Network Robustness

Exploring robustness in networks by affecting their clustering coefficient

Author: Julio Correa

1 Introduction

The analysis of real-world network properties by adjusting different models of random networks is widely studied. Albert-Laszlo Barabasi, in his book 'Network Sciences,' chapter 2 [1], makes a remarkable introduction to random graphs using as an example a party where guests connect randomly. On the other hand, there are some relevant questions that network models can help answer. Some of these are: 1. What values of any given measure are possible? 2. Is it an observed value large or small ?; 3. Is there a relationship between the values ?; 4. Can the fitted random network model predict or explain some dynamics in the first network?

There is considerable literature about how a network following a power-law degree distribution is robust against random failure, but not against targeted attacks. One of the most relevant characteristics of networks is that their clustering coefficient is small. In fact, according to Holme et. al. [2], some algorithms can be proposed to introduce a tunable clustering coefficient, which produces a higher clustering than the one observed in a purely preferential attachment generated network.

In this report, I will explore how affecting the topology of a given network leads to interesting results in robustness. The mechanism chosen to do so, is an edge swapping operation and a sampling process, for some given real-world networks. The sampling process seeks to decrease the clustering coefficient of the network and emulate the low clustering property of a scale-free one. On the other hand, the deletion process aims to simulate a random attack process, by removing edges and nodes according to certain probability q. Finally, we compare the behavior of each network against this perturbation process.

2 Degree Distribution characteristics and how to approach them

2.1 Introduction

Networks whose degree distribution follows a power-law appear to be more robust against random attacks. There is academic discussion about the real existence of networks following this type of degree distribution, and the model introduced by Barabasi and Albert, at the dawn of the new millennium has been said and questioned enough. However, the model has been useful to model a range of real-world networks adequately.

One of the characteristics of the model is that when generated using the algorithm, the resulting graph has a low Clustering Coefficient, something that often does not occur with real-world networks. A development of this idea and a proposal to improve the model was introduced by Holme et. al. [2].

2.2 Motivation and Problem Statement

As I mentioned in the introduction, the networks generated by the preferential attachment algorithm are robust against random failures, and one of their most relevant characteristics is that they have low value for its clustering coefficient. In this sense, this project aims to take some real-world networks and sample them using an edge swapping operation. Its objective is to make the clustering coefficient to go down, emulating what would be this characteristic of a scale-free network. Then study how they respond to random failures.

The paragraph above leads me to state the following hypothesis: *By introducing a swapping operation to sample actual network datasets (different networks) and reduce their clustering coefficient, I -**somehow**[1] - retrieve the robustness of a power-law against random failures, which is the robustness of a network with low clustering.*

3 Reducing the Clustering Coefficient and testing the robustness of real-world networks: A two step approach

In the present section, I explain how the edge swapping operation works and how it reduces the clustering coefficient of a real-world network. Then, I introduce a random deletion process to compare the different responses to perturbations in the original network and the sampled one. The edge swapping operation follows the principles of a MCMC, because it is actually creating many different correlated samples starting from the original one. As a complement, the size of the sample space is huge, approximately equivalent to $|\Omega| \alpha 2^{15783^2}$, being omega the size of the sample space (very large number.)

3.1 Description of the processes

3.1.1 Edge swapping operation

Transformation: The objective of this transformation is to preserve degree distribution and maintain the number of nodes.

Definition of the swapping operation.

1. Picking two edges randomly. Let's say (i,j) and (l,m)
2. Define an acceptable swap:
   2.1. (i,l) and (j,m), or
   2.2. (j,l) and (i,m)
3. Choose one acceptable swap by chance and perform it.

Figure 1 shows the shape of the two acceptable swaps defined for this purpose. It is important to notice that by maintaining the number of nodes and the number of edges, I am restricting the sample space to

[1] I use the word somehow because I expect the randomised network to respond better against random failures. Better means that the fraction of nodes belonging to the largest connected component after deletion is higher in the randomised network.

all the possible acceptable swaps that retrieve a sample belonging to the previous definition of the sample space, $|\Omega|$.

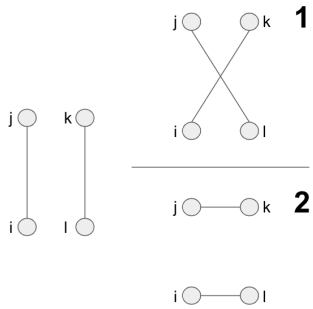


Figure 1. Edge swapping process and acceptable swaps.

