The Journal of Open Source Software

MiNAA: Microbiome Network Alignment Algorithm

Reed Nelson[1], **Rosa Aghdam**[2], **and Claudia Solis-Lemus**[2,3¶]

1 Department of Computer Science, University of Wisconsin-Madison, Madison, WI, United States of America **2** Wisconsin Institute for Discovery, University of Wisconsin-Madison, Madison, WI, United States of America **3** Department of Plant Pathology, University of Wisconsin-Madison, Madison, WI, United States of America ¶ Corresponding author

DOI: 10.xxxxxx/draft

Software

- Review
- Repository
- Archive

Editor: Gracielle Higino

Reviewers:

- @Becheler
- @waynebhayes

Submitted: 12 December 2022
Published: unpublished

Summary

A microbial network is a mathematical representation of a microbial community where nodes represent microbes and edges represent interactions. It is well-recognized that microbes are among the main drivers of biological phenotypes in soil, plants, and animals alike, and thus, their study has vast implications for soil, plant and human health. In particular, recognizing the microbial, environmental, and agricultural factors that drive plant and soil phenotypes is crucial to comprehend processes connected to plant and soil health, to identify global practices of sustainable agriculture, as well as to predict plant and soil responses to environmental perturbations such as climate change.

The adaptability of microbes to thrive in every environment poses challenges for scientists who try to understand their communities. Indeed, two microbial communities with the exact same players can interact differently depending on the environmental conditions. It is thus desirable to identify commonalities and differences on two microbial networks, hence the need for computational tools that can match (or *align*) them.

Statement of need

Our Microbiome Network Alignment Algorithm (MiNAA) aligns two microbial networks using a combination of the GRAph ALigner (GRAAL) algorithm (Kuchaiev et al., 2010) and the Hungarian algorithm (Kuhn, 1955; Pilgrim, 1995). Network alignment algorithms find pairs of nodes (one node from the first network and the other node from the second network) that have the highest *similarity*. Traditionally, similarity has been defined as topological similarity such that the neighborhoods around the two nodes are similar. Recent implementations of network alignment methods such as NETAL (Neyshabur et al., 2013) and L-GRAAL (Malod-Dognin & Pržulj, 2015) also include measures of biological similarity, yet these methods are restricted to one specific type of biological similarity (e.g. sequence similarity in L-GRAAL). Our work extends existing network alignment implementations by allowing *any* type of biological similarity to be input by the user. This flexibility allows the user to choose whatever measure of biological similarity is suitable for the study at hand. In addition, unlike most existing network alignment methods that are tailored for protein or gene interaction networks (Chen et al., 2020; Ma & Liao, 2020), our work is the first one suited for microbiome networks.

Description of the algorithm

Input. Two networks, represented by adjacency matrices, are the main inputs for our algorithm. Let G and H be such input networks such that $|G| \leq |H|$, where $|G|$ indicates the size of G. Optionally, the user may add biological similarity (as a $|G| \times |H|$ matrix) to be weighed into the alignment. This matrix could include gene similarity, phylogenetic similarity, functional similarity, among others. Our algorithm also includes additional options that allow the user to specify how much weight should be placed on biological versus topological information. Specifics on all the input arguments can be found on GitHub https://github.com/solislemuslab/minaa.

Algorithms. For each input network (G, H), we calculate the *graphlet degree vector*, also denoted the *node signature* of each node. This topological descriptor characterizes a node based on its local neighborhood within a 5-node radius (Pržulj, 2007).

Currently, we use the same algorithm for calculating node signatures as in GraphCrunch2 (Kuchaiev et al., 2011) which is a $O(ed^3)$ algorithm where e is the number of edges, and d is the maximal node degree. Future work will see this implementation replaced by ORCA (Hočevar & Demšar, 2016), a functional equivalent with runtime $O(ed^2)$ for sparse graphs such as microbial networks.

Using the node signatures, we calculate the topological difference between each node $g \in G$ and $h \in H$, according to the formula described in (Milenković & Pržulj, 2008). In keeping with the flexibility of the original formula, we include a parameter α, such that the topological distance $T_{i,j} = \alpha \cdot S_{i,j} + (1 - \alpha) \cdot N_{i,j}$, where $S_{i,j}$ represents the difference in node signatures between nodes i and j, and $N_{i,j}$ represents the difference in degree between the nodes i and j. It is recommended to use the default $\alpha = 1$. We store the resulting values in the topological cost matrix T whose computation has complexity $O(|G||H|)$.

At this time, if the user provided a biological cost matrix B, the two matrices are combined into an overall cost matrix O, such that $O_{i,j} = \beta \cdot T_{i,j} + (1 - \beta) \cdot B_{i,j}$. The user can specify the value of the parameter β, and by default, we use $\beta = 1$ which represents the case in which only topological information is considered.

