**Random network generation algorithm.**

An algorithm was developed in Matlab (MathWorks Inc., Natick, MA, 2018) for the purpose of simulating and analysing randomly generated biological networks consisting of reversible enzymatic reactions. The algorithm workflow is as follows:

1. User defines the number of species in the network as well as their uniform initial abundance (au) and scale of rate constant values.  
   * Note that noise will be added to this value.
2. The algorithm creates a numerical vector of size corresponding to the user-defined number of species so that each species has a unique numerical identifier.
3. The species are duplicated into species and species where corresponds to the inactive form of the species with identifier and corresponds to the active form of the species with identifier .
4. A random species is chosen. The algorithm can then chose a random species or a random species with equal probability.  
   * If an species is chosen an activatory reaction is formalised as:

And assigned a rate constant where .

* + If an species is chosen an inhibitory reaction is formalised as:

And assigned a constant where .

* + Note that the selection of a random or value will follow a uniform probability in the case of random topology network or will be biased in favour of species with more existing interactions in the case of a scale-free network topology.

1. Repeat step (iv) for a user-defined number of reactions.   
   * User defines the number of reactions as a proportion of the total number of all possible interactions where is the user-defined number of species in the network. Such interactions encompass all pairwise interactions between species as well as interactions between species and species whilst excluding all self-interacting reactions for simplicity.
2. The generated rate constants, initial abundances as well as the strings corresponding to reactions, rate laws and species names are inputted into ‘Simbiology’ to create an ordinary differential equation (ODE) – based model where all reactions follow mass action kinetics. An extra reaction in the form of is introduced to affect a randomly chosen species to model a network stimulus.  
   * This stimulus is introduced as an ‘Event’ after time is allowed for model equilibration. Default stimulus strength = 10·InitialSpeciesAbundance.
3. The model is simulated using Matlab’s ode15s solver. If the model displays a minimum number (default = 10) of species which display a recognisable change from steady state in response to the network stimulus (in the form of peaks/troughs) the absolute- or fold- change values for the responding species are saved into an excel file with a unique model identifier.   
   * Note that only species are analysed for peaks/troughs.
   * The Simbiology model file is also saved with the same unique identifier.
4. Steps (iv) to (vii) are repeated for a user-defined number of randomly generated models.
5. A separate script must be run in the same directory which iteratively opens each saved model and its corresponding excel file.
6. All the parameters in each imported model are sequentially perturbed (default = x10) and the absolute- or fold- change from the corresponding reference peaks/troughs saved in the excel file are calculated and averaged.   
   * An increase in peak/trough responsiveness to the stimulus will show as positive numbers.
   * A decrease in peak/trough responsiveness to the stimulus will show as negative numbers.
   * Results display the average change in the absolute- or fold-change value of responding species in the network for each parameter perturbation (x-axis) for each network structure probed (y-axis). Results can be viewed as a heatmap and saved as an excel file.

**Things to note:**

-In all cases run the 'RandomNetworkGeneration.m' file first (with user-defined inputs) and once the script has finished running and all the models have been saved to the current directory then the 'RandNetGen\_analysis.m' script can be executed. Note that when running the second script the user must define the Time at which the stimulus Event takes place (around line 7). This is assigned automatically as the ‘EventTime’ variable in the former script as the halfway timepoint of the defined timecourse duration (around line 135 in 'RandomNetworkGeneration.m' file).

-Timecourse duration is defined by the ‘Timecourse\_Duration’ variable through a rule of thumb that depends on the rate constant scale (around line 117 in 'RandomNetworkGeneration.m' file).

-The minimum number of species in the network that must display a minimum responsiveness of 1a.u of peak/trough prominence is defined by the ‘StimulatedSpeciesThreshold’ variable (around line 134 in 'RandomNetworkGeneration.m' file). Note that only ‘F’ species are analysed. The minimum peak prominence is defined in the ‘findpeaks’ matlab function as 1a.u (around lines 146 1nd 150 in 'RandomNetworkGeneration.m' file).

-In the ‘RandNetGen\_Analysis.m’ file the type of parameter perturbation is defined around line 33) and files are opened in pairs (around line 17). The file opening might be the source of a bug if in different PCs .sbproj files are listed after .xls by Matlab. If this happens, change their reading order through lines 17 and 18.