Nine Quick Tips for Analyzing Network Data

Vincent Miele, Catherine Matias, Stephane S. Robin, Stéphane Dray

Nine Quick Tips for Analyzing Network Data

Vincent Miele[1,*], Catherine Matias[2], Stéphane Robin[3], and Stéphane Dray[1]

[1]Université de Lyon, F-69000 Lyon; Université Lyon 1; CNRS, UMR5558, Laboratoire de Biométrie et Biologie Évolutive, F-69622 Villeurbanne, France
[2]Sorbonne Université, Université Paris Diderot, Centre National de la Recherche Scientifique, Laboratoire de Probabilités, Statistique et Modélisation, F-75005 Paris, France
[3]UMR MIA-Paris, AgroParisTech, INRA, Université Paris-Saclay, F-75005, Paris, France
[*]Corresponding author: vincent.miele@univ-lyon1.fr

Abstract

These tips provide a quick and concentrated guide for beginners in the analysis of network data.

Introduction

Nowadays, networks are widely used in Biology, Bioinformatics, Ecology, Neuroscience or Epidemiology to represent interaction data [1]. A network contains a set of entities (the *nodes* or *vertices*) that are connected by *edges* (or *links*) depicting some interactions or relationships. These relationships may be either directly observed or deduced from raw data. The first case encompasses protein-protein interaction (PPI) networks where interactions between two proteins are experimentally assessed or plant-pollinator interactions that are directly observed in the field. Gene regulatory networks reconstructed from gene expression data, co-occurrence networks inferred from species abundances or animal social contact networks deduced from GPS tracks are some examples of the second case. New kinds of networks are still emerging (for instance, cell-cell similarity networks [2], Hi-C networks and image similarity networks [3]).

Networks are very attractive objects and many methods have been developed by network experts to analyze their structure. However, biological networks are often analyzed by non-specialists and it may be difficult for them to navigate through the plethora of concepts and available methods. In this paper, we propose the subsequent list of quick tips to avoid common pitfalls and enhance the analysis of network data by biologists.

Tip 1: Formulate questions first, use networks later

Network theory is well established and truly powerful but it cannot be used as a "black-box". Building a network should not be considered as an end in itself. We recommend to (i) establish a list of scientific questions and hypotheses before manipulating the data and then (ii) evaluate if these questions naturally translate into a

series of network analyses – rather than making network analyses first and checking whether they raise questions after. Indeed, it is generally immediate to represent/model the data with a network, but much trickier to translate a question into a network-based analysis.

To this end, it is important to go further the network formalism and embrace the network viewpoint. It relies on a cornerstone idea that makes the strength but also the complexity of network modelling: any potential interaction is considered within its context taking into account the other interactions that occur (or not). In particular, it assumes that any interaction between the pair of vertices (A,B) has to be analyzed respectively to the other pairs (A,C) and (B,C). For instance, the importance of a particular edge between two genes will be differently assessed if the target gene is or is not a *hub* (i.e., regulated by many genes). This viewpoint does not consider interactions as independent objects and is thus the exact opposite of examining the set of interactions one by one.

Finally, it is obviously recommended to check whether your questions and data actually fit the network viewpoint before performing any analysis. Let us consider, for instance, a correlation matrix: it can be viewed as a correlation network (one edge per correlation value, see next Tip) but it would be more natural to use network-free methods that can naturally be applied on a complete correlation matrix (e.g., hierarchical clustering or principal component analysis).

Tip 2: Categorize your network data correctly

To grab the cutting-edge concepts and methods in the networks field, learning the appropriate vocabulary from *graph theory* is a prerequisite. In particular, it is important to categorize your network properly to be sure you apply suitable methods. Different network categories for different data lead to different approaches: don't put a square peg in a round hole!

