

netrd: A library for network reconstruction and graph distances

Stefan McCabe¹, Leo Torres¹, Timothy LaRock¹, Syed Arefinul Haque¹, Chia-Hung Yang¹, Harrison Hartle¹, and Brennan Klein^{1, 2}

¹ Network Science Institute, Northeastern University, Boston, MA, USA ² Laboratory for the Modeling of Biological and Socio-Technical Systems, Northeastern University, Boston, USA

DOI: [10.21105/joss.02990](https://doi.org/10.21105/joss.02990)

Software

- [Review](#) ↗
- [Repository](#) ↗
- [Archive](#) ↗

Editor: [Arfon Smith](#) ↗

Reviewers:

- [@adi3](#)
- [@sonal-raj](#)

Submitted: 29 October 2020

Published: 01 June 2021

License

Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC BY 4.0](#)).

Statement of need

Complex systems throughout nature and society are often best represented as *networks*. Over the last two decades, alongside the increased availability of large network datasets, we have witnessed the rapid rise of network science ([Amaral & Ottino, 2004](#); [Barabási, 2016](#); [Newman, 2018](#); [Vespignani et al., 2008](#)). This field is built around the idea that an increased understanding of the complex structural properties of a variety systems will allow us to better observe, predict, and even control the behavior of these systems.

However, for many systems, the data we have access to is not a direct description of the underlying network. More and more, we see the drive to study networks that have been inferred or reconstructed from non-network data—in particular, using *time series* data from the nodes in a system to infer likely connections between them ([Brugere et al., 2018](#); [Runge, 2018](#)). Selecting the most appropriate technique for this task is a challenging problem in network science. Different reconstruction techniques usually have different assumptions, and their performance varies from system to system in the real world. One way around this problem could be to use several different reconstruction techniques and compare the resulting networks. However, network comparison is also not an easy problem, as it is not obvious how best to quantify the differences between two networks, in part because of the diversity of tools for doing so.

The netrd Python package seeks to address these two parallel problems in network science.

Summary

netrd offers, to our knowledge, the most extensive collection of both network reconstruction techniques and network comparison techniques (often referred to as *graph distances*) in a single library. Below, we expand on these two main functionalities of the netrd package.

The first core use of netrd is to reconstruct networks from time series data. Given time series data, TS , of the behavior of N nodes / components / sensors of a system over the course of L timesteps, and given the assumption that the behavior of every node, v_i , may have been influenced by the past behavior of other nodes, v_j , there are dozens of techniques that can be used to infer which connections, e_{ij} , are likely to exist between the nodes. That is, we can use one of many *network reconstruction* techniques to create a network representation, G_T , that attempts to best capture the relationships between the time series of every node in TS . netrd lets users perform this network reconstruction task using 17 different techniques. This means that up to 17 different networks can be formed created from a single time series dataset. For example, in [Figure 1](#) we show the outputs of 15 different reconstruction techniques applied

to time series data generated from an example network (Barucca, 2014; Donges et al., 2009; Edelman & Rao, 2005; Friedman et al., 2008; Hoang et al., 2019; Ledoit & Wolf, 2003; Mishchenko et al., 2011; Peixoto, 2019; Sheikhattar et al., 2018; Stetter et al., 2012; Sugihara et al., 2012; Zeng et al., 2013).

Practitioners often apply these network reconstruction algorithms to real time series data. For example, in neuroscience, researchers often try to reconstruct functional networks from time series readouts of neural activity (Mishchenko et al., 2011). In economics, researchers can infer networks of influence between companies based on time series of changes in companies' stock prices (Squartini et al., 2018). At the same time, it is often quite helpful having the freedom to *simulate* arbitrary time series dynamics on randomly generated networks. This provides a controlled setting to assess the performance of network reconstruction algorithms. For this reason, the `netrd` package also includes a number of different techniques for simulating dynamics on networks.

