NetworkBuilder

%%Inputs

Number of neurons (excitatory and inhibitory)

Number of neurons/nodes to be removed after initial development

Radius of the sphere to spread nodes on it

Number of loops for the simulation to go through to build connections between neurons

Mean value for sigmoid function to determine capability of a neuron to accept or make connections (decreases as the neuron reaches the set value)

Where to start removing nodes

%%Find xy-coordinates for each node on the surface of the sphere

Call function: ‘ParticleSampleSphere’.

This function is developed by Anton Semechko (2022). Suite of functions to perform uniform sampling of a sphere (https://github.com/AntonSemechko/S2-Sampling-Toolbox), GitHub. Retrieved April 19, 2022.

%%Determine the coordinates of inhibitory neurons with a uniform distribution

Use the same ‘ParticleSampleSphere’ function but with just the number of inhibitory neurons

%Find the closest corresponding node on the original sphere

for all the neurons

for inhibitory neurons

difference between xyz-coordinate of original sphere and the new sphere

euclidean distance of the difference between node position on the two spheres

end

end

find minimum distance

%find inhibitory coordinates

for inhibitory neurons

define index of the inhibitory node by finding the node that has the closest distance on the second sphere

end

%%Define set values for connections

Define the number of receiving and sending connections for both excitatory and inhibitory neurons

%%Find arc distances between any two nodes

Call function: ‘ArcDistCalc’

ArcDistCalc

%Find arc distances between any two nodes

for number of neurons

for number of neurons

calculate arc distance of nodes

end

end

%Create sigmoid distributions for a distance-dependent relationship between neurons and their distances

depending on the scenario that is picked the mean value for sigmoid function is set differently

zero self connection probability

%%Develop the network connectivity

Call function: ‘ConnDev’

ConnDev

Create sigmoid distribution to decrease the ability of a neuron to form or receive new connections as its total synaptic strength increases

set the number of inhibitory connections that inhibitory neurons or excitatory neurons receive

set the number of excitatory connections that inhibitory neurons or excitatory neurons receive

set eta as the number of neurons to make connections at each round(loop)

For number of loops

find total post-/pre-synaptic strength of which neurons is less than the set values

if no neuron is lacking connection

break

end

else

if number of neurons lacking connection is less than eta

these specific neurons are to be picked for connectivity creation

else

pick eta neurons randomly but weighted based on their capacity, based on the sigmoid distribution

randomly pick a partner neuron for each of the selected neurons, based on both distance-dependent and synaptic-weight probability distributions

check whether picked partners are excitatory or inhibitory

if pre-synaptic partner is missing excitatory(inhibitory) and the selected post- synaptic partner is missing excitatory(inhibitory) input, then

create connection between them

end

end

end

end

%%Remove nodes and related edges

Call function: ‘rmvnodeedge’

rmvnodedge

%Adjust the dimension of the matrices of network parameters

Adjust xyz-coordinate matrices

Adjust distance-dependent connection probability matrix

Adjust the arc distance matrix

remainder neurons = number of neurons - removed nodes

find the node with highest z coordinate

%find the en passant arc connections through lesion (removed nodes)

for all the remainder neurons

find all the connections that remainder neurons made

for all the partner neurons

create nodes at the 1/4, 1/2, and 3/4 of the arc connection between partner nodes

if z-coordinate of the 1/4 or 1/2 or 3/4 node > highest z coordinate

remove the connection

end

end

end

find all the nodes that miss connection(s)

%%Develop compensatory connections after cut/node removal

Call function: ‘ConnDevAftercut’

ConnDevAftercut

%All the steps are similar to ConnDev function, except:

1.Sigmoid function for distance-dependent probability is different based on the scenario that has been picked by user in the ‘NetworkBuilder’ function

2.After selection of synaptic partners,

if a middle-node on the arc between these two nodes < highest node at the edge of lesion

make connection

end

%%Save the .mat files to hold both the geometry and the connectivity map of intact and lesioned networks