Links can be *directed* (from a *source* to a *target*), possibly including *self-loops* (e.g., a protein interacting with itself or cannibalism in food webs). Ignoring this information –for the sake of simplicity– would actually betray the original data. When dealing with edges embedding a value (a *weight*), we strongly advise you to avoid transforming the network into a *binary* one using any *ad-hoc* threshold value because (i) it clears a significant part of the available information and (ii) methods handling weighted networks are usually available. If you do it, just do it as an exploratory step (for instance, to facilitate a first visualization step - see Tip 4), otherwise it will certainly bias your analysis (see [4] for an example with ecological networks). Furthermore, the data analyst must be very cautious since, in the literature, weights can be considered as intensity-based (the greater the weight, the stronger the edge is) as well as distance-based (the smaller the weight, the closer the nodes are).

Additional information on the nodes is often available. Nodes can belong to different categories and edges can be allowed only between nodes of the same category (*bi/tri/multi-partite* networks; e.g., nodes as hosts/parasites or as plant/fungus/seed dispersers). Here, the expected number of edges that is used in many statistical approaches (see Tip 5) is clearly different than in the general case. Also, nodes can have spatial positions/coordinates (e.g. nodes as habitat patches or farms in 2D, brain area in 3D) and be more likely to interact when being close to each other. It is highly recommended to select methods (or methods' extensions) that explicitly handle such properties, either to understand if they truly contribute to organize the network, or to look for some remaining structure, once accounted for their effect [5]. If the analysis does not include such *covariates* in some way, it is still feasible to use them *a posteriori* in the interpretation of results [6] – but there is a risk that you simply rediscover this

information, for instance the categorical or geographical structure of the nodes.

Tip 3: Choose with care an analysis tool

A range of swiss-army-knife software is dedicated to network analysis. It is therefore a waste of time trying to use unspecific tools with a do-it-yourself approach: don't reinvent the wheel! [7] These software tools belong to two distinct categories that have pros/cons: graphical user interface (mouse-based navigation) and software packages (command line interface or programming). The first category is mainly dedicated to powerful and interactive visualization (see Tip 4). It includes the two major open source software tools `Gephi` and `Cytoscape`, both supported by an active community. They also offer the computation of some network metrics (see Tip 5). The second category is dominated by the two leading general-purpose network packages `NetworkX` and `igraph`, but there exist plenty of more specific packages (for instance `statnet` or `bipartite` in `R`). Browser-based visualization tools recently emerged as an intermediate category, including a collection of `javascript` libraries (`Sigma.js`, for instance). Finally, there exist a limited set of common network file formats (e.g. adjacency list in the format `source target`) that you should adopt from the very beginning to be able to easily switch between these tools.

Meanwhile, we strongly suggest that you learn programming and scripting your analysis to enhance reproducibility. First, the aforementioned packages are highly versatile and they include the most recent state-of-the-art techniques to perform a complete analysis on any category of networks. Since their use demands much more computing skills, we highly recommend the reading of papers in the "10 simple rules" collection ([7] for instance). Moreover, scripting your analysis will guarantee the reproducibility [8], one of the emerging concepts of this decade in Science. By no way you can bypass it. On another perspective, reproducibility is also convenient: it takes just the snap of a finger to re-run the complete analysis on a modified version of your raw data, on different datasets or with others colleagues interested in the modelling approach.

Tip 4: Be aware that network visualization is useful but possibly misleading

As already said, networks take their origin in their ability to represent interaction data with a mathematical object (a *graph*) that can be represented from different angles. Indeed, networks can be visualized in two dimensions, nodes being spread in the plane and edges drawn with the objective to achieve the most aesthetic design. Meanwhile, this apparently simple task is in fact a very hard combinatorial problem. An active research community proposed a series of heuristics (often called *layout*) aiming at obtaining a nice network view in a reasonable time, despite the growing size of available networks. The aforementioned tools (see Tip 3) embed a wide range of easy-to-use layouts.

However, special care is required to not over-interpret visualization. A layout does not only provide a nice representation of a network, it makes it optimal for a given set of objectives (e.g., maximizing attractions between neighbors) that you often ignore. As a consequence, what you see with your eyes is biased – most often aesthetic, but biased. When visualizing a network, always keep in mind that the position of a node in such a display is not part of the data, but results from an algorithm. To this respect, the distance between two nodes should not be interpreted as an intrinsic measure of proximity as another display algorithm would result in a –possibly very– different

distance (see Figure 1). In summary, visualization must remain a tool to find a track for further network analysis but only in rare cases (mostly when your network is very small) it can provide relevant conclusions.

