University of Padova

Department of Department of Mathematics

Master Thesis in MSc. Data Science

Benchmarking Reverse Engineering Methods for Interaction Networks of Microbial Communities

Supervisor
Professor Di Camillo Barbara
University of Padova

Master Candidate
Nora Nikoloska

Academic Year

2021-2022

Abstract

In many ecosystems, from soil to ocean and lake water, including the human microbiome and saliva, there are hundreds of microbial species cohabiting the environment including bacteria, fungi, archaea and protozoa. They are connected in an intrinsic network of interactions defined by the shared resources and consumption dynamics. These types of interactions can be divided based on the type of influence one species has on another which can be positive, negative or neutral. By studying these interactions closely, an insight into the difference between healthy and diseased tissues can be gained which in turn could facilitate the development of suitable treatments for gut related diseases. With the advance of metagenomics sequencing techniques, the relative abundance of these species can be simultaneously experimentally recorded and studied. One of the steps in the analysis of the data obtained in this way is the attempt to construct the network of microbial interactions using a certain reverse engineering method. There are different methods available in the literature often based on correlation or mutual information. However, without knowing the original network their accuracy cannot be calculated. In order to be able to construct a ground truth network, the species abundance data used in this Thesis is simulated using the Community Simulator package. In addition, experimental noise is added to the data using the metaSPARSim simulator in order to obtain relative abundances that are as close as possible to the experimental data. This allows for a network comparison between the real and method-obtained interactions over multiple metrics as a comparison study. In addition, statistical results on the obtained data distribution are reported and compared with publicly available microbial data.

Contents

Listing of figures

Listing of tables

Listing of acronyms

GED	Graph Edit Distance
DNA	Deoxyribonucleic acid
RNA	Ribonucleic acid
PCR	Polymerase chain reaction
NGS	Next Generation Sequencing
OTU	Operational Taxonomic Unit
MicroCRM	Microbial Consumer Resource Model
GC	Giant Component
MGH	Multivariate Hypergeometric distribution
CDF	Cumulative Distribution Function
ROC	Reciever Operator Characteristic
TP	True Positives
FP	False Positives
TN	True Negatives
FN	False Negaives
TPR	True Postive Rate
FPR	False Positive Rate
AUC	Area Under the Curve
AUPR	Area Under the Precision-Recall curve
RMSD	Root Mean Square Deviation

1 Introduction

According to some predictions, the estimated amount of microbial species on Earth is 1 trillion (10^{12}) species, a value larger that the number of stars in the night sky [1]. The estimated number of all individual bacteria organisms, however, is 5×10^{30}. The scale of these numbers is similar when it comes to microbial populations in the human body. The human body is estimated to be a host to over 39 trillion microbial cells, which is more than the estimated number of human cells (30 trillion). Interestingly, the human body is highly selective of the types of microbial species which it cultivates a symbiotic relationship with. Gut microbiota in adults is dominated by members of only two divisions of bacteria—the Bacteroidetes and Firmicutes and one member of Archaea. Irregularities in the human gut microbiome are a known cause for a number of diseases such as obesity, IBS, Chron's disease, chronic skin inflammation and others. In addition, there exists evidence to link the gut micriobiome with neurological, cardiovascular and respiratory diseases [2]. Understanding how microbial communities interact and influence their environment is a crucial step in the process of developing treatment options for these diseases. These interactions cannot be recorded directly and can only be inferred from the species abundances using different methods.

The microbial interactions in general can be represented as networks, where nodes represent species and edges are added between interacting species with a weight proportional to the strength of the interaction. This thesis focuses on the methods of network construction for microbial interactions and their comparison, with the necessary steps of data simulation, processing and analysis. The text is organized as follows: the methods and technologies of data collection and processing for metagenomics studies are explained in Chapter 2. The main types of network topologies are outlined in Chapter 3 along with properties from graph theory and network science. The steps of data simulation from absolute and relative abundance of microbial species and the generation of a ground truth network is detailed in Chapter 4. Different implementations of methods for reverse engineering of the ground truth networks are tested in Chapter 5, with details for each of the methods and an overall comparison of the results. The code, the simulations and figures generated in this work are available on the following Google drive link.

