Correlation and Co-occurrence Networks in Taxonomic Classification

Jim Brunner

August 16, 2017

1 Introduction

[5][3][6][2][7][9][1][10]

2 Network Construction

2.1 Correlation construction

We construct a network based on sample correlation, as in [4]. With N samples, that is:

$$\rho_{xy} = \frac{1}{N}\frac{(\boldsymbol{x}-\mu_x\mathbf{1})\cdot(\boldsymbol{y}-\mu_y\mathbf{1})}{\sigma_x\sigma_y}$$

where μ_x is the mean abundance of taxa x, and σ_x is the standard deviation of abundance of taxa x across samples. This can be constructed by modifying the data matrix and then multiplying with it's transpose, making it much faster to compute than the binning method. In expectation, this the co-variance divided by the product of the variances. I then construct a graph with weighted edges

$$w_{ik} = \begin{cases} \rho_{\boldsymbol{r}_i\boldsymbol{r}_k} & \rho_{\boldsymbol{r}_i\boldsymbol{r}_k} \geq 0.8 \\ 0 & \rho_{\boldsymbol{r}_i\boldsymbol{r}_k} < 0.8 \end{cases}$$

Alternatively, I can first "thresh-hold" the abundances given, converting to binary presence/absence data. To do this, I find the mean of all non-zero abundance values μ_{nz}, and then

$$r_{ij}^* = \begin{cases} 1 & r_{ij} \geq 0.01\mu_{nz} \\ 0 & r_{ij} < 0.01\mu_{nz} \end{cases}$$

The network is then constructed using sample correlation just as before.

Again, I check the significance of each edge, removing edges that have a high probability of occurring in a null model. I check this using Monte-Carlo simulation by estimating $p_{ik} \approx P(\rho_{\boldsymbol{r}_i^{null}\boldsymbol{r}_k^{null}} > \rho_{\boldsymbol{r}_i\boldsymbol{r}_k})$. I take enough Monte-Carlo trials for a 95% confidence interval of length 0.03. I then adjust edges so that

$$p_{ik} > 0.05 \Rightarrow w_{ik} = 0$$

2.2 Null Model

Rather than simply generate an Erdős-Rényi random graph with some chosen parameter, I generate a graph that more closely reflects the idea of random data. That is, I start with random data and then generate a graph. I generate random data in the following way. Let

$$N_j = |\{i : r_{ij} \neq 0\}|$$

and

$$P_i = \frac{|\{j : r_{ij} \neq 0\}|}{|\{(i,j) : r_{ij} \neq 0\}|}$$

then I take

$$\hat{r}_{ij} = binom(N_j, P_i)$$

as randomly generated abundance data. The values N_i are the counts of appearances of taxi i, while the values P_j are the proportions of all appearances which happen within each sample.

Then, I construct our random graph from this random data. If $\hat{\boldsymbol{r}}_i$ is the vector of random "abundances" of taxa i, I have edges

$$w_{ik}^{null} = \frac{1}{N} \frac{(\hat{\boldsymbol{r}}_{\boldsymbol{i}} - \hat{\mu}_i \mathbf{1}) \cdot (\hat{\boldsymbol{r}}_k - \hat{\mu}_k \mathbf{1})}{\hat{\sigma}_i \hat{\sigma}_k}$$

The null model's use of a binomial random variable can be interpreted in the following way. A "success" for a taxa/sample pair is an appearance of the taxa in the sample. The number of trials is the number of times the taxa appears in the data set. The probability of success is the proportion of all appearances which happen within the sample. This does have some drawbacks: notably that it does not preserve total abundance, or even total appearances. It allows more than one "success" for a single taxa/sample pair, which could be interpreted as higher abundance. Our "random samples" should be scaled by average abundances of a taxa across samples to "look like" real data. This doesn't effect correlation, because $Cov(ax, by) = abCov(x, y)$, and $\sigma_x = Cov(x, x)^{1/2}$, implying that $w_{ik} = Cor(\hat{\boldsymbol{r}}_i, \hat{\boldsymbol{r}}_k) = Cor(a\hat{\boldsymbol{r}}_i, b\hat{\boldsymbol{r}}_k)$. Notice also that the null model is the same as in the binning procedure, with a different construction of a correlation graph.