In a more anecdotal way, we also advise to consider visualizing the *adjacency matrix* as a heatmap (a colored matrix). It allows to represent the edges' presence or weight (colored cells) and their absence (blank matrix cells). This is particularly relevant when a reordering method is applied to sort the matrix rows/columns in an informative manner (e.g., ordering by increasing value of a metric or according to some clustering results; see Tips 5 and 6 and also Figure 2 a).

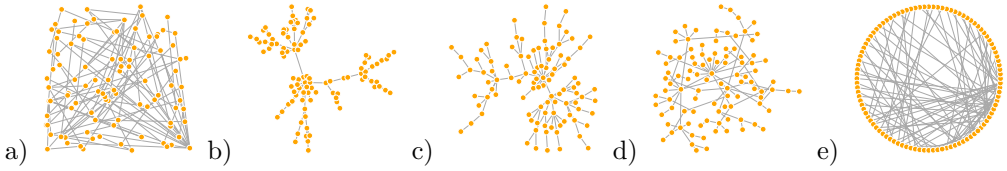

Fig 1. Same synthetic network vizualized with the package `igraph` with five different algorithms: a) random layout, b) Fruchterman and Reingold layout, c) Kamada and Kawai layout, d) Davidson and Harel layout and e) degree-sorted circle layout.

Tip 5: Avoid blind use of metrics, learn formulas instead

Describing a network, beside visualizing it, usually requires to compute appropriate summary statistics and the beginner will immediately find the path to a series of *network metrics*: one number per node or edge (fine-scale metrics; e.g., *degree*) or one number for the whole network (coarse-scale metrics; e.g., *connectance* or *modularity*). Metrics have proliferated and it can be challenging to separate the wheat from the chaff. Consequently, it is strongly advised to take time to read carefully the mathematical definition of the metrics you have on hand (see also Tip 9): the deeper the understanding, the easier the interpretation is. For instance, reading the definition of the widely-used *betweenness centrality*, you can understand it is based on *shortest paths*. If you intend to use this centrality measure, it is therefore necessary to check if the shortest path is a relevant concept associated to a process under interest (such as energy fluxes in food webs) or if it is more questionable (paths in functional brain networks [9] or paths in contact networks when information or disease diffusion is not under study [10]).

Special care is also required when analysing directed and/or weighted networks with extensions of metrics to this case. For instance, the formula of the weighted degree accounts for two effects (how many *neighbors* and how large the weights are) that are impossible to disentangle (a similar objection can be raised for the weighted path [11]): a weighted degree of 2 can correspond to a single edge of weight 2 or four edges of weights 0.5. Lastly, coarse grain metrics are often used to compare networks (networks from different data or conditions, or simulated networks as mentioned in Tip 7). Again, only their mathematical formulation can confirm that these numbers can be compared when, for instance, network size or density is varying (as mentioned in [12] for brain networks).

Alternatively, one could be tempted to compute all the available metrics and try to pull a rabbit out of the hat. This approach has a clear interest – it helps understanding how metrics are correlated – but it is not hypothesis-driven as recommended in Tip 1. We prefer suggesting an incremental approach where you select a metric related to a

particular hypothesis, and continue with another metric unless you have the answer to your fundamental questions on the data.

Tip 6: Choose the clustering method you need, not the one you know

With the data avalanche arising this decade, it is more and more frequent to investigate large networks. Luckily, we can bet that vertices in your network can be aggregated into *clusters* because they tend to be connected in a similar way. In other words, we can simplify the complexity by identifying the network *meso-scale* structure (i.e. zooming out the network).