2

Metagenomics and Microbial Data

> "We are in the midst of the fastest growing revolution in molecular biology, perhaps in all of life science, and it only seems to be accelerating"
>
> - J. C. Wooley, A. Godzik, and I. Friedberg, "A primer on metagenomics"[3]

The estimated quintillions (5×10^{30}) of prokaryotic cells that live on the Earth influence the health of ecosystems, food chains, environments and interacting species. The advancements in metagenomics sequencing technologies have opened the possibility to explore the vast world of microbial life and to answer questions about the way they influence the world. This chapter introduces the definition of genetics and the analysis techniques of sequence data. It starts with a section of metagenomics and its development. It follows by focusing on 16S rRNA count or 16S DNA-seq data and its properties which are the type of data used in this work. Finally, types of microbial interactions are listed in Section 2.3 in order to explain the expected feature relationships in detail.

2.1 Metagenomics

Metagenomics is the term used for structural and functional analysis of the genetic material (nucleotide sequences) obtained by bulk sequencing of all of the organisms present in a sample.

The organisms are typically microbial consisting of viruses, bacteria, fungi, archaea or protistis. There are diverse types for the samples of the material too. Environmental samples range form water (oceans and lakes) to soil environments. Human samples are typically taken from the part of the body where the complex interaction with the mirobiota happens subject to the immunoregulation processes of the host. Typical sites for sample collection include: oral cavity, tongue and throat, vagina and cervix, stool and the intestinal gut along the *mucosal firewall*. These samples often include a large number of different species of microbes ranging from the hundreds to the thousands, a number lower than the typical number of collected samples in a given study which ranges from the tens to the hundreds. The number of samples required for a given study can be estimated by plotting a *refraction curve*. This line shows the number of single organism sequences against the number of sequences in a sample. When the slope of the line decreases and an almost constant amount of sequences of the organism are found despite increasing the total number of sequences, the minimal sample size is reached.

The next step in identifying which species are present in the samples is the sequencing process - identifying the unique sequence of nucleotides from which the DNA molecule of each organism is composed of. Each nucleotide contains one of the following nitrogen bases: Adenine (A), Cytosine (C), Guanine (G) and Thymine (T) and their specific sequence carries the genetic information of the species. The unit of measure of the length of a given sequence is *base pairs (bp)* or the number of nucleotides. There are a number of sequencing techniques developed since the first bacterial genome *(Haemophilus influenza)* was sequenced in 1995. A great advantage is that these methods that do not require lab cultured colonies in order to identify the species. However, since the methods read length (number of nucleotides per fragment) ranges from 20 to 700bp, fragmented sequences need to be assembled. In metagenomics sequencing, because the fragments originate from different organisms, computational challenges arise in the assembly phase.

2.1.1 First Generation Sequencing

All sequencing methods include library preparation steps such as filtration, DNA extraction and a cloning technique. The first sequencing method for metagenomics data is environmental shot-gun sequencing (ESS). After fragmenting the isolated DNA, the technique clones the fragments into plasmid vectors of a growing colony in order to increase the number of fragments. The detected signals are then sequenced by the Sanger *chain-termination method* [4]. Since there are multiple organisms present in the sample, primers are used to sequencing regions of interest.

2.1.2 Next Generation Sequencing (NGS)

Newer sequencing methods were developed that overcame some of the challenges with the traditional Sanger sequencing [5]. They make cost for sequencing is significantly lower with higher throughput, simplifying the library preparation step and removing the cloning process. This opened the door in exploring diverse species, since only it has been reported that <2% of bacteria can be cultured in a lab and are compatible with the bacterial vector.

The first sequencing technology is Roche that uses emulsion PCR (ePCR) for sample amplification while the DNA fragments are fixed on micro beads. During the sequencing step, in each iteration nucleotides of one base are released into the plate and each binding of the base with its complementary pair on the template results in a release of pyrophosphate. This is also referred to as *pyrosequencing*.