3.1.2 Random deletion process

3.1.2.1 Background and description

Wu [3] presents a complete development on scale-free networks and proposes a model to improve their structural response in terms of robustness. The first two sections make a comprehensive analysis of the reaction of this type of networks to random failures. It concludes that they do it better than other types of networks, for example, those generated using a Poisson distribution. This is the primary motivation to study the response of a network to random failures, reducing its clustering coefficient.

What I do here is to simulate a random deletion process (a.k.a. random attack) to nodes and edges in both, the original network and the sample chosen after the edge swapping operation applied to it. The random failure process works as follows:

1. Start from a graph sampled from the original one. Thus, with fixed degree distribution.
2. Delete links (or nodes) with probability q.
3. For each deletion step, store the fraction of nodes from both the actual and the randomised networks, that belongs to the largest connected component after deletion.

Each experiment to remove either nodes or edges was run 1,000 times. The result for the fraction of nodes in the largest connected component was averaged over all of these values.

What I do to implement the function in Python. For development purposes, I show the case with a di-graph of blogs [i], whose number of edges is 15,783 and its number of nodes is 1,224. Also, at the end of the present section, I include some results of the analysis of C-Elegans [ii] (Nodes = 453 , Edges = 2,415) and Star Wars characters [iii] (Nodes = 109 , Edges = 398) networks, just to illustrate how this results apply to different networks.

3.2 Simulations and results

3.2.1 Results for MCMC simulations

Figure 2 presents the results in the variations of three measures for the network: 1. Clustering Coefficient; 2. Average Shortest Path; 3. Diameter.