This idea is often understood as finding the *modular* structure of a network (i.e. dense clusters of nodes poorly connected with others). *Community detection* methods [13] are therefore suitable to these situations where the network is *de facto* modular. Notably, they also have the advantage of being highly computationally efficient. However, it is frequent that your network has a non-modular structure ([14], Figure 2 a) – for instance genes that are hubs connected to other peripheral genes – while the detection methods always give a solution (Figure 2 b). This is the reason why a series of alternative methods propose to infer the unknown structure that is actually present -not necessarily a modular structure (quoting Betzel and colleagues, nodes can be involved in a "diversity of meso-scale architecture" [15]). Methods based on the *stochastic block model* are a valuable option ([14, 16], Figure 2 c), recently used in Biology [15, 17]. They have the advantage of explicitly (and therefore correctly) modelling edge directions and weights by means of different statistical distributions [18]. Another current approach consists in performing a clustering on node characteristics: it includes *motifs*-based approaches [19] and a wide range of innovative *node's embedding* techniques ([20], Figure 2 c) including deep learning perspectives [21].

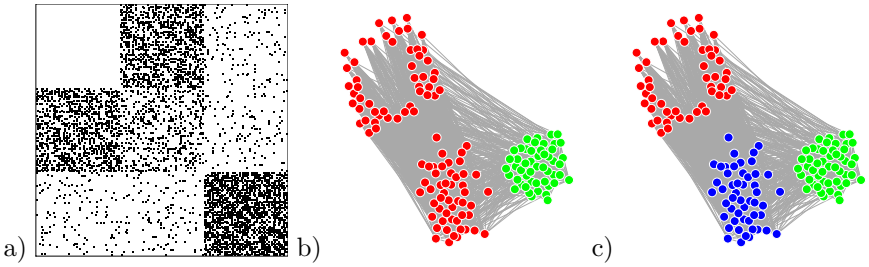

Fig 2. When a network is not fully modular, the community detection can fail to capture the actual network structure. This figure illustrates this on a network simulated with three clusters of 50 nodes linked by a mixture of assortative (*i.e.* module-like) and dis-assortative patterns. a) Adjacency matrix represented with black cells showing edges' presence and with a column/row ordering consistent with the three clusters. Nodes of cluster 1 (first 50 rows/columns) are not connected between themselves but connected with those of cluster 2 (next 50 lines/columns). Nodes of cluster 2 are slightly connected between themselves, whereas cluster 3 (50 last rows/columns) forms a module. b-c) Network visualized with cluster 1 on top-left, cluster 2 on bottom and cluster 3 on the left. Nodes' color is given by the clusters retrieved by b) the community detection method `Louvain` [22] c) the stochastic block model parameters estimated by the `CRAN R` package `blockmodels`, as well as the node's embedding method `node2vec` [23] (followed by k-means clustering).

Tip 7: Don't choose the easy way when simulating networks

Simulating networks consists in drawing "realistic" networks, which can then be compared with the observed one. Simulated networks are supposed to display typical properties arising from a *null* model. The rationale is then to focus on the differences between the observed network and the simulated ones to highlight interesting, *significant* features of the observed one.

Many random graph models (e.g., *Erdös-Renyi, small-world, scale-free, SBM, Exponential Random Graph, configuration model*) exist and can be used as null models, but we recommend not to use them as "black-boxes". Indeed, in many situations, the analyst is aware of a series of properties that should be displayed by the network: imbalanced degree distribution, different nodes' roles associated with available side information, forbidden interactions, etc. Such expected properties must be encoded in the simulation process, otherwise they will emerge and be detected as significant effects, bringing no new insight for the problem at hand. The simulation process must therefore preserve all the data properties except those to be addressed. For instance, when assessing whether the number of feed-forward loops is unexpected in a given transcription network, the simulation procedure must rely on fixed number of nodes and degrees whereas the number of these loops remains free.

Lastly, when the network under study is not directly observed but built from raw data interpretation, it is often relevant to simulate the whole construction process. Consider the case of contact networks inferred from trajectories/movement data: one can either simulate trajectories keeping some properties of the original data and then build a contact network, or directly simulate a "realistic" contact network. The former approach will intrinsically account for the uncertainties and biases induced by the construction steps, which are likely to be overlooked by the later approach. Hence, assuming a significant effect is revealed by the later approach, it can be related to a truly atypical property of the network under study, but it can also result from a bias in the construction process.