The second technique is created by Illumina. The template fragments are placed on a flow cell and folded in a bridge shape with the appropriate adaptors. The bridge amplification step creates clusters of copies around each fragment. The sequencing step allows for all bases to be present in the flow cell, but the attachment of bases is regulated to one per iteration. Each base has a unique color emission which is detected in the sequence.

Finally, the Ion Torrent technique uses the same ePCR methodology for the amplification step and similarly, one base is released in the cell at a time. However, it exploits the fact that a hydrogen ion is released with every attachment of a nucleotide to the template. During the sequencing step it detects the change of pH produced by the reaction and transmits the signal.

The third generation sequencing are technologies that do not require the fragments amplification step. Even though the NGS technologies have numerous advantages, they come with their own set of challenges such as poor quality reads or adapter interference. There are different pipelines available for the downstream analysis of the data depending on the type of study. It is important to note that agreed on standards that ensure the quality of the data should be respected in every step of the process, from sample collection to metadata annotation.

2.2 16S rRNA data

Metagenomics samples contain the DNA of all organisms present in the sample, including viruses. It is useful to filter out the viruses out of the sample and analyse the the remaining prokaryotes. Whole genome sequencing is inefficient for metagenomics data, and the read limit of most sequencing technologies is around 800bp. For these reasons, a shorter sequence present in all self-replicating organisms which can be used for distinct species identification is needed. The sequence most often used in studies is 16S rRNA which codes for the 30S small ribosomal subunit. The regions V1-V19 of this gene can be used for identification of procaryotes in the sample. Depending on the type of study, different regions may be sequenced instead of the whole gene, such as regions V3-V5 for compositonal analysis or V8-V9 for clustering [6]. In this work the simulated data is designed to resemble a count matrix with samples on columns and species on rows. Species are often referred to operational taxonomic units (OTUs) or taxa which are defined as a group of closely related individuals. All terms are used interchangeably in this text. The observed species\OTU abundance in a given sample is shown as an entry in the count matrix. Often this matrix is normalized by rows, and species abundances are fractions with the constraint of the constant sum of 1 in each sample. This transformation makes the data compostional and specific transformations are applied to the data in order to reduce the compositional effects, later explained in Chapter 5.

2.3 Microbial interactions

Microbes live in complex environments where they constantly interact with hundreds of different species. There are there are several types of inter-specific interactions. They differ between the influence one species has to another which can be positive, negative or neutral. As outlined in [7], there are 7 types of microbial interactions:

Positive Interactions:

1. Mutualism (+ +)
   This is is a positive interaction which benefits the co-existence of both species and more often refers to individual microbes more than groups.
2. Photocooperation or Synergism (+ +)
   This interaction is equally positive for both species as mutualism, but it is not necessary for the survival of any of the species.
3. Commensialism (+ /)
   Commensialism is a positive interaction which results in benefit for one of the interacting species, while for the other it has no effect.

Negative Interactions

1. Predation (+ −)
   Predation is an interaction where one species (the predator) hunts down and consumes another (the prey).

2. Parasitism (+ −)
   Parasitism is an interaction where one of the species benefits over the other. It is positive for the benefiting party (the parasite) and negative for the host.

3. Ammensalism (− /)
   This is an interaction where a given species is negatively impacted by another which remains unaffected by the interaction.

4. Competition (− −)
   When two species are competing for a common resource crucial for survival the interaction is impacting them both negatively.

Symbiotic interactions are of special interest. They are defined as a close and long-term interactions between different species and can be mutualistic, commensalistic or parasitic. The large difference in the genomes of the interacting species makes the species identification task easier during the assembly faze. In addition, the study of the resources exchanged between long-term interacting species can give insight into their overall function.