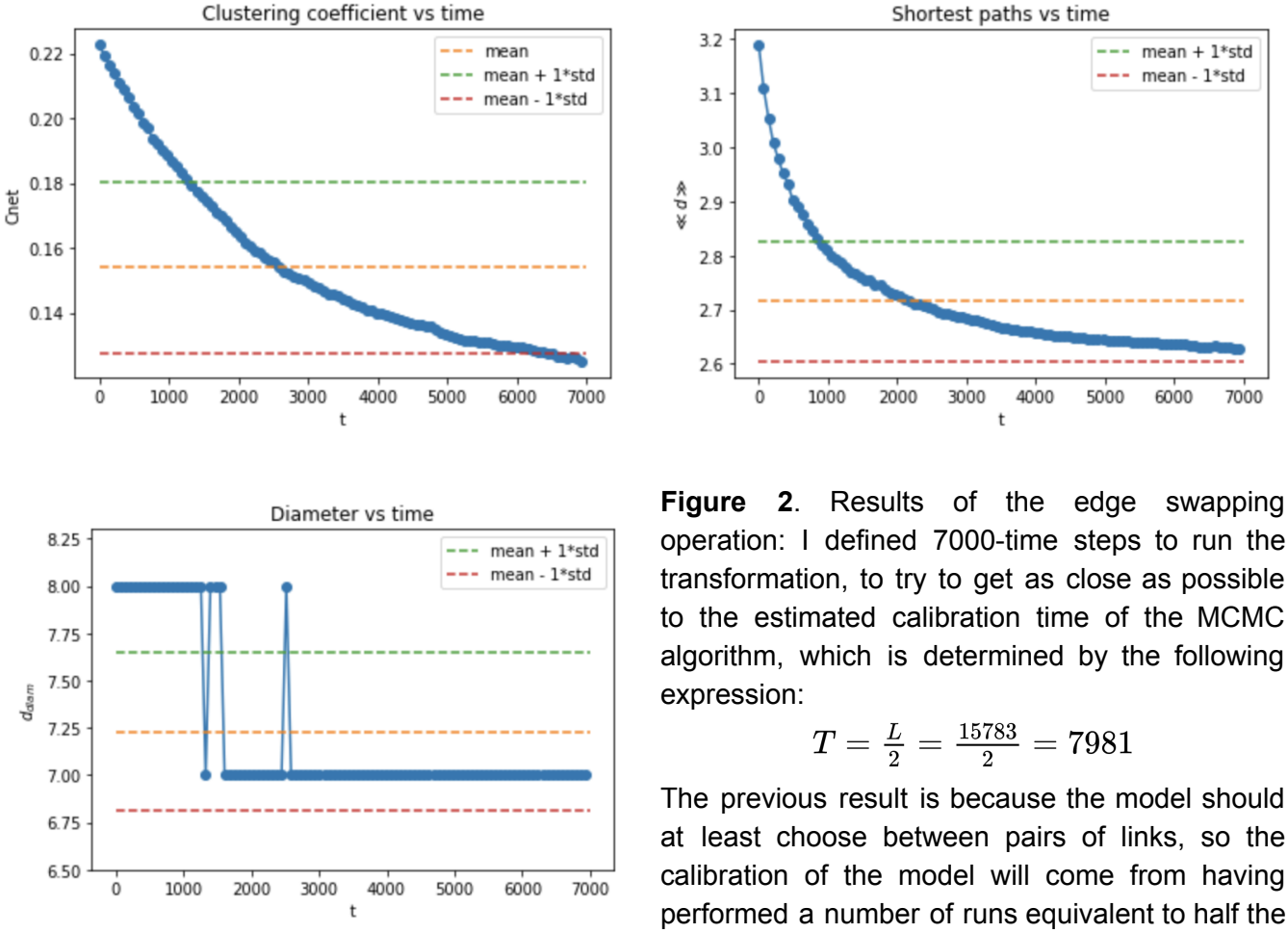


Figure 2. Results of the edge swapping operation: I defined 7000-time steps to run the transformation, to try to get as close as possible to the estimated calibration time of the MCMC algorithm, which is determined by the following expression:

$$T = \frac{L}{2} = \frac{15783}{2} = 7981$$

The previous result is because the model should at least choose between pairs of links, so the calibration of the model will come from having performed a number of runs equivalent to half the total number of links, in this case, 7,981.

The main result is that effectively after the edge swapping process, the clustering coefficient decreases substantially. It is interesting to note that both the diameter of the network and the average shortest path also decrease, showing small-world properties.

3.2.3 Results for Blogs Network random deletion

Figure 3 displays the results for the blog directed graph. It is very clear from the plots that the randomized network behaves better when faced with random failures.

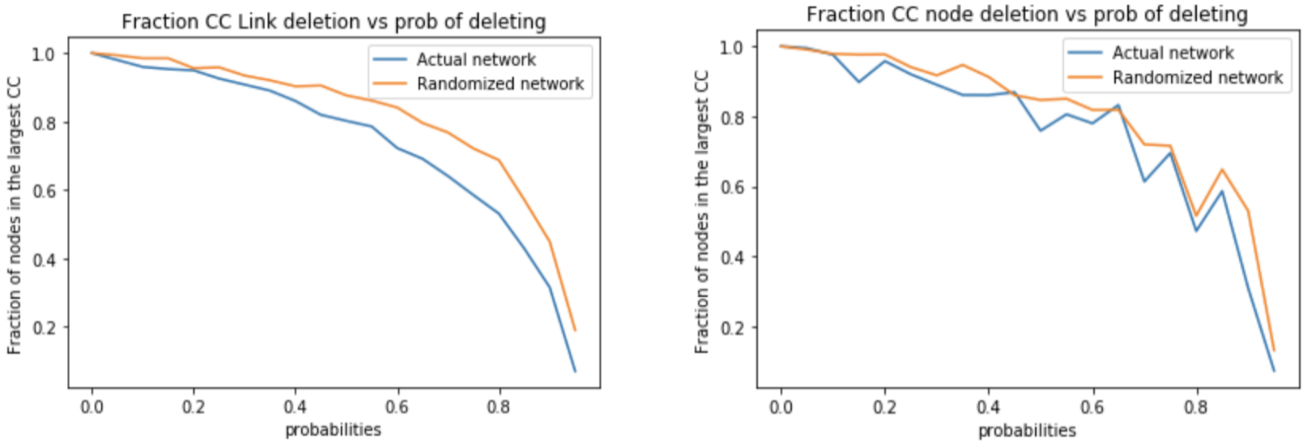


Figure 3. Results of node deletion and link deletion for blogs di-graph

This allows me to conclude that for both random failure processes, the randomized network behaves better when faced with failures. Furthermore, it is even more evident that the best network in this phenomenon, the one that retains the largest fraction of nodes connected to the largest component, is the randomised network; it is better in the sense of consistency. The same applies for removing edges. In addition, appendix 1 includes a brief sensitivity analysis of these results.