In [8], this model is called the "binomial model". Interestingly, that paper argues that an "edge swapping" model is more accurate. A random permutation of the data also used in [4]. However, I have chosen the binomial model for speed.

2.3 Significance of the network

The networks connected by the methods above would not occur in randomized data, and so there is some significance to these networks. I generate random data as in the correlation construction of the network. I then use Monte-Carlo simulation to estimate the following

- $P(\mu_d < \mu_d^{null})$, where μ_d is the mean degree of the nodes of a network
- $P(\sigma_d < \sigma_d^{null})$, where σ_d is the variance of degrees of the nodes of a network
- $P(\lambda_{max} < \lambda_{max}^{null})$, where λ_{max} is the largest eigenvalue of the adjacency matrix
- $P(\lambda_{min} < \lambda_{min}^{null})$, where λ_{min} is the smallest eigenvalue of the adjacency matrix
- $P(|\mathcal{E}| < |\mathcal{E}^{null}|)$

The result with $10,000$ Monte Carlo trials was that it was extremely rare for a network constructed from a null model to have higher mean degree, higher degree variance, larger maximum eigenvalue, smaller minimum eigenvalue, or more edges. This suggests that the networks constructed from real data would be very unexpected according to the null model.

In addition to testing the complete network, I performed the same analysis on networks built from randomly drawn subsets of the data.

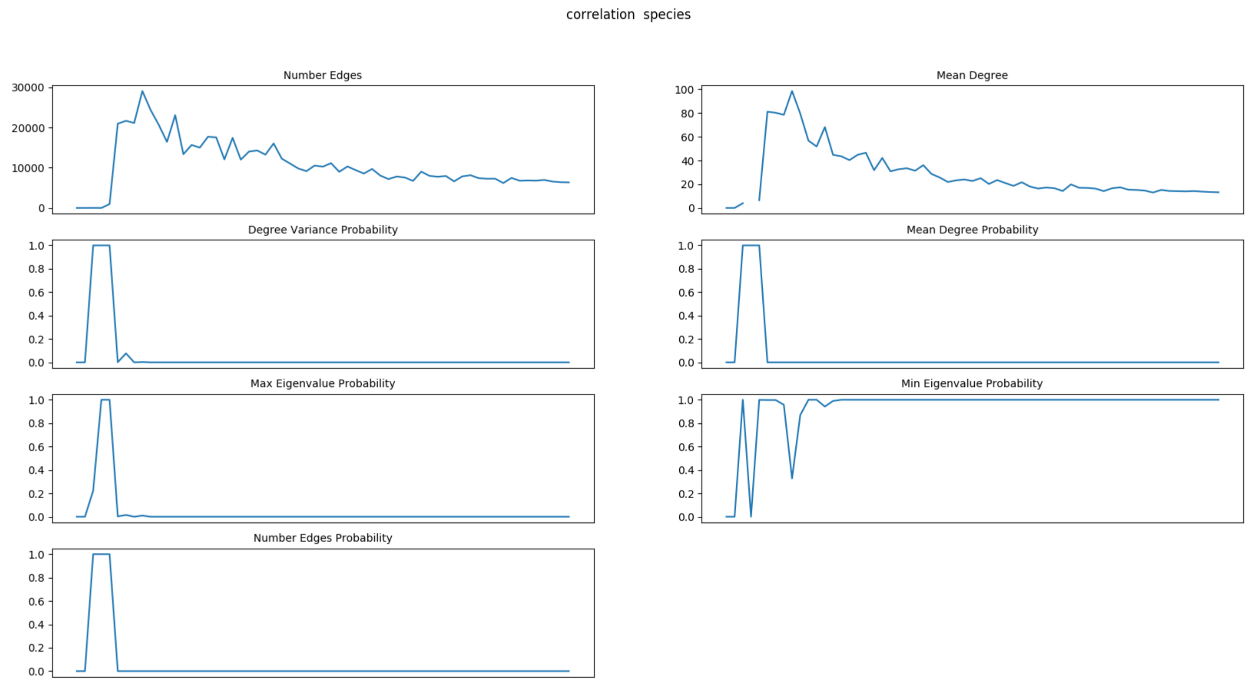


Figure 1: *The result of Monte-Carlo estimation of the likelihood of various network statistics, as well as the number of edges and mean degree. The networks were made from correlations using randomly drawn subsets of increasing size of the data at the species level, from 1 sample to 60 samples. Samples were GOTTCHA output of data from (FIND OUT). For each network, 1000 random trials were generated according to the above null model in order to estimate likelihood of statistics.*