3
Network topology

This chapter focuses on introducing the necessary network theory concepts for the analysis of a given model of microbial interactions. The first section 3.1 introduces the graph object, adjacency and incidence matrices and their connection to networks. The second section 3.2 introduces the most commonly used network properties in the process of network comparison such as the degree distribution, clustering coefficient and centrality measures. The third section focuses 3.3 on the scale-free property as one of the most important ones found in different types of network including both biological and metabolic networks for species interaction. The graph theory definitions are taken from *A textbook of graph theory* [8] and the network science properties definitions closely follow the ones defined in the *Network Science* book by A. Barabási, and M. Pósfai [9].

3.1 Graph Theory

The definition of a graph object is given as a triple comprised of the set of vertices, the set of edges and their relation:

Definition 1. *(Graph)*
A graph is an ordered triple $G = (V(G), E(G), I_G)$, where $V(G)$ is a nonempty set, $E(G)$ is a set disjoint from $V(G)$ and I_G is an "incidence" relation that associates with each element of $E(G)$ an unordered pair of elements (same or distinct) of $V(G)$. Elements of $V(G)$ are called

vertices (or nodes or points) of G, while elements of $E(G)$ are called edges (or lines) of G. $V(G)$ and $E(G)$ are named vertex set and edge set of G, respectively.

The incidence function I_G is a mapping between the set of edges and the set of vertices producing a pair of vertices associated with a given edge. For example: If $I_G(e) = \{u, v\}$ then the vertices u and v are called the end vertices or ends of the edge e. [8].

Definition 2. *(Incidence matrix)*
Incidence matrix is a logical matrix that shows the relationship between two classes of objects. The graph incidence matrix, however is best pictured as a matrix with the set of vertices $V(G)$ on rows and the set of edges $E(G)$ on columns, where the existing connections are represented by ones (1) in the corresponding cell.

When representing the graph in a graph diagram, the commonly used matrix is the square *adjacency matrix* which is composed of the set of vertices $V(G)$ both on rows and columns where values in cells are one (1) if there exists and edge between the corresponding vertices. The *self-loops*: edges with the same starting and ending vertex are shown on the diagonal.

Each edge in the graph can be *directed* from a chosen starting to an ending vertex, annotated with an arrow in the graph diagram and a 1 in the cell between the starting vertex (on rows) to the ending vertex (on columns) in the adjacency matrix. Alternatively, a given edge can be *undirected* which semantically depicts a connection between the vertices, without a starting and an ending vertex. Graphs comprised of undirected vertices are *undirected graphs* and graphs comprised of directed vertices are *undirected graphs*. In the adjacency matrix an undirected edge is inputed twice for the both variations of the vertices. Thus, the adjacency matrix for undirected graphs is symmetric with respect to the main diagonal. In some cases, graphs can be both directed and undirected. Such is the example of metabolic graphs where the edges represent reactions which can be reversible (represented by undirected edges) and in the same graph some are irreversible (represented with a directed edge).

Graphs can also be *weighted* if the there is a weight associated with each edge. In this case, in the adjacency matrix, instead of ones that represent the presence of an edge, a weight value is given for each edge.

If the graph vertices and edges are unlabeled, the same graph can have more than one form. Any graph that has the same number of edges and vertices and the same edge connectivity is said to be *isomorphic* to the original. One metric that is used to assess directed and unlabeled graph similarity is the Graph Edit Distance (GED). This is a metric calculated between two graphs defined as the minimum number of elementary graph operations needed to get to the first graph to the second. These operations can be: insertion, deletion or substitution of vertices and edges The equation is given as:

$$GED_{(g_1,g_2)} = min_{(e_1,\ldots,e_k)\in\mathbf{P}_{(g_1,g_2)}} \sum_{i=1}^{k} c(e_i) \tag{3.1}$$

where $c(e_i)$ is the cost of each graph operation e in the set of edit paths $\mathbf{P}$ from graph $g1$ to $g2$. The computation of this metrics is slower for large graphs and in that case optimized algorithm can be used.

The terms *graph* and *network* are usually used as synonymous. The nuanced difference between the two terms is that networks refer to the real systems that are being analyzed, where graph is the preferred term for the mathematical definition of the main properties [9]. After introducing the necessary graph theory objects, this text will continue with the interchangeable usage of these terms ([graph, vertex, edge] and [network, node, link]).