Tip 8: Reconsider the data to build multiple network layers

A network is classically the result of some data aggregation. For instance, interactions can be recorded during a large period of time, by snapshots (rounds of data collection) or continuously (continuous data acquisition) and *aggregated* into a single network. On the other hand, interactions can be observed at different spatial locations but ultimately gathered in a single dataset. Finally, different kind of interactions may be observed but, for the sake of simplicity, they are considered as equivalent and modelled by a single edge type. It is now time to dis-aggregate your network and obtain multiple networks layers (e.g. networks in time or space, multiple types of edges). Indeed, working with *multilayer* networks is no more off the beaten track and represents a unique opportunity to tackle new questions [24]. You are strongly urged to keep in mind (and on hand) the different layers of data (time, space, type,...) and consider networks composed by multiple layers.

Among *multilayer* networks, we suggest you keep up-to-date at least on the following instances. A network is called *dynamic* when it gathers a time series of networks (i.e. different snapshots, the nodes' list possibly varying in time). There already exist extensions of the concepts developed in Tips 5,6 to this case - in particular dynamic community detection [25]. If the edges are observed continuously (appear and disappear

continuously with time), this flow of edges is called a *temporal* network [26]. When networks are observed at different spatial locations, the concept of *metanetwork* has emerged in particular in Ecology [27]. Finally, when different layers correspond to different interaction types, a *multiplex* network is built: between any two nodes, there possibly exists more than one edge - one per interaction type at most (often represented with different colors; see for instance trophic and non-trophic interactions between species in [17]).

Tip 9: Dive into the network literature, beyond your discipline

Network science has emerged "at the dawn of the 21st century", as mentioned by one of the pioneers Albert-László Barabási (`http://networksciencebook.com`). It now involves a hyper-active community of researchers from different domains such as Physics, Statistics, Computer Science or Social Science. As a result, a massive literature on networks exists – and it is challenging for biologists to dive into it. Indeed, we are not used to explore the bibliography outside our research domain. However, without any doubt you will highly benefit from a round trip in this literature, provided that you make the effort to learn the appropriate vocabulary in the field. Reviews [13] and reference books [24, 28] are obviously good entry-points for developing your network skills. For instance, the reader can find thousands of articles dedicated to the network clustering problem, dealing with different particular settings. This quantity of knowledge must be a source of inspiration. Lastly, why would you not experience attending one of the complex networks' conferences (e.g. NetSci conference series)?

Conclusion

The 9 tips presented here must be a way for the data analyst to get a foot in the door of network data analysis. These tips are not exclusive and we are aware of other network-based questions that deserve a special interest, including diffusion on networks for instance. Still, the network non-specialist must be confident in his ability to learn, step by step, the network concepts and methods with a productive effect on his scientific questions.

Acknowledgements

This work was partially supported by the grant ANR-18-CE02-0010-01 of the French National Research Agency ANR (project EcoNet).

References

1. T. Ideker and R. Nussinov, "Network approaches and applications in biology," *PLoS computational biology*, vol. 13, no. 10, p. e1005771, 2017.

2. M. Zitnik, R. Sosi, and J. Leskovec, "Prioritizing network communities," *Nat Commun*, vol. 9, no. 1, p. 2544, 2018.

3. B. Wang, A. Pourshafeie, M. Zitnik, J. Zhu, C. D. Bustamante, S. Batzoglou, and J. Leskovec, "Network enhancement as a general method to denoise weighted biological networks," *Nat Commun*, vol. 9, no. 1, p. 3108, 2018.

4. P. P. Staniczenko, J. C. Kopp, and S. Allesina, "The ghost of nestedness in ecological networks," *Nature communications*, vol. 4, p. 1391, 2013.

5. P. Expert, T. S. Evans, V. D. Blondel, and R. Lambiotte, "Uncovering space-independent communities in spatial networks," *Proceedings of the National Academy of Sciences*, vol. 108, no. 19, pp. 7663–7668, 2011.

6. M. A. Fortuna, R. G. Albaladejo, L. Fernández, A. Aparicio, and J. Bascompte, "Networks of spatial genetic variation across species," *Proceedings of the National Academy of Sciences*, vol. 106, no. 45, pp. 19044–19049, 2009.