3 Taxonomic Classification Analysis

We would like to identify taxa likely to be present in a sample, given an initial estimate of abundances of taxa. Inspired by spectral clustering, I solve the diffusion equation on the graph in order to identify nodes that are likely present, given the sample data. The diffusion equation is

$$\frac{\partial}{\partial t}u(v,t) = Lu(v,t)$$

where v takes values in the vertex set of the graph. Then, we can encode "known" information in three ways: initial values, boundary values, or a forcing vector.

Method 1 (Initial Value Problem)**.** *Let $u_i(t)$ be the solution at node v_i to the discrete diffusion problem*

$$\frac{d}{dt}\boldsymbol{u}(t) = -L\boldsymbol{u}$$

where L is the graph Laplacian with initial conditions $u_i = 1$ if node v_i is known to be "on", $u_j = 0$ if v_j is known to be "off", and $u_k = 0.5$ (or 0, or perhaps encoded with some confidence in $[0,1)$) if v_k is unknown. I then normalize the initial vector so that it represents a probability distribution on the nodes. Then, if $\boldsymbol{K}$ is the information "known" and the values of v_k and v_l are unknown,

$$\int_0^\infty u_k(t)dt - \int_0^\infty u_l(t)dt > 0 \Rightarrow P(v_k = 1|\boldsymbol{K}) > P(v_l = 1|\boldsymbol{K})$$

These comparisons are easily computed. Solutions to the diffusion equation are of the form

$$\boldsymbol{u} = \sum_{i=1}^{a} c_i \mathbf{1}|_{cc} + \sum_{i=a+1}^{n} c_i e^{-\lambda_i t}\boldsymbol{\xi}_i$$

where a is the number of connected components of the graph, and $\lambda_i, \boldsymbol{\xi}_i$ are eigenvalue, eigenvector pairs of L. Note that each eigenvalue $\lambda_i \geq 0$, with $\lambda_i > 0$ for $i = a+1, ..., n$ [11]. Then, assuming there is some initial mass on the connected component containing v_1 and v_2,

$$\int_0^\infty u_k(t)dt - \int_0^\infty u_l(t)dt = \sum_{i=a+1}^n c_i(\xi_{ki} - \xi_{li}) \int_0^\infty e^{-\lambda_i t}dt = \sum_{i=a+1}^n \frac{c_i}{\lambda_i}(\xi_{li} - \xi_{ki})$$

and $\boldsymbol{c} = V^{-1}\boldsymbol{u}(0)$. Therefore, it is straightforward to compute and compare the transitive terms of the solution:

$$\int_0^\infty \left(u_k(t) - \sum_{i=1}^a c_i \right) dt = \sum_{i=a+1}^n \frac{c_i}{\lambda_i}\xi_{ki} \tag{1}$$

We can relate this method to spectral clustering. Spectral clustering using computes a k-means clustering algorithm on the rows for a matrix of the first k eigenvectors ξ_k. Thus, spectral clustering uses a distance metric from the inner product

$$\|u_l\|_s = (\boldsymbol{V^T})_{\boldsymbol{l}}^T \begin{pmatrix} \boldsymbol{I}_{k\times} & 0 \\ 0 & \boldsymbol{0}_{n-k\times n-k} \end{pmatrix} (\boldsymbol{V^T})_{\boldsymbol{l}}$$

and clusters nodes l and m if $\|u_l - u_m\|_s$ is small, or

$$\langle u_m, u_l \rangle_{sc} = (\boldsymbol{V}_m^T)^T \begin{pmatrix} \boldsymbol{I}_{k\times} & 0 \\ 0 & \boldsymbol{0}_{n-k\times n-k} \end{pmatrix} \boldsymbol{V}_l^T \tag{2}$$

is large. Similarly, inspection of eq. (1) reveals that we will rank nodes highly when

$$\langle \boldsymbol{V}^{-1}\boldsymbol{u}(0), u_l \rangle_{ivp} = (\boldsymbol{V}^{-1}\boldsymbol{u}(0))^T \boldsymbol{\Lambda}^{-1}\boldsymbol{V}_l^T \tag{3}$$

is large. Notice that the ordering of the eigenvalues insures that the terms on the diagonal of $\boldsymbol{\Lambda}^{-1}$ decay along the diagonal, and all terms are positive, making this matrix and the matrix used in eq. (1) similar. We are in some sense then asking about how close a node is to the initial condition given in a metric similar to that used in spectral clustering. In fact, if we have as an initial condition $\boldsymbol{u}(0) = \boldsymbol{V}\boldsymbol{V}_m^T$, this metric is very close to that used in spectral clustering.