3.2 Network properties

This section focuses on defining the basic network properties that are used to compare the different topologies. One of the main network properties that characterize a network is the *degree distribution*. For each node in the network, the *node degree* is the number of edges that a given node has. The degree distribution therefore gives the probability p_k that a randomly selected node has degree k. An important property is the *average node degree* across all nodes. The *total node degree* is the sum of all node degrees. For a undirected network this number is halved, accounting for the fact that each link is counted twice. When the network edges are directed, the node degree is a sum between the *in degree* and *out degree* of the node which refers to the sum of edges pointing to and from the node.

An important measure for quantifying the distances between nodes is the *shortest path length*. It is the minimum number of steps needed to get from node i to node j. The *average shortest path length* of a network gives information about the overall distance between the nodes. In a fully connected network, this value is always 1.0. The maximum shortest path length in a network is called the *diameter* of a network.

Another metric for that gives information about the connectedness of the network is the *clustering coefficient* C. It shows the level of linkage between the neighbourhoods of a given node i and it is defined as:

$$C_i = \frac{2L_i}{k_i(k_i - 1)} \tag{3.2}$$

where L_i is the number of edges of the k_i neighbours of node i. More useful metric for network characterization is the *average clustering coefficient* across all nodes.

3.3 Scale-free networks

Many real world network such as the WWW, the citation network and biological network including protein and microbial interactions are not random networks and share the scale-free property. A scale-free network is a network whose degree distribution follows a power law, and the probability that a given node has degree k is given by:

$$\begin{gathered} p_k = Ck^{-\gamma} \\ \sum_{k=1}^{\infty} p_k = 1 \end{gathered} \tag{3.3}$$

where C is the normalization constant and γ is the degree exponent. When the degree exponent γ is in the interval (2, 3) the network has the ultra-small world property.

The main property of scale-free network is that there are a low number of nodes with a high degree called *hubs*, whereas the majority of nodes have low degree. Another property of real networks and in particular of 16S rRNA count matrices is that they are sparse. In order to test different scenarios, data with four different network topolgies is simulated in Chapter 4 including fully connected, sparse, scale-free and the randomized topology.

4
Data Simulations

In order to conduct a comparative analysis on the different methods for reverse engineering of a microbial interaction network, simulated data is used in this work. This allows for the possibility of having a ground truth network with which the obtained results are compared in the next chapter. In this one, the detailed process of obtaining the data is explained and three different types of networks (fully connected, sparse, scale free and randomized) used for testing are compared.

Section 4.1 explains the first simulation of absolute abundances and the ground truth adjacency matrix. The obtained network topologies are compared in Section 4.2. Then, the process of simulating the relative abundances with experimental noise is explained in Section 4.3. Finally, the obtained distributions are compared with available microbial interaction data to confirm the similarity.

4.1 Absolute abundances simulation

Firstly, the absolute abundances of taxa and resources are simulated using the Community Simulator implemented in Python [10]. The simulator is designed to mimic the steps of a batch culture experiment for growing microbial communities. The growth of the bacteria is simulated over a given time T, over a defined number of wells. At the end, the absolute abundance of taxa and resources can be recorded.

The dynamics of the system are based on the Microbial Consumer Resource Model (MicroCRM). It is defined by three energy fluxes: J^{in}, J^{out}, and J^{grow} that each represent the energy flux entering the cell, the flux leaving the cell and the energy used for cell growth, respectively. Their relationship must satisfy $J^{in} = J^{out} + J^{growth}$ with the addition of a leakage fraction of $J^{out} = l \cdot J^{in}$. The full model which is used for the simulations is presented with the equations 4.1 and 4.2 for the consumer and resource dynamics, respectively.

