MATLAB-READY VERSIONS ADULT HERMAPHRODITE AND MALE CONNECTOMES (constructed from "SI 5 Connectome adjacency matrices, corrected July 2020.xlsx")

Connectome	MATLAB *.MAT File	Vertices	Edges	Isolated Vertices	Self-Edges
Adult Herm Chem	GHermChem.mat	454	4879	8	38
Adult Herm Gap Jn Sym	GHermElec_Sym.mat	469	1450	7	17
Adult Male Chem	GMaleChem.mat	575	5306	16	60
Adult Male Gap Jn Sym	GMaleElec_Sym.mat	586	1758	43	34

NOTES:

I. Each of the four *.mat files contains the data corresponding to one connectome. To be specific, each *.mat has the following information:
   A. Digraph (in case of a chemical connectome, e.g., GHermChem) or undirected graph (in case of a gap junction symmetric connectome, e.g., GHermElec_Sym).
   B. Adjacency matrix (e.g., AHermChem or AHermChem_Sym). Its (i, j)-th element is the edge weight from node i to node j.
   C. Neuron names in each group. e.g., InterNeurons contains all the neurons that belong to the "InterNeurons" group.
   D. Neuron names in each subgroup, e.g., InterNeurons_0 contains all the neurons that belong to the "InterNeurons_0" subgroup of the "InterNeurons" group.

II. Digraphs/Graphs are represented in MATLAB's graph data structure which has a node table and an edge table.
   A. Each row of the node table represents one neuron and each neuron has 3 variables:
      1. Name: I2L, I1L, etc.
      2. Group: Pharynx, InterNeurons, MotorNeurons, etc.
      3. Subgroup: InterNeurons_2, MotorNeurons_Sublateral, etc.
   B. Each row of the edge table represents one edge and each edge has 2 variables:
      1. End node pairs as (StartNode, EndNode): (I2L, I1L), (I2L, I4), etc.
      2. Weight: 2, 1, etc.

III. While the adjacency matrix, neuron groups and subgroups are included in the *.mat files, the digraph/graph data structures already contain these meta-data. For example, consider GHermChem.
   A. To get the adjacency matrix, use
      >> AHermChem = adjacency(GHermChem, 'weighted');
   B. To extract the node information, use
      >> GHermChem.Nodes
      and you get the complete node table (with node names and their corresponding groups and subgroups).

C. To extract the edge information, use
>> GHermChem.Edges
and you get the complete edge table (with edge end node pairs and their corresponding edge weights).

IV. Each digraph/graph may have isolated nodes and/or self-edges (self-loops).

A. To remove isolated nodes, use
>> GHermChem_sans_IsolatedNodes = rmnode(GHermChem, find(indegree(G) == 0 & outdegree(GHermChem) == 0)); % for a digraph;
>> GHermChem_Sym_sans_IsolatedNodes = rmnode(GHermChem_Sym, find(degree(G) == 0)); % for an undirected graph;

B. To remove self-edges, use
>> GHermChem_sans_SelfEdges = rmedge(GHermChem, (1:numnodes(GHermChem)), (1:numnodes(GHermChem)));
Note that removal of self-edges may create additional isolated nodes (this occurs when the original digraph has nodes having only self-edges).

The group/subgroup names were transferred from the *.xlsx files. If there are errors, please inform me: **kamal@miami.edu**

Kamal Premaratne
Victor P Clarke Professor
Department of Electrical and Computer Engineering
University of Miami
Coral Gables, Florida 33146
USA