This method provides a ranking of taxa which attempts to estimate an ordering of the probabilities

$$\mathbb{P}(t_i \in S | m(S))$$

where $t_i \in S$ indicates taxa i is present in a sample, and $m(S)$ is the GOTTCHA output for the sample.

3.1 Matching samples to networks

Suppose we have different networks built from different types of data - i.e. healthy and unhealthy, or different regions, etc. We would like to match a sample with the network that makes most the sense. That is, we would like to say that a sample is more likely to be from one type of data than the other based on the network.

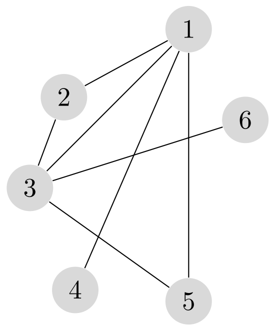
Clearly, the first pass is just asking how many taxa detected in the sample appear in each network. Next, we might ask how many clusters of a network are represented, or similarly how many taxa of the sample are in the same cluster in the network. The diffusion (method 1) idea can do something like this. We should see less nodes ranked highly that aren't in the sample if the sample "fits" a network well. If we imagine a sample is created by performing a random walk on a graph and recording the nodes most often visited, the diffusion idea should identify nodes that the random walk "should" have seen. If there are a bunch of those missing from the sample, we might think the network is not the right one. We could consider the map from rank to abundance, and integrate this against a kernel that weights towards the high ranks. As an example, let $\boldsymbol{s}$ be the sample abundances (so s_i is the abundance of organism i), and $\boldsymbol{r}$ the ranking (so r_i is the rank of organism i). We might take

$$F(\boldsymbol{s}, \boldsymbol{r}) = \sum_{i=1}^n c^{r_i/n} s_i$$

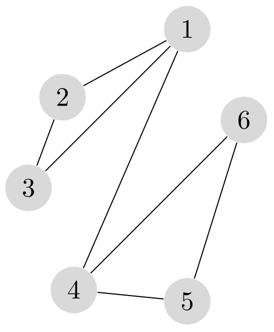
where $c < 1$. Given a sample, we get ranking from a network N_j. Let $\boldsymbol{r}^j(\boldsymbol{s})$ be the ranking given by diffusion on network j. Then,

$$F_j(\boldsymbol{s}) = F(\boldsymbol{s}, \boldsymbol{r}^j(\boldsymbol{s})) = \sum_{i=1}^n c^{r_i/n} s_i$$

and we can attempt to optimize over the networks we have (if there's only a few that's easy). The conjecture is then



