1 Computation of conserved moieties

This folder contains functions for computing conserved moieties in metabolic networks by analysis of the corresponding atom transition networks with graph algorithms. The two main functions are:

1. buildAtomTransitionNetwork: Builds an atom transition network corresponding to a metabolic network from reaction stoichiometry and atom mappings (see Section 2).

2. computeConservedMoieties: Computes conserved moieties by analysis of the atom transition network with graph algorithms. Moieties are output both as nonnegative integer vectors in the left null space of the stoichiometric matrix for the metabolic network, and as sets of atoms (nodes in the atom transition network).

The moiety vectors computed with "computeConservedMoieties" can sometimes be decomposed into two independent moiety vectors. The function "decomposeMoieties" decomposes the moiety vectors, if possible, by solving a mixed-integer linear program with Gurobi. See "**Example/DAS_example_driver.m**" for an example of how to use these functions.

2 Necessary inputs

Building an atom transition network requires reaction stoichiometry and atom mappings for all reactions in a metabolic network. The reaction stoichiometry is obtained from the stoichiometric matrix in the metabolic network reconstruction. The atom mappings can be generated manually, or predicted with algorithms such as DREAM (http://selene.princeton.edu/dream/). Reactions are input to DREAM in rxnfile format (Accelrys, San Diego, CA). Rxnfiles can be written with the function "writeRxnfile". Required inputs to this function are reaction stoichiometry and metabolite structures in molfile format (Accelrys, San Diego, CA). See "Example/Data/write_DAS_Rxnfiles.m" for an example of how to write rxnfiles. Note that metabolite charges in molfiles need to match charges in the metabolic network reconstruction. Otherwise, computed moiety vectors may not be in the left null space of the stoichiometric matrix.

Rxnfiles include a field for atom mappings between reactants and products. The function "writeRxnfile" writes rxnfiles without atom mappings. The function "buildAtomTransitionNetwork" requires rxnfiles with atom mappings. DREAM outputs rxnfiles with predicted atom mappings that can then be input directly to "buildAtomTransitionNetwork". Atom mappings can also be entered in the rxnfiles manually or with other atom mapping algorithms.