3.2.3 Results for C-elegans and Star Wars networks

Then, I used two much smaller networks and undirected ones. One is a network of C-elegans and the other one is Star Wars networks. The results are shown in **figure 4.**

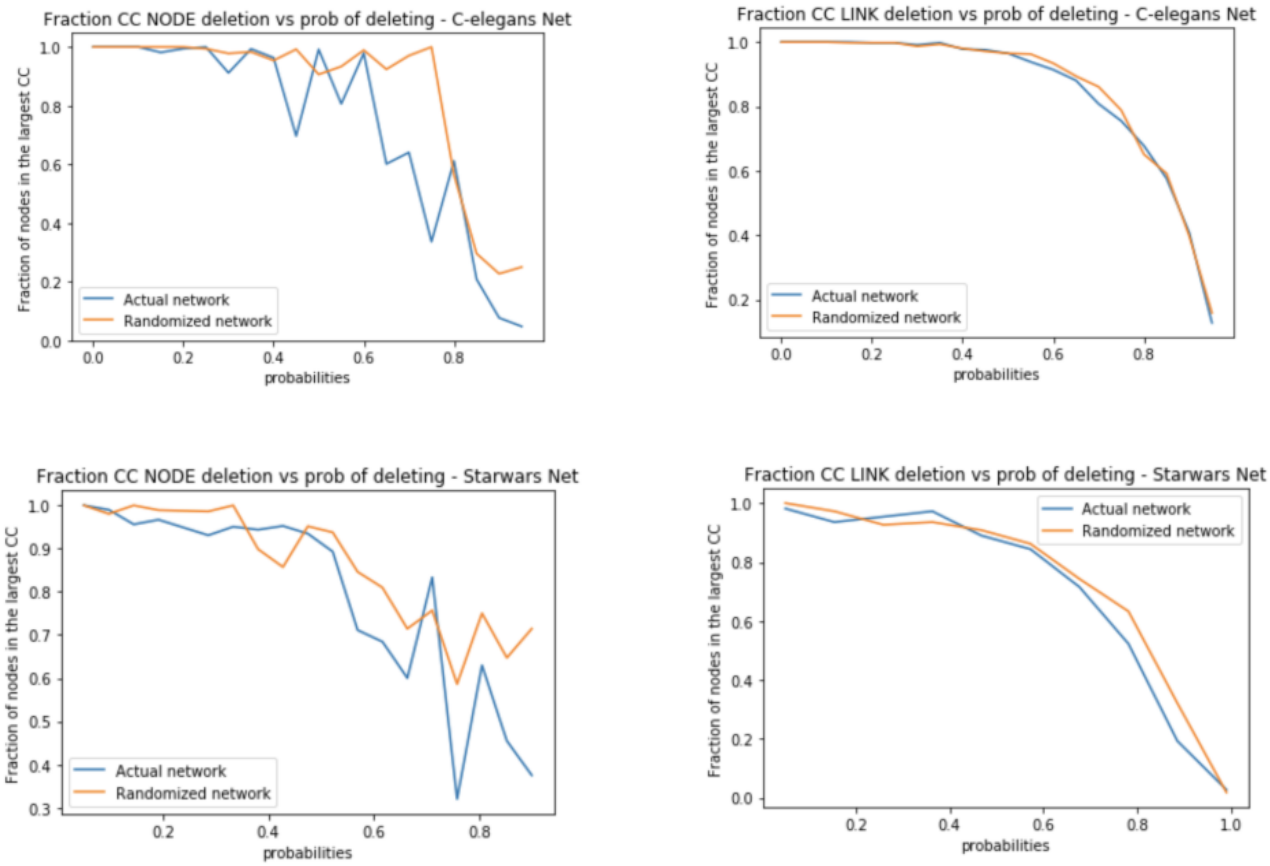


Figure 4. Results for edge and node deletion in C-Elegans and Star Wars networks

As one can see, in the case of the node deletion process, the best response in most cases comes from the randomised network. Except for two or three probabilities, in the other 19 used to define the experimental setting, it can be seen that this network gives a better response to deletion.

In the link deletion process, it is not clear if the randomized network is better than the actual network. However, one thing is clear: It is not worse than it. We see, for example, that in the case of the Starwars network, in most cases, it does show a better response to failures. In the case of the C-Elegans network, both networks behave very similarly.

3.3 Summary

From the different plots, we can conclude that the transformation applied reduces the clustering coefficient of the network. In particular, initially the value was 0.22 and at the end of the process it is 0.135, which means that we could destroy triangles in the network. Therefore, the first part of the proposed methodology gives the expected output.

