# Exponential fit

Script for calculating the slope of an exponential fit of the components’ sizes, and for calculating the p-value of the hypothesis that two slopes are the same.

* CalculatePValues.py: this script assumes that two files are in the same directory, “netSizes1.npy” and “netSizes2.npy”, each one including a numpy array with the size of components. The resulting slopes are stored in the variable “Exponents”, the width of the confidence interval in “Intervals”. Finally, “PValues” encodes the p-values of the pairwise comparison.
* ExpFit.py: script with auxiliary functions. Specifically, “GetExpFit” returns the exponential fit of the distribution, and “CompareTwoExpFits” the p-value of a pairwise comparison.

# Random network creation

Script for creating random graphs with the same number of nodes and links than the original networks. The algorithm is based on the one published in:

Wandelt, S., Sun, X., Menasalvas, E., Rodríguez-González, A., & Zanin, M. (2019). On the use of random graphs as null model of large connected networks. *Chaos, Solitons & Fractals*, *119*, 318-325.

* ProgramRandom.py: main script. Input networks have to be located in the directory /Networks/, while random networks are saved into /RndNetworks/. Each input network must be a square numpy matrix, representing the adjacency matrix of the graph. The output is a three-dimensional numpy array, where the first dimension defines the random graph to be accessed.
* LargeRandomNetworks.py: script for generating the random graphs; please refer to the above paper for further details on the algorithm.

# Topological analysis

Scripts for calculating the topological properties of networks, normalised by what expected in random graphs with the same number of nodes and links.

* ProgramFeatures.py: script for calculating the topological features of the network, except for motifs. The original networks should be in the /Networks/ folder, and the corresponding random networks in /RndNetworks/. Note that the name of a network and its randomised version must coincide. The properties of all networks are saved in the allProp.npy file, as a bidimensional numpy array in which the first dimension points to the network, and the second to the feature. Features are: number of nodes, link density, max. degree, efficiency, modularity, assortativity, transitivity, Information Content, Diameter, Average shortest path length and Small-worldness.
* ProgramMotifs.py: script for calculating the frequency of motifs. Input and output files are the same as in the previous script.

The following auxiliary scripts are further used:

* NetworkProperties.py: calculation of several basic topological metrics.
* Communities.py: script for calculating the modularity of a network. Copyright: ﻿Thomas Aynaud, [thomas.aynaud@lip6.fr](mailto:thomas.aynaud@lip6.fr).
* NetInfCont.py: calculation of the Information Content.
* Motifs.py: fast algorithm for detecting 3-nodes motifs.
* Motifs4B.py: calculation of 4-nodes symmetric motifs.

# Requirements

* Python, version 3.\*
* Standard libraries, including numpy, scipy, network, glob.