$$\frac{dN_i}{dt} = g_i N_i \left[\sum_{\alpha} (1 - l_\alpha w_\alpha u^{in}_{i\alpha} \sigma(c_{i\alpha} R_\alpha) - m_i \right] \tag{4.1}$$

$$\frac{dR_\alpha}{dt} = h_\alpha(R_\alpha) - \sum_{j} N_j u^{in}_{j\alpha} \sigma(c_{j\alpha} R_\alpha) + \sum_{j\beta} N_j u^{in}_{j\beta} \sigma(c_{j\beta} R_\beta) \left[l_\beta D_{\alpha\beta} \frac{w_{beta}}{w_\alpha} \right] \tag{4.2}$$

As shown in equation 4.1, the dynamics for a given species N referenced with indices i and j depends on: the species conversion factor from energy to growth rate g and for each resource: the leakage fraction l, the energy content w, the metabolic regulation u, the response function σ as well as the minimal energy uptake for simply maintenance of the species m.

In the same time, the the resource dynamics defined in equation 4.2 depends also on the choice of a replenishment mode h for the given resource. A given resource R is referenced with indices α and β. The total number of consumers is defined as S and the total number of resources is M.

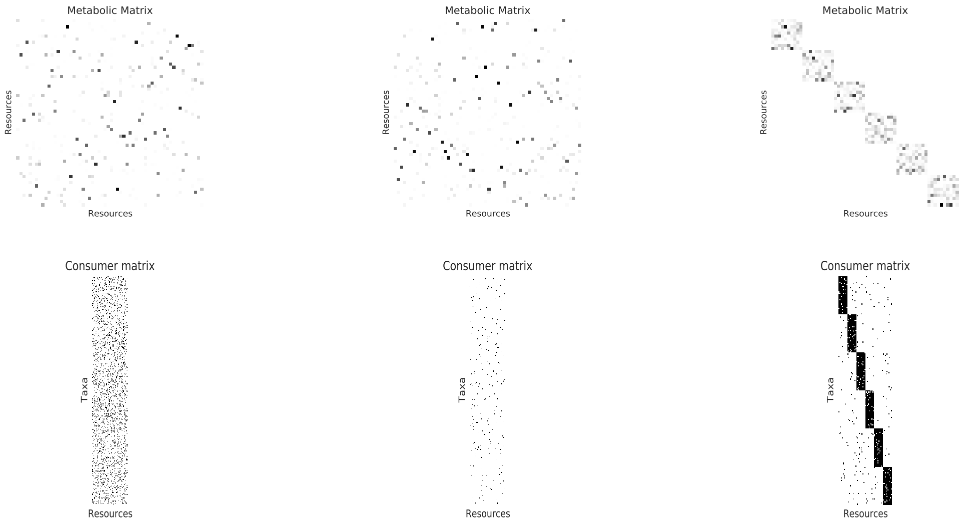
In addition to these user defined parameters, the values for $c_{i\alpha}$ and $D_{\alpha\beta}$ are taken from the consumer and metabolic matrices. The consumer matrix $c : (S \times M)$ defines the uptake rate of a given resource α by the species i. The metabolic matrix $D : (M \times M)$ defines the fraction of byproducts converted from resource β to resource α. These conversions are divided in three types: conversion of a resource to a waste class of unusable byproducts such as carboxylic acids, conversion between resources of the same type (sugars into alcohols), and others. Following additional user defined parameters, these matrices are sampled from a random distribution. For the consumer matrix that is a choice between Bernoulli, Gamma and Gaussian distribution whereas a Dirichlet distribution is used for the metabolic matrix in order to ensure that the sum of columns amounts to one. The choice of the parameters when creating these matrices is what influences the interaction network topology of the simulation. In this work, the parameters were selected in order to create three different network topologies: fully connected, sparse and scale free. The exact parameters used in each simulation are given in table A.1 of the Appendix.

The different consumer and metabolic matrices obtained in each case are shown on Figure 4.1. The parameters that influence the topology the most are:

1. fs and fw: The fractions of resource conversion of to the same (fs) resource class and to the waste (fw) class. Their sum is smaller than one with the remainder converted into other resources. The tuning of the fw parameter to a value close to zero and fs close to one allows for a resulting sparse topology since the species aren't connected through the waste class.