The second core use of `netrd` addresses a common goal when studying networks: describing and quantifying the differences between two networks. This is a challenging problem, as there are countless axes upon which two networks can differ; as such, a number of *graph distance* measures have emerged over the years attempting to address this problem. As is the case for many hard problems in network science, it can be difficult to know which (of many) measures are suited for a given setting. In `netrd`, we consolidate over 20 different graph distance measures into a single package (Bagrow & Bollt, 2019; Berlingerio et al., 2012; Carpi et al., 2011; Chen et al., 2018; De Domenico & Biamonte, 2016; Donnat & Holmes, 2018; Golub & Loan, 2013; Hamming, 1950; Hammond et al., 2013; Ipsen & Mikhailov, 2002; Jaccard, 1901; Jurman et al., 2011, 2015; Koutra et al., 2016; Mellor & Grusovin, 2019; Monnig & Meyer, 2018; Schieber et al., 2017; Torres et al., 2019; Tsitsulin et al., 2018). Figure 2 shows an example of just how different these measures can be when comparing two networks, G_1 and G_2 . This submodule in `netrd` has already been used in recent work with a novel characterization of the graph distance literature (Hartle et al., 2020).

This package builds on commonly used Python packages (e.g. `networkx` (Hagberg et al., 2008), `numpy` (Harris et al., 2020), `scipy` (Virtanen et al., 2020)) and is already a widely used resource for network scientists and other multidisciplinary researchers. With ongoing open-source development, we see this as a tool that will continue to be used by all sorts of researchers to come.

Related software packages

In the network reconstruction literature, there are often software repositories that detail a single technique or a few related ones. For example Lizier (2014) implemented a Java package (portable to Python, octave, R, Julia, Clojure, MATLAB) that uses information-theoretic approaches for inferring network structure from time-series data (Lizier, 2014); Runge et al. (2019) created a Python package that combines linear or nonlinear conditional independence tests with a causal discovery algorithm to reconstruct causal networks from large-scale time series datasets (Runge et al., 2019). These are two examples of powerful and widely used packages though neither includes as wide-ranging techniques as `netrd` (nor were they explicitly designed to). In the graph distance literature, the same trend is broadly true: many one-off software repositories exist for specific measures. However, there are some packages that do include multiple graph distances; for example, Wills (2017) created a `NetComp` package that includes several variants of a few distance measures included here (Wills, 2017).

Figures

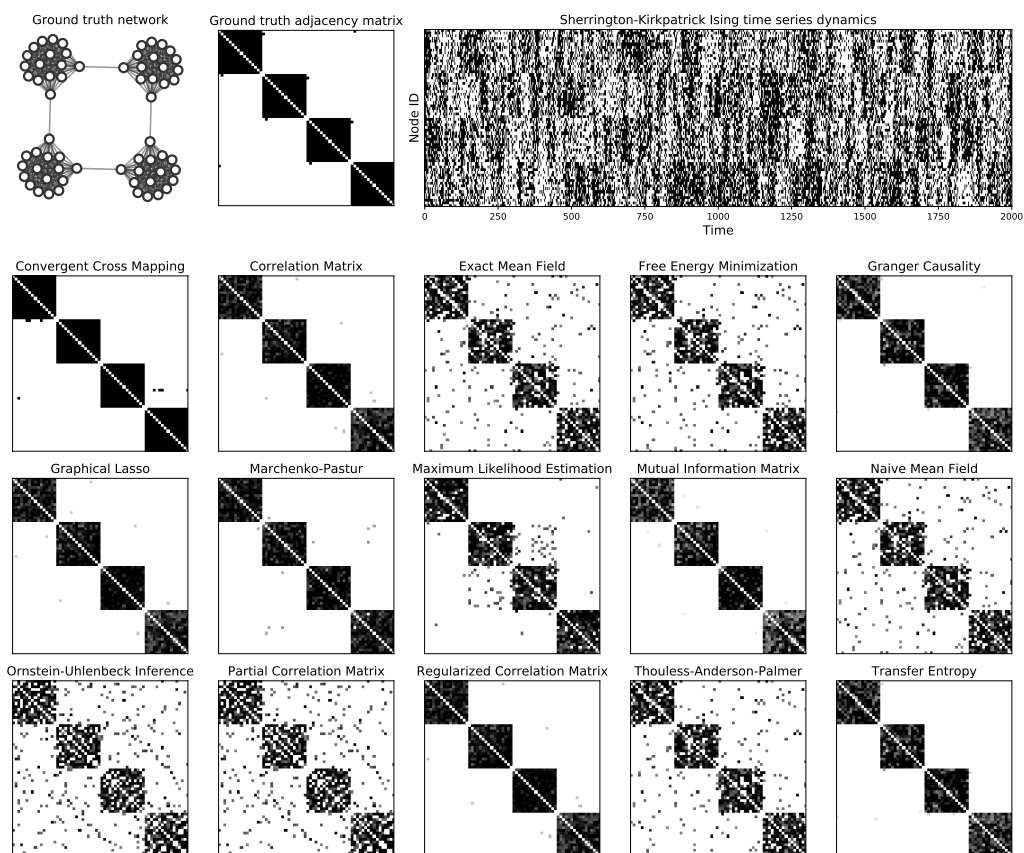


Figure 1: Example of the network reconstruction pipeline. (Top row) A sample network, its adjacency matrix, and an example time series, TS , of node-level activity simulated on the network. (Bottom rows) The outputs of 15 different network reconstruction algorithms, each using TS to create a new adjacency matrix that captures key structural properties of the original network.