7. M. A. Carey and J. A. Papin, "Ten simple rules for biologists learning to program," *PLoS Comput. Biol.*, vol. 14, no. 1, p. e1005871, 2018.

8. G. K. Sandve, A. Nekrutenko, J. Taylor, and E. Hovig, "Ten simple rules for reproducible computational research," *PLoS Comput. Biol.*, vol. 9, no. 10, p. e1003285, 2013.

9. M. Rubinov and O. Sporns, "Complex network measures of brain connectivity: uses and interpretations," *Neuroimage*, vol. 52, no. 3, pp. 1059–1069, 2010.

10. D. R. Farine and H. Whitehead, "Constructing, conducting and interpreting animal social network analysis," *J Anim Ecol*, vol. 84, no. 5, pp. 1144–1163, 2015.

11. A. Costa, A. M. M. Gonzalez, K. Guizien, A. M. Doglioli, J. M. Gomez, A. Petrenko, and S. Allesina, "Ecological networks: Pursuing the shortest path, however narrow and crooked," *bioRxiv*, p. 475715, 2018.

12. B. C. Van Wijk, C. J. Stam, and A. Daffertshofer, "Comparing brain networks of different size and connectivity density using graph theory," *PloS one*, vol. 5, no. 10, p. e13701, 2010.

13. S. Fortunato and D. Hric, "Community detection in networks: A user guide," *Physics Reports*, vol. 659, pp. 1–44, 2016.

14. M. E. Newman and E. A. Leicht, "Mixture models and exploratory analysis in networks," *Proceedings of the National Academy of Sciences*, vol. 104, no. 23, pp. 9564–9569, 2007.

15. R. F. Betzel, J. D. Medaglia, and D. S. Bassett, "Diversity of meso-scale architecture in human and non-human connectomes," *Nature Communications*, vol. 9, no. 1, p. 346, 2018.

16. J.-J. Daudin, F. Picard, and S. Robin, "A mixture model for random graphs," *Statistics and computing*, vol. 18, no. 2, pp. 173–183, 2008.

17. S. Kéfi, V. Miele, E. A. Wieters, S. A. Navarrete, and E. L. Berlow, "How Structured Is the Entangled Bank? The Surprisingly Simple Organization of Multiplex Ecological Networks Leads to Increased Persistence and Resilience," *PLoS Biol.*, vol. 14, no. 8, p. e1002527, 2016.

18. M. Mariadassou, S. Robin, C. Vacher, *et al.*, "Uncovering latent structure in valued graphs: a variational approach," *The Annals of Applied Statistics*, vol. 4, no. 2, pp. 715–742, 2010.

19. D. B. Stouffer, M. Sales-Pardo, M. I. Sirer, and J. Bascompte, "Evolutionary conservation of species' roles in food webs," *Science*, vol. 335, no. 6075, pp. 1489–1492, 2012.

20. P. Goyal and E. Ferrara, "Graph embedding techniques, applications, and performance: A survey," *Knowledge-Based Systems*, vol. 151, pp. 78–94, 2018.

21. W. L. Hamilton, R. Ying, and J. Leskovec, "Representation learning on graphs: Methods and applications," *IEEE Data Engineering Bulletin*, 2017.

22. V. Blondel, J. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *Journal of Statistical Mechanics : Theory and Experiment*, 2008.

23. A. Grover and J. Leskovec, "node2vec: Scalable feature learning for networks," in *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 855–864, ACM, 2016.

24. G. Bianconi, *Multilayer Networks: Structure and Function*. Oxford university press, 2018.

25. G. Rossetti and R. Cazabet, "Community discovery in dynamic networks: a survey," *ACM Computing Surveys (CSUR)*, vol. 51, no. 2, p. 35, 2018.

26. P. Holme, "Modern temporal network theory: a colloquium," *The European Physical Journal B*, vol. 88, no. 9, p. 234, 2015.

27. M. Ohlmann, V. Miele, S. Dray, L. Chalmandrier, L. O'Connor, and W. Thuiller, "Diversity indices for ecological networks: a unifying framework using Hill numbers," *Ecology letters*, 2019.

28. M. Newman, *Networks*. Oxford university press, 2018.