2. Specialised families: The species can either be a part of a specialized family that shares the same set of resources, or a generalist species which consumes a random subset of all resources. By creating 6 specialised families and 20 generalist species for the scale free network, the generalist species become the hubs of the network, while the rest of the species have a lower number of connections. Each specialized family is constrained to consume mostly one the resources in of the 6 resource groups.

3. Muc (Mean sum of consumption rates): This parameter controls the amount of consumption of a given species byproducts by another. By lowering the amount total consumption, sparsity is enforced on the resulting network and the values in the consumer matrix are overall lower in the sparse network case as seen on Figure 4.1 b).



(a) Fully connected network **(b)** Sparse network **(c)** Scale free network

Figure 4.1: Metabolic and consumer matrices comparison

For each simulation, the number of species (taxa) is 320 and the number of resources is 60. These are simulated in 60 independent conditions where one resource is removed per simulation. The initial vectors for the taxa abundance are sampled from a Gamma distribution of values available in the metaSPARSim R package [11]. This package is a 16S rRNA count data simulator and vectors of paired intensity and variability values for microbial species are available in several presets. These vectors are cleaned of missing data and the initial abundances for the simulations are distributed:

$$m_{ij} \sim Gamma\left(\frac{1}{\phi_{ij}}, \phi_{ij} \cdot \mu_{ij}\right) \tag{4.3}$$

where

- m_{ij} is the abundance of species i in the sample j. For the simulations only one sample is created per condition in a single experimental well.
- μ_{ij} is the mean abundance of the species i in its group. Since every condition contains only one group, this is the intensity level of the species i in the sample j.
- ϕ_{ij} refers to the biological variability of the species i in the sample.

The initial resource abundance is set to 5000 for all resources with an external supply type.

One initial simulation of the system with all of the resources present is ran in order to bring the system to a steady state. The initial plate is propagated for 500 time-steps and the taxa and resources abundances are recorded after each step. After this, 60 different conditions are simulating using as initial state the state obtained after the first simulation. Each condition corresponds to one resource having an initial abundance of zero and not being externally resupplied. The second simulation done for every condition has a duration of 50 time-steps.

The change of abundance during the first and second simulation is shown on Figure A.1 in the Appendix.

4.1.1 Randomized network simulation

In addition to these three network topologies, a random network topology has been simulated with a probabilistic parameterization of the metabolic matrix. This matrix determines the number of interactions in which a product is involved in as a product (by rows) or as a substrate (by columns). The values in the columns are set in a way to differentiate between the probability that the resources of the same resource class are the products when interacting with this resource more often than resources of all other classes. With every consumption of a resource, at least one other resource from the same class is produced as well as at least one waste class resource. The entries in each column for a resource i to be produced by resource j are sampled with a probability proportional to:

$$P_{ij} = \frac{1}{(\sum_0^{j-1} d_{ij} \neq 0 + 1)^e} \tag{4.4}$$

where the parameter e controls the sparsity.

In a similar way every row of the consumer matrix has been sampled, where a given species i is consuming resource j with a probability proportional to:

$$P_{ij} = \frac{1}{(\sum_0^{i-1} c_{ij} \neq 0 + 1)^e} \tag{4.5}$$

where a higher value for the parameter e increases the inter-species competition, and with that the number of negative interactions.

This approach helps to control the probabilities that a given resource is consumed and produced. When a given resource that is consumed by many species is produced by a few, the species that produce it are hub nodes in the network. On the other hand, when a resource is produced by many species but only consumed by few, the resource behaves like a waste resource. The degree distribution of the randomized network is stable among all of nodes with a small variation.

The absolute abundance for the randomized network is simulated for 60 species and 50 resources in 200 different conditions. These conditions are created in the following way: for 50 conditions the concentration of each resource is lowered and for additional 50 it is increased, with a final 100 conditions which are simulated by lowering the concentration of different resource pairs.