(a) *Network A_1*



(b) *Network A_2*

Figure 2: *Test networks.*

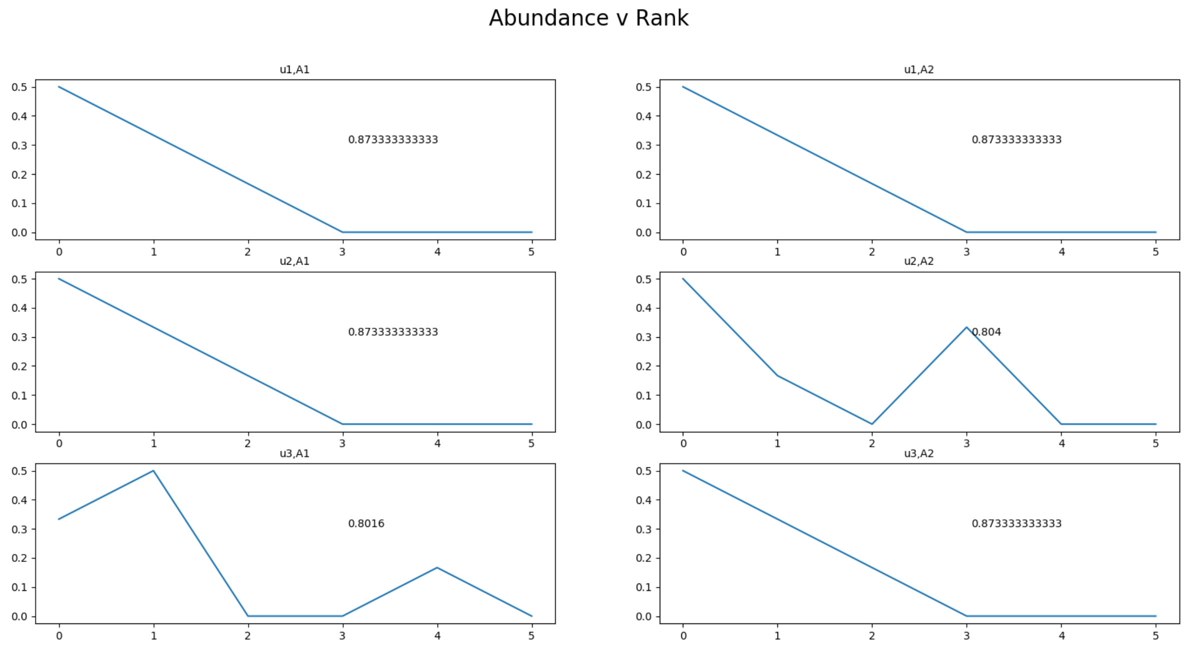


Figure 3: *Results of sample assignment: u_1 cannot be decided, u_2 is assigned to network A_1, and u_3 is assigned to network A_2.*

Conjecture 1. *Assume that $F_j(\boldsymbol{s}) > F_k(\boldsymbol{s})$. Then $P(\boldsymbol{s}|N_j) > P(\boldsymbol{s}|N_k)$.*

Let's start with a very simple example. Consider the two networks shown in fig. 2. Using "samples" $u_1 = (1/6, 1/2, 1/3, 0, 0, 0)$, $u_2 = (1/6, 1/2, 0, 0, 1/3, 0)$, and $u_3 = (0, 0, 0, 1/6, 1/2, 1/3)$ gave results shown in fig. 3.

I did the same on a column of training data for the species level network, a column of data that was held out of the network building, and a randomly generated sample. The results, shown in fig. 4, have the training data and held out data more likely to "fit" the network than the random data.

Formally, we have

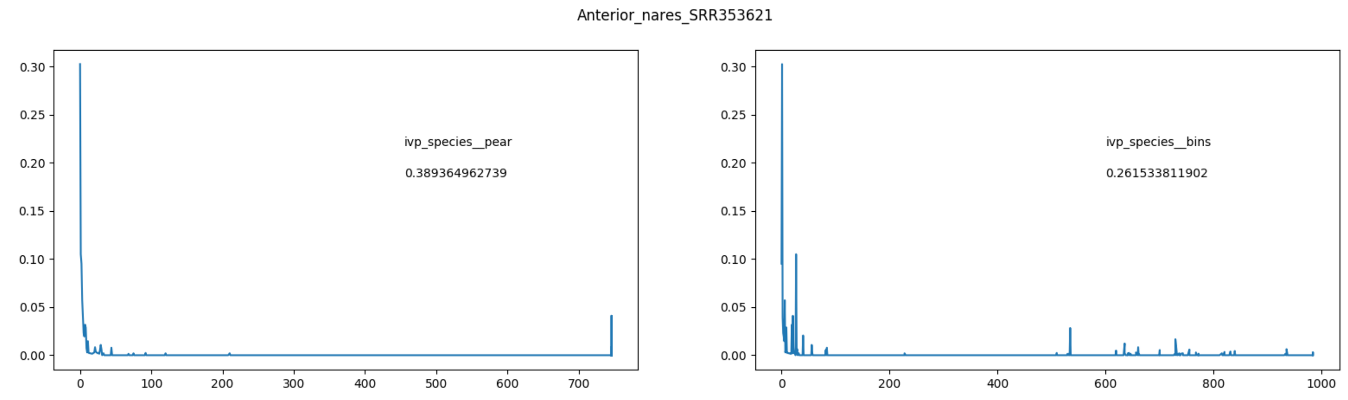
$$F_j(\boldsymbol{s}) = \frac{1}{\|\boldsymbol{s}\|}\sum_{i=0}^{n-1} c^{i/n} s_i$$

