matrix. Journal of Theoretical Biology, 499, 110276. https://doi.org/10.1016/j.jtbi.2020.110276
2. Rahou, H., Haraldsdóttir, H. S., Martinelli, F., Thiele, I., & Fleming, R. M. T. (2026). Characterisation of conserved and reacting moieties in chemical reaction networks. Journal of Theoretical Biology, 112348. https://doi.org/10.1016/j.jtbi.2025.112348