We conclude that for this particular case, the randomised network behaves better against random failures rather than the original network. One conclusion is that by swapping edges I *'destroy'* triangles and therefore, get closer to a theoretical scale-free network.

4 Conclusion

Two disturbance processes to a network have been proposed: edge swapping operation to diminish the value of the clustering coefficient and also an edge/node deletion process to simulate a random failure process. The aim was to emulate the low clustering of scale-free networks and its robustness against random failures. It is clear that for almost every single case analysed, the randomised network by the swapping operation, behaves better than the original network. There is just one case (out of 6,) where it is hard to conclude whether the randomised network behaves better or not than the actual network. Finally, it is somehow clear that the results are on the right track to confirm the hypothesis. Nevertheless, a more sophisticated statistical method is required to conclude.

Some limitations of the proposed approach are:

1. As long as it is focused on the Clustering Coefficient, I do not explore other dynamics. It would be interesting to test different transformations to explore, for example, how the largest connected component generates meso-structures, such as blocks and modules.
2. The computational time for each of these simulations grows with the square of the number of nodes, because it has to test pairs. So the computational complexity of the presented algorithm is $O(N^2)$. I have to wait a long time to generate the simulations.
3. My approach considers to sample just one network, which is the one at the end of the process. If I had more computational power, I could increase the number of samples and average over them. It is important to notice that my approach still holds and the hypothesis does not change because of this.

Recent development:

1. In [5], Lee et. al., propose different approaches to study the causes of percolation in different real-world systems. Percolation is a statistical physics concept widely applied to different disciplines, in particular to network resilience. They propose a model, called the Pott model, to test universal behaviours in networks under perturbation.
2. Finally, in [6], Gray et. al., propose an approach to create a much interesting way to sample networks. Their scope is to use a Bayesian setting to sample a Random Graph given a fixed value for certain parameters.

Next steps.

After working this report out and the simulations, I think that one interesting future development would be to test more complex sampling methods, such as Metropolis-Hastings. By doing so, it would be possible to test more than one measure at a time and also, it would be possible to extend the result to an information theory framework, where it is possible to compute the log likelihood of each sample to understand how informative it is.

Also as it is stated in [6], it would allow to define criteria of acceptance in the MCMC process, so it could be much interesting to include the idea of acceptable swaps for any given value of the clustering coefficient. So I could restrict the transformation to just a certain fraction of the sample space.

Also, I could explore this Bayesian setting, where I could choose a sample graph given, for instance, a certain set of values of the largest cliques, or for the largest connected components.

Appendix ii shows further applications & research associated with the present report.

References

[1] Barabasi, A.-L. Network Science. Cambridge University Press, 2016.
[2] Holme, P., Kim, B.J. Growing Scale-Free Networks with Tunable Clustering. Department of Theoretical Physics, Umea University, Sweden (2001).
[3] Wu, J., Tan, S., Liu, Z., Tan, Y., Lu, X. Enhancing structural robustness of scale-free networks by information disturbance. Sci Rep 7, 7559 (2017).
[4] Martin Rosvall, Carl T. Bergstrom. An information-theoretic framework for resolving community structure in complex networks, Proceedings of the National Academy of Sciences, May 2007, 104 (18) 7327-7331; DOI: 10.1073/pnas.0611034104.
[5] Deokjae Lee, Y. S. Cho, K.-I. Goh, D.-S. Lee, B. Kahng. Recent advances of percolation theory in complex networks. arXiv:1808.00905v1 [physics.soc-ph], Aug 2018.
[6] Caitlin Gray, Lewis Mitchell, Matthew Roughan. Generating Connected Random Graphs. arXiv:1806.11276v2 [cs.SI] 26 Oct 2018.

Networks.

[i] http://konect.uni-koblenz.de/networks/moreno_blogs
[ii] Gabasova, E. (2016). Star Wars social network. DOI: https://doi.org/10.5281/zenodo.1411479.
[iii] Marcus Kaiser and Claus C. Hilgetag. "Nonoptimal component placement, but short processing paths, due to long-distance projections in neural systems." PLoS Comput Biol 2.7 (2006): e95.