where $\boldsymbol{s} = (u_{j_1}(0), u_{j_2}(0), ..., u_{j_n}(0))$ such that if

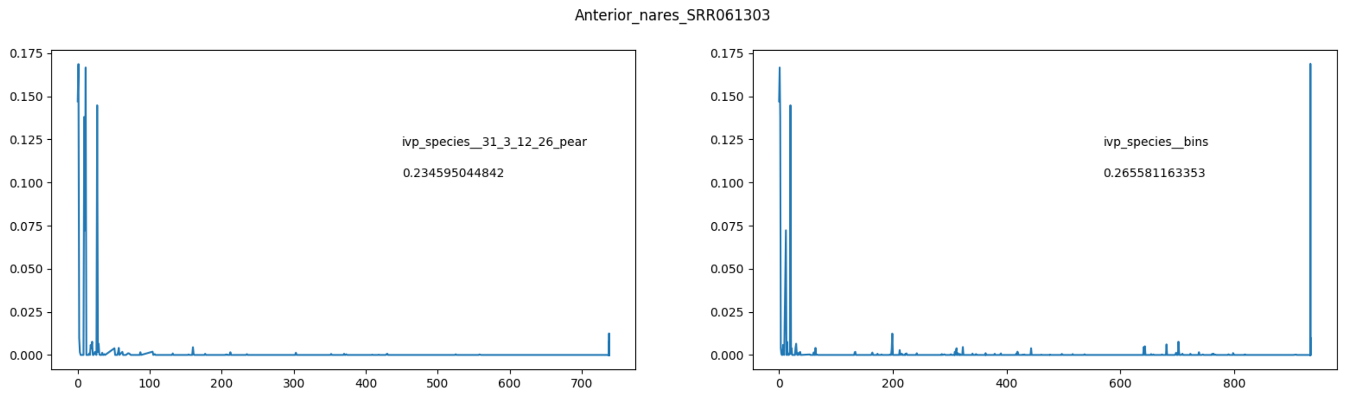
$$\frac{d}{dt}\boldsymbol{u} = -L\boldsymbol{u}$$

and $U_l = \int_0^\infty u_l(t)dt$ then

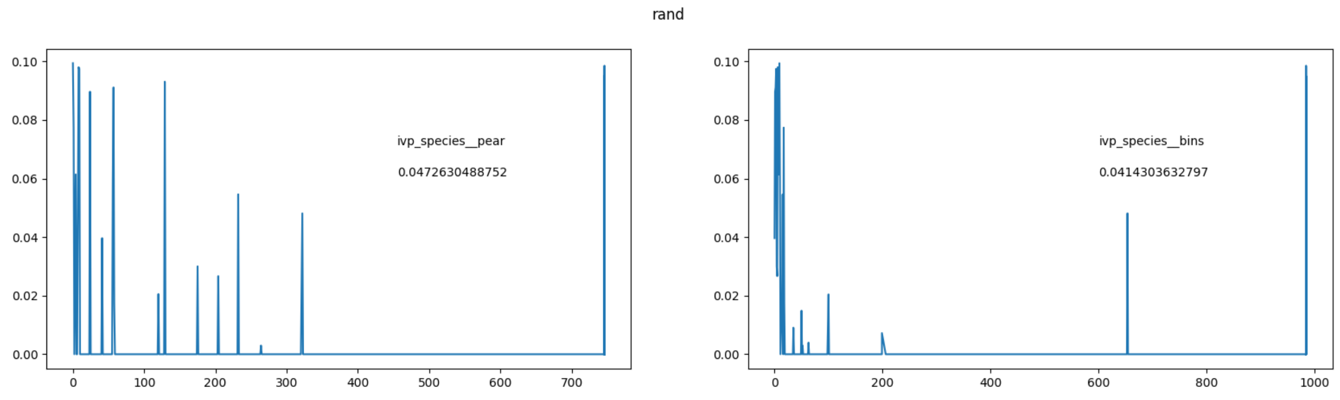
$$U_{j_1} \geq U_{j_2} \geq \cdots \geq U_{j_n}$$



(a) Assignment of column of training data to networks built with binning (right) and with correlation (left).



