**modelUpdatePrep.py**

This algorithm was developed to add new annotations to an existing model. The core of the algorithm are two recursive functions calling one another. One establishes all possible chains of reactions that might connect two points in the existing network. The other selects for valid chains to be added. Valid chains are defined as chains of reactions in which all the metabolites can be connected to the existing network (either they are already in the network can be added via valid reactions in the chain). The algorithm formats a COBRA compatible m file that can be run in MATLAB and automatically add all the new reaction to a cobra model structure. The code also creates lists of added reactions, genes, gprs , potential loops encountered during chain selection and the list of metabolites that broke chain validity and are still unconnected or not included in the network.

**Inputs used:**

1. *Prochlorococcus marinus* MED4 template model
2. Pmm.txt – The list of all KEGG pmm (*prochlorococcus marinus* MED4) annotations – created 04/12/19 by Luca.
3. Kbase reaction file – 02/19 used to establish reaction direction.
4. Keg\_reaction\_bank.txt – contains the new reaction that can be potentially added to the model. This will act as the basis for the bank.
5. KEGG API – for pathway mapping.

**Major dictionaries and variables used:**

1. Neighbors – a set of metabolites neighboring the metabolites added to the network.
2. reacBank – a reaction bank dictionary including reaction direction, substrates, products and pathways.
3. Missing – a set of metabolite that were not connected to the network.
4. Chains- a dictionary storing all possible chains of reactions that can be added through a certain metabolite.
5. Pat0- the metabolite of origin from which the chains are built.
6. Cand – a dictionary containing the connected valid chains selected to be added to the network.
7. Connected- a dictionary collecting all connected metabolites.
8. Dead – a dictionary of blocked metabolites. If a metabolite is added to this collection it means the algorithm could not connect it to anything after it exhausted the reaction bank.
9. Bad- a dictionary containing all the metabolites breaking the validity of chains. Meaning they were part of a chains but are not connected to the overall network and thus render the reaction they’re in as invalid.
10. Mmets- model metabolites dictionary.
11. Mreac – model reaction dictionary.
12. Genes- model genes dictionary.
13. Reac2gene – reaction to gene mapping – used to create new gprs to add to the model.
14. Added – a list of reactions to add to the existing model.

**Functions**:

**addNetEntry(reaction, net) :** This function adds entries to the network and collects the added neighbors.

Inputs:

Reaction – a reaction (KEGG)

Net – existing network.

Outputs:

Updates the network file.

Updates the neighbors list.

**isValid(reaction)**: this function checks if all the metabolites in a reaction are connected to the network.

Inputs:

Reaction – reaction name.

Outputs:

Returns True if the reaction is valid.

**updateDict(query, dict, value)**: this function updates a dictionary entry that is a list.

Inputs:

Query – dictionary key

Dict- the dictionary to be updated.

Value – the value to be updated in the list.

Output:

An updated dictionary entry.

**printChain(metabolite, reaction, chains, level, marker)**: this function traverses up the chains of reactions and collects candidate chains. A candidate chain is a chain of valid reactions, meaning reactions in which all metabolites are connected to the network. This function is recursive and only goes up from the metabolite to the start of the chain. If at some point a downwards movement is detected it warns for a possible loop.

Inputs:

Metabolite – metabolite name

Reaction – Reaction name

Level- recursive depth of the chain

Marker – denotes the end point of the chain. If reached – the level is set to 0.

Outputs:

All valid connected chains are added to the cand dictionary.

Suspected loops are written to the loop file.

The network file is updated.

**connectReac(reaction, reactionBank, step)**: This function creates a chain of reactions to connect to the network. When it finishes building the chains by either finding no more available reactions in the bank or it reached a valid reaction (see isValid function), it calls the printChain function to select for valid chains to add to the network.

Inputs:

Reaction – reaction name (KEGG).

reactionBank – Reaction bank, here all the known annotations from KEGG that are not already included in the model.

step – recursion depth.

Outputs:

Together with printChain it creates a dictionary of chains of reactions and metabolites to add to the network.

**General Outputs:**

Outputnet.txt – network file for Cytoscape visualization – bipartite.

modelRefresh.m – a MATLAB file formatted to add reactions to a cobra model. Complies with the COBRA MATLAB toolbox format.

loopLog.txt – a list of potential loops detected during chain selection.

Chains.txt – the list of chains added to the network.

Pathways.txt – pathway mapping of the added reactions (Needs improvement as I found out the initial mapping is lacking).

Added\_reactions.txt – a list of the added reactions.

New\_added\_genes.txt – a list of added genes.

Gprs.txt – a list of correctly formatted gprs to add to the model.

Breaking\_bad\_mets.txt – a list of metabolites that broke the chains they were in.