The overall cost matrix, O, is the input to the Hungarian algorithm (Kuhn, 1955; Pilgrim, 1995), an $O(|G|^3)$ solution to the *assignment problem*. It is important to note that the Hungarian algorithm will align every node G to some node in H. Optimally, we should only align nodes which we have sufficient confidence ought to be aligned, but this *partial assignment problem* is computationally harder (likely, NP hard), and beyond the scope of this work. We show a graphical abstract of our algorithm in Figure 1.

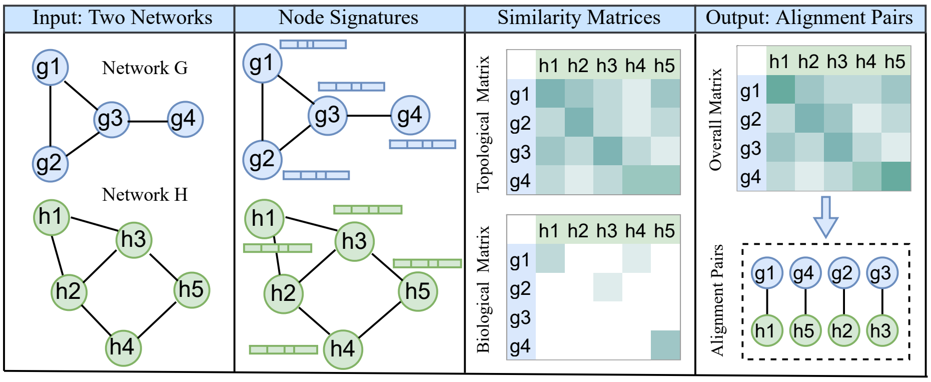

Figure 1: Graphical abstract of MiNAA. The algorithm takes two networks as input (G, H), and for each node in the networks, it computes a node signature vector based on the topological information of the node's neighborhood. With the node signatures, a topological cost matrix is computed which is then combined with a biological cost matrix input by the user. This method then applies the Hungarian algorithm to the overall cost matrix, resulting in a globally optimal alignment. Note that we show similarity matrices rather than cost matrices in this figure. While the algorithm works with cost matrices, similarity matrices are more intuitive for the graphical representation, and it is easy to convert one for the other with a transformation.

Output. The algorithm's main output is the complete alignment of G and H. That is, we return a list of pairs of nodes, one node from network G and one node from network H that have been identified by the algorithm as having the lowest alignment cost. We also return an alignment score matrix with dimensions $|G| \times |H|$ with the alignment score of each pair of nodes (the highest value, the more evidence for alignment). Along the way, G and H's node signature sets are saved as files, as are the intermediate matrices T, B, and O.

Simulations

We simulate networks with the R package SPIEC-EASI (SParse InversE Covariance Estimation for Ecological Association Inference) (**kurtz2015sparse?**). We focus on the "Cluster" network topology since cluster networks are conducive to speculative ecological scenarios. Indeed, for microbial communities that inhabit many discontinuous niches (clusters) and have minimal interactions between niches, cluster networks may serve as archetypal models (Peschel et al., 2021; Yang et al., 2019).

A network is simulated with certain number of nodes, and then a proportion of its edges are flipped to produce a second network. Because the second network is just a perturbation of the first network, we expect our alignment method to align the same nodes correctly. For example, for the original simulated network (Network G) with 10 nodes and with 5% edge change rate, five out of every 100 edges in the adjacency matrix that are 1 are replaced by 0, and similarly, five out of every 100 edges that are 0 are replaced with 1. The modified network from Network G is named Network H. Then our algorithm is applied on the original and modified networks to detect the alignment pairs. This results in a alignment matrix with its elements represented as a_{ij}. Each a_{ij} shows the similarity value of the i's node in Network G with j's node in Network H. We expect the alignment matrix to be close to the identity matrix.

We iterate over different numbers of nodes (10, 30, 50, 100, 250 and 500), and different proportions of edges changed (0.05 (5%), 0.1, and 0.9). To achieve reliable results, each scenario is repeated 30 times and the average of the alignment matrices is computed. We expect Network G to be properly aligned with a perturbation of itself with edge change rate of 0.05 or 0.1, but not with an edge change rate of 0.9.

Figure 2 displays the results of the averaged similarity matrices for all simulated networks, with rows representing the edge change rate and columns representing the number of nodes. For instance, for 10 nodes with 0.05 edge changes, the heatmap displays the mean of 30 alignment matrices. Given that we observe diagonal matrices for a small rate of edge change (0.05 and 0.1), we can conclude that our algorithm correctly aligns nodes under small perturbations of the networks. However, the results for the same networs with the edge change rate of 0.9 is far from diagonal which is also what we expected.