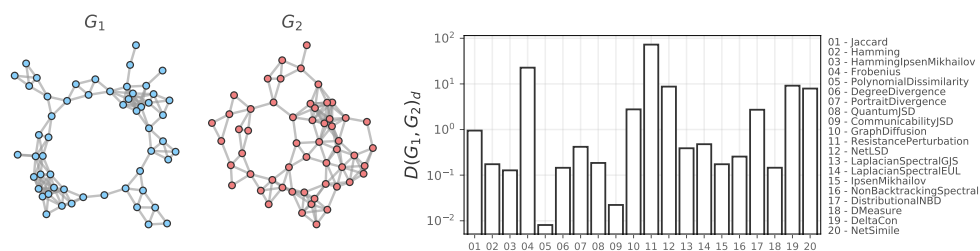


Figure 2: Example of the graph distance measures in `netrd`. Here, we measure the graph distance between two networks using 20 different distance measures from `netrd`.

Acknowledgements

The authors thank Kathryn Coronges, Mark Giannini, and Alessandro Vespignani for contributing to the coordination of the 2019 Network Science Institute “Collabathon,” where much of the development of this package began. The authors acknowledge the support of

ten other contributors to this package: Guillaume St-Onge, Andrew Mellor, Charles Murphy, David Saffo, Carolina Mattsson, Ryan Gallagher, Matteo Chinazzi, Jessica Davis, Alexander J. Gates, and Anton Tsitulin. **Funding:** This research was supported by the Network Science Institute at Northeastern University.

References

- Amaral, L. A. N., & Ottino, J. M. (2004). Complex networks. *European Physical Journal B - Condensed Matter*, 38(2), 147–162. <https://doi.org/10.1140/epjb/e2004-00110-5>
- Bagrow, J. P., & Boltt, E. M. (2019). An information-theoretic, all-scales approach to comparing networks. *Applied Network Science*, 4(45), 1–15. <https://doi.org/10.1007/s41109-019-0156-x>
- Barabási, A.-L. (2016). *Network Science*. Cambridge University Press. <https://doi.org/10.1177/0094306116681814>
- Barucca, P. (2014). Localization in covariance matrices of coupled heterogenous Ornstein-Uhlenbeck processes. *Physical Review E*, 90(6), 1–5. <https://doi.org/10.1103/PhysRevE.90.062129>
- Berlingerio, M., Koutra, D., Eliassi-Rad, T., & Faloutsos, C. (2012). Netsimile: A scalable approach to size-independent network similarity. *arXiv:1209.2684*. <https://arxiv.org/abs/1209.2684>
- Brugere, I., Gallagher, B., & Berger-Wolf, T. Y. (2018). Network structure inference, a survey. *ACM Computing Surveys*, 51(2), 1–39. <https://doi.org/10.1145/3154524>
- Carpi, L. C., Rosso, O. A., Saco, P. M., & Ravetti, M. G. (2011). Analyzing complex networks evolution through Information Theory quantifiers. *Physics Letters A*, 375(4), 801–804. <https://doi.org/10.1016/j.physleta.2010.12.038>
- Chen, D., Shi, D. D., Qin, M., Xu, S. M., & Pan, G. J. (2018). Complex network comparison based on communicability sequence entropy. *Physical Review E*, 98(1), 1–8. <https://doi.org/10.1103/PhysRevE.98.012319>
- De Domenico, M., & Biamonte, J. (2016). Spectral entropies as information-theoretic tools for complex network comparison. *Physical Review X*, 6(4), 34–37. <https://doi.org/10.1103/PhysRevX.6.041062>
- Donges, J. F., Zou, Y., Marwan, N., & Kurths, J. (2009). The backbone of the climate network. *Europhysics Letters*, 87(4). <https://doi.org/10.1209/0295-5075/87/48007>
- Donnat, C., & Holmes, S. (2018). Tracking network dynamics: A survey using graph distances. *Annals of Applied Statistics*, 12(2), 971–1012. <https://doi.org/10.1214/18-AOAS1176>
- Edelman, A., & Rao, N. R. (2005). Random matrix theory. *Acta Numerica*, 14, 233–297. <https://doi.org/10.1017/S0962492904000236>
- Friedman, J., Hastie, T., & Tibshirani, R. (2008). Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3), 432–441. <https://doi.org/10.1093/biostatistics/kxm045>
- Golub, G. H., & Loan, C. F. van. (2013). *Matrix Computations* (Fourth). Johns Hopkins University Press. ISBN: 9781421407944
- Hagberg, A. A., Schult, D. A., & Swart, P. J. (2008). Exploring network structure, dynamics, and function using NetworkX. In G. Varoquaux, T. Vaught, & J. Millman (Eds.), *Proceedings of the 7th python in science conference* (pp. 11–15). http://conference.scipy.org/proceedings/SciPy2008/paper_2/

- Hamming, R. W. (1950). Error detecting and error correcting codes. *The Bell System Technical Journal*, 29(2), 147–160. [https://doi.org/10.1016/S0016-0032\(23\)90506-5](https://doi.org/10.1016/S0016-0032(23)90506-5)
- Hammond, D. K., Gur, Y., & Johnson, C. R. (2013). Graph diffusion distance: A difference measure for weighted graphs based on the graph Laplacian exponential kernel. *2013 IEEE Global Conference on Signal and Information Processing, GlobalSIP 2013 - Proceedings*, 419–422. <https://doi.org/10.1109/GlobalSIP.2013.6736904>
- Harris, C. R., Millman, K. J., Walt, S. J. van der, Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., Kerkwijk, M. H. van, Brett, M., Haldane, A., R'io, J. F. del, Wiebe, M., Peterson, P., ... Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, 585(7825), 357–362. <https://doi.org/10.1038/s41586-020-2649-2>
- Hartle, H., Klein, B., McCabe, S., St-Onge, G., Murphy, C., Daniels, A., & Hébert-Dufresne, L. (2020). Network comparison and the within-ensemble graph distance. *Proceedings of the Royal Society A*, 1–18. <https://doi.org/10.1098/rspa.2019.0744>
- Hoang, D.-T., Song, J., Periwai, V., & Jo, J. (2019). Network inference in stochastic systems from neurons to currencies: Improved performance at small sample size. *Physical Review E*, 99, 023311. <https://doi.org/10.1103/PhysRevE.99.023311>
- Ipsen, M., & Mikhailov, A. S. (2002). Evolutionary reconstruction of networks. *Physical Review E*, 66(4), 4. <https://doi.org/10.1103/PhysRevE.66.046109>
- Jaccard, P. (1901). Etude de la distribution florale dans une portion des Alpes et du Jura. *Bulletin de La Societe Vaudoise Des Sciences Naturelles*, 37, 547–579. <https://doi.org/10.5169/seals-266450>
- Jurman, G., Visintainer, R., Filosi, M., Riccadonna, S., & Furlanello, C. (2015). The HIM glocal metric and kernel for network comparison and classification. *Proceedings of the 2015 IEEE International Conference on Data Science and Advanced Analytics, DSAA 2015*, 1–10. <https://doi.org/10.1109/DSAA.2015.7344816>
- Jurman, G., Visintainer, R., & Furlanello, C. (2011). An introduction to spectral distances in networks. In N. N. W. B. Apolloni & others (Eds.), *Neural nets WIRN10: Proceedings of the 20th italian workshop on neural nets* (Vol. 226, pp. 227–234). <https://doi.org/10.3233/978-1-60750-692-8-227>
- Koutra, D., Vogelstein, J. T., & Faloutsos, C. (2016). DELTACON: A principled massive-graph similarity function. *ACM Transactions on Knowledge Discovery from Data*, 10(3), 162–170. <https://doi.org/10.1145/2824443>
- Ledoit, O., & Wolf, M. (2003). Honey, I shrunk the sample covariance matrix. *SSRN*, 1–22. <https://doi.org/10.2139/ssrn.433840>
- Lizier, J. T. (2014). JIDT: An information-theoretic toolkit for studying the dynamics of complex systems. *Frontiers in Robotics and AI*, 1–11. <https://doi.org/10.3389/frobt.2014.00011>
- Mellor, A., & Grusovin, A. (2019). Graph comparison via the nonbacktracking spectrum. *Physical Review E*, 99, 052309. <https://doi.org/10.1103/PhysRevE.99.052309>
- Mishchenko, Y., Vogelstein, J. T., & Paninski, L. (2011). A Bayesian approach for inferring neuronal connectivity from calcium fluorescent imaging data. *Annals of Applied Statistics*, 5(2 B), 1229–1261. <https://doi.org/10.1214/09-AOAS303>
- Monnig, N. D., & Meyer, F. G. (2018). The resistance perturbation distance: A metric for the analysis of dynamic networks. *Discrete Applied Mathematics*, 236, 347–386. <https://doi.org/10.1016/j.dam.2017.10.007>
- Newman, M. E. J. (2018). *Networks: An Introduction* (Second, pp. 1–784). Oxford University Press. <https://doi.org/10.1093/acprof:oso/9780199206650.001.0001>

- Peixoto, T. P. (2019). Network reconstruction and community detection from dynamics. *Physical Review Letters*, 123, 128301. <https://doi.org/10.1103/PhysRevLett.123.128301>
- Runge, J. (2018). Causal network reconstruction from time series: From theoretical assumptions to practical estimation. *Chaos*, 28(7). <https://doi.org/10.1063/1.5025050>
- Runge, J., Nowack, P., Kretschmer, M., Flaxman, S., & Sejdinovic, D. (2019). Detecting and quantifying causal associations in large nonlinear time series datasets. *Science Advances*, 5(11), eaau4996. <https://doi.org/10.1126/sciadv.aau4996>
- Schieber, T. A., Carpi, L. C., Díaz-Guilera, A., Pardalos, P. M., Masoller, C., & Ravetti, M. G. (2017). Quantification of network structural dissimilarities. *Nature Communications*, 8(13928), 1–10. <https://doi.org/10.1038/ncomms13928>
- Sheikhhattar, A., Miran, S., Liu, J., Fritz, J. B., Shamma, S. A., Kanold, P. O., & Babadi, B. (2018). Extracting neuronal functional network dynamics via adaptive Granger causality analysis. *Proceedings of the National Academy of Sciences*, 201718154. <https://doi.org/10.1073/pnas.1718154115>
- Squartini, T., Gabrielli, A., Garlaschelli, D., Gili, T., Bifone, A., & Caccioli, F. (2018). Complexity in neural and financial systems: From time-series to networks. *Complexity*, 1–2. <https://doi.org/10.1155/2018/3132940>
- Stetter, O., Battaglia, D., Soriano, J., & Geisel, T. (2012). Model-free reconstruction of excitatory neuronal connectivity from calcium imaging signals. *PLoS Computational Biology*, 8(8). <https://doi.org/10.1371/journal.pcbi.1002653>
- Sugihara, G., May, R., Ye, H., Hsieh, C., Deyle, E., Fogarty, M., & Munch, S. (2012). Detecting causality in complex ecosystems. *Science*, 338, 496–500. <https://doi.org/10.1126/science.1227079>
- Torres, L., Suárez-Serrato, P., & Eliassi-Rad, T. (2019). Non-backtracking cycles: Length spectrum theory and graph mining applications. *Applied Network Science*, 4(1), 41. <https://doi.org/10.1007/s41109-019-0147-y>
- Tsitsulin, A., Mottin, D., Karras, P., Bronstein, A., & Müller, E. (2018). NetLSD: Hearing the shape of a graph. *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2347–2356. <https://doi.org/10.1145/3219819.3219991>
- Vespignani, A., Barthélemy, M., & Barrat, A. (2008). *Dynamical Processes on Complex Networks* (p. 367). Cambridge University Press. <https://doi.org/10.1080/00107510903084036>
- Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M., Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., ... SciPy 1.0 Contributors. (2020). SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17, 261–272. <https://doi.org/10.1038/s41592-019-0686-2>
- Wills, P. (2017). *NetComp*. <https://github.com/peterewills/NetComp>
- Zeng, H. L., Alava, M., Aurell, E., Hertz, J., & Roudi, Y. (2013). Maximum likelihood reconstruction for Ising models with asynchronous updates. *Physical Review Letters*, 110(21), 1–5. <https://doi.org/10.1103/PhysRevLett.110.210601>