(b) Assignment of column of held out data to networks built with binning (right) and with correlation (left).



(c) Assignment randomly generated data to networks built with binning (right) and with correlation (left). This random sample had 63 nodes chosen to be present, uniformly, and given an abundance drawn from a $unif[0, 0.1]$ distribution.

Figure 4: *A training data column fits the network better than a randomly generated sample.*

4 Validation

To validate this approach, we build correlation and thresholded correlation networks from GOTTCHA output of 1,367 samples from the NIH Human Microbiome Project (`08_14_14_57_networks`). We constructed a network using all of these samples, as well as networks for 6 body sites for which at least 50 samples were present in the full 1,367:

- Anterior Nares (168 samples)
- Buccal Mucosa (206 samples)
- Posterior Fornix (110 samples)
- Stool (288 samples)
- Supragingival Plaque (220 samples)
- Tongue Dorsum (223 samples)

For each network we assessed the significance of the statistics listed in section 2.3. (`stats.txt` in the folder with the networks.)

We analyzed 300 samples from the NIH Human Microbiome Project that were not used in the network training. We computed the fit score of each sample on a the networks made from the set of 1,367 samples, as well as the body site networks. We hoped to analyze the fit score's use in identifying the body site a sample was taken from. We compared fit of each sample across networks to identify the body site.

We also computed the fit on the full networks of simulated samples created using the null model describe in section 2.2.

For each sample, we also performed the following experiment. We set the taxa with n^{th} highest abundance in the sample to 0 abundance and used the diffusion method to find the rank of this taxa, for n from 1 to 100.

5 Discussion

References

[1] Aaron Clauset, M. E. J. Newman, and Cristopher Moore. Finding community structure in very large networks. *Phys. Rev. E*, 70:066111, Dec 2004.

[2] Karoline Faust and Jeroen Raes. Microbial interactions: from networks to models. *Nature Reviews Microbiology*, 10:538–550, 2012.

[3] Tracey Allen K. Freitas, Po-E Li, Matthew B. Scholz, and Patrick S.G. Chain. Accurate read-based metagenome characterization using a hierarchical suite of unique signatures. *Nucleic Acids Research*, 43(10), 2015.

[4] Lianshuo Li, Zicheng Wang, Peng He, Shining Ma, Jie Du, and Rui Jiang. Construction and analysis of functional networks in the gut microbiome of type 2 diabetes patients. *Genomics Proteomics Bioinformatics*, 14:314 – 324, 2016.

[5] Po-E Li, Chien-Chi Lo, Joseph J. Anderson, Karen W. Davenport, Kimberly A. Bishop-Lilly, Yan Xu, Sanaa Ahmed, Shihai Feng, Vishwesh P. Mokashi, and Patrick S.G. Chain. Enabling the democratization of the genomics revolution with a fully integrated web-based bioinformatics platform. *Nucleic Acids Research*, 45(1):67, 2017.

[6] Zhanshan (Sam) Ma, Chengchen Zhang, Qingpeng Zhang, Jie Li, Lianwei Li, Linyi Qi, and Xianghong Yang. A brief review on the ecological network analysis with applications in the emerging medical ecology. In Terry J. McGenity, Kenneth N. Timmis, and Balbina Nogales Fernndez, editors, *Hydrocarbon and Lipid Microbiology Protocols*. Springer, 2016.

[7] Kevin P. Murphy. *Machine Learning: A Probabilistic Perspective*. The MIT Press, 2012.

[8] Heiko Mller and Francesco Mancuso. Identification and analysis of co-occurrence networks with netcutter. *PLOS ONE*, 3(9):1–16, 09 2008.

[9] M. E. J. Newman. Analysis of weighted networks. *Phys. Rev. E*, 70:056131, Nov 2004.

[10] Paul Shannon, Andrew Markiel, Owen Ozier, Nitin S Baliga, Jonathan T Wang, Daniel Ramage, Nada Amin, Benno Schwikowski, and Trey Ideker. Cytoscape: A software environment for integrated models of biomolecular interaction networks. *Genome Research*, 13(11):2498–2504, 11 2003.

[11] Ulrike von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416, 2007.