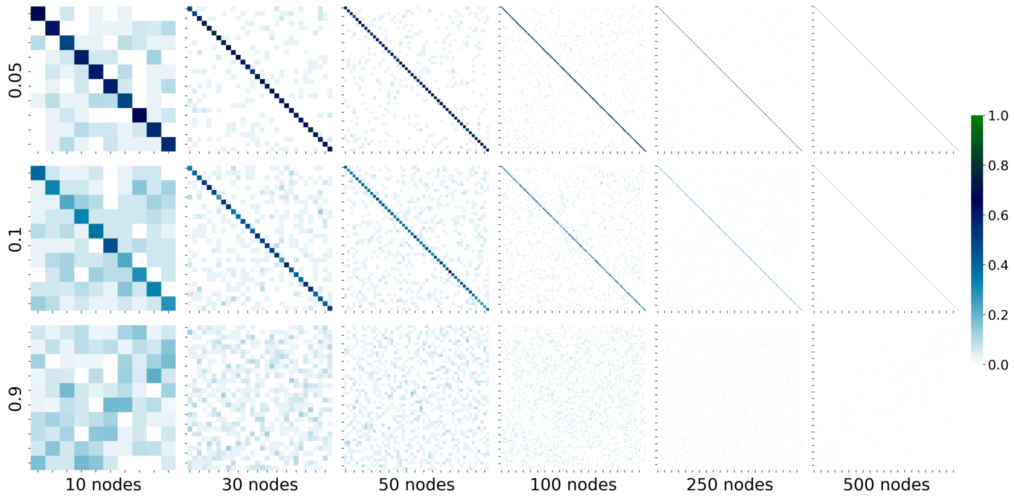

Figure 2: Averaged alignment matrices for all simulated networks, rows representing the rate of altered edge change and columns representing the number of nodes. As expected, for 0.05 or 0.1 edge change rate, the alignment matrices are close to identity matrices illustrating our method's ability to align the same nodes on perturbed networks.

We also present results on running time (see table below). We average the runtime over 90 alignments (30 replicates with 3 levels of edge change rate each) for each network size (as in the simulated data described above). Benchmarking was done in a single thread on a 72 core/3.10GHz Intel CPU, with 1TB available RAM.

Number of nodes	Average Runtime (ms)
10	25.100
30	40.167
50	66.767
100	139.133
250	972.433
500	7603.233

Acknowledgements

This work was supported by the Department of Energy [DE-SC0021016 to CSL]. The authors thank Arnaud Becheler for meaningful discussions about ORCA.

References

Chen, Y., Zhu, Y., Zhong, M., Peng, R., & Liu, J. (2020). GPPIAL: A new global PPI network aligner based on orthologs. *2020 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, 100–107. https://doi.org/10.1109/bibm49941.2020.9313552

Hočevar, T., & Demšar, J. (2016). Computation of graphlet orbits for nodes and edges in sparse graphs. *Journal of Statistical Software*, *71*, 1–24. https://doi.org/10.18637/jss.v071.i10

Kuchaiev, O., Milenković, T., Memišević, V., Hayes, W., & Pržulj, N. (2010). Topological network alignment uncovers biological function and phylogeny. *Journal of the Royal Society Interface*, *7*(50), 1341–1354. https://doi.org/10.1098/rsif.2010.0063

Kuchaiev, O., Stevanović, A., Hayes, W., & Pržulj, N. (2011). GraphCrunch 2: Software tool for network modeling, alignment and clustering. *BMC Bioinformatics*, *12*(1), 1–13. https://doi.org/10.1186/1471-2105-12-24

Kuhn, H. W. (1955). The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, *2*(1-2), 83–97. https://doi.org/10.1002/nav.20053

Ma, C.-Y., & Liao, C.-S. (2020). A review of protein–protein interaction network alignment: From pathway comparison to global alignment. *Computational and Structural Biotechnology Journal*, *18*, 2647–2656. https://doi.org/10.1016/j.csbj.2020.09.011

Malod-Dognin, N., & Pržulj, N. (2015). L-GRAAL: Lagrangian graphlet-based network aligner. *Bioinformatics*, *31*(13), 2182–2189. https://doi.org/10.1093/bioinformatics/btv130

Milenković, T., & Pržulj, N. (2008). Uncovering biological network function via graphlet degree signatures. *Cancer Informatics*, *6*, CIN–S680. https://doi.org/10.4137/cin.s680

Neyshabur, B., Khadem, A., Hashemifar, S., & Arab, S. S. (2013). NETAL: A new graph-based method for global alignment of protein–protein interaction networks. *Bioinformatics*, *29*(13), 1654–1662. https://doi.org/10.1093/bioinformatics/btt202

Peschel, S., Müller, C. L., Mutius, E. von, Boulesteix, A.-L., & Depner, M. (2021). NetCoMi: Network construction and comparison for microbiome data in r. *Briefings in Bioinformatics*, *22*(4), bbaa290. https://doi.org/10.1101/2020.07.15.195248

Pilgrim, R. (1995). Tutorial on implementation of munkres' assignment algorithm. *Lecture Notes*. https://doi.org/10.13140/RG.2.1.3572.3287

Pržulj, N. (2007). Biological network comparison using graphlet degree distribution. *Bioinformatics*, *23*(2), e177–e183. https://doi.org/10.1093/bioinformatics/btq091

Yang, P., Yu, S., Cheng, L., & Ning, K. (2019). Meta-network: Optimized species-species network analysis for microbial communities. *BMC Genomics*, *20*(2), 143–151. https://doi.org/10.1186/s12864-019-5471-1