Appendix i. Sensitivity analysis of deletion process Blogs network

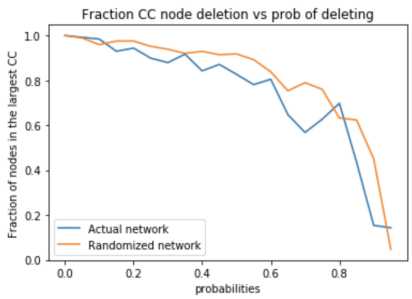
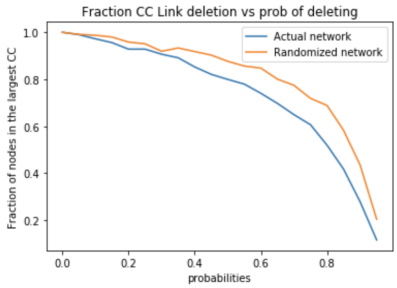
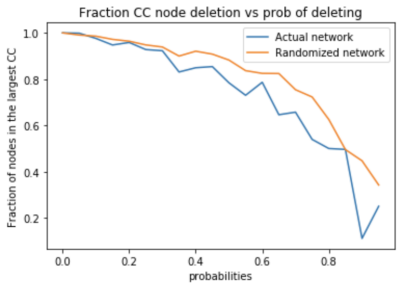
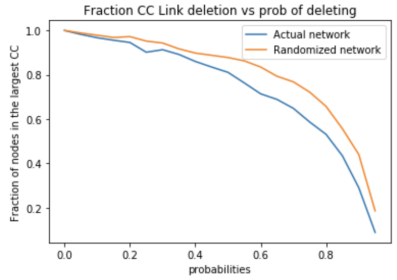
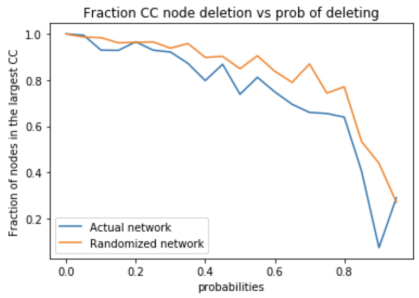
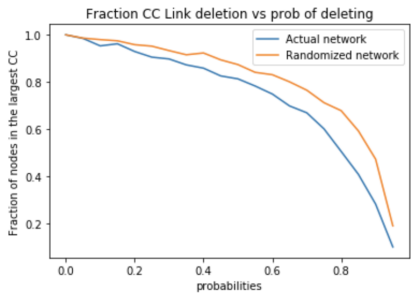
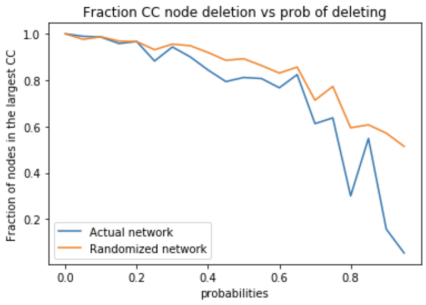
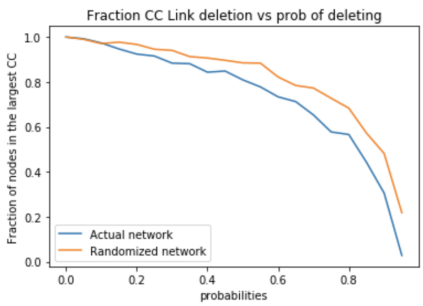
Runs	Node deletion	Link deletion
50		
100		
200		
300		

Figure 4. Sensitivity analysis for node and link deletion

As we can notice in figure 4 (above,) the only valid conclusion to this sensitivity analysis is that the randomized network is better than the actual against random failure.

Appendix ii. Further applications of the swapping / deletion processes

Some applications or further research for this work could be the following:

1. Introduce more complex sampling models, like MCMC Metropolis-Hastings methods to randomize the network. It should allow to examine in more detail when a network different configurations of the sampled one and define criteria for the different acceptable swapps.
2. Explore what happens with the equilibration of the MCMC when the time steps tend to infinity. This means that we could keep sampling our actual network to be absolutely sure that there is no correlation between samples.
3. Explore larger networks, maybe using GPU (Google Colab, AWS, etc.). It is computationally expensive to run these operations in my machine, so that it will be useful to use some computational power available.
4. Dive deeper into the differences between node deletion and link deletion process. Even though the overall result is similar, in the detailed dynamic they may behave differently.
5. As Martin Rosvall, Carl T. Bergstrom states in [4], it would be useful to explore and evaluate these techniques to the ones from information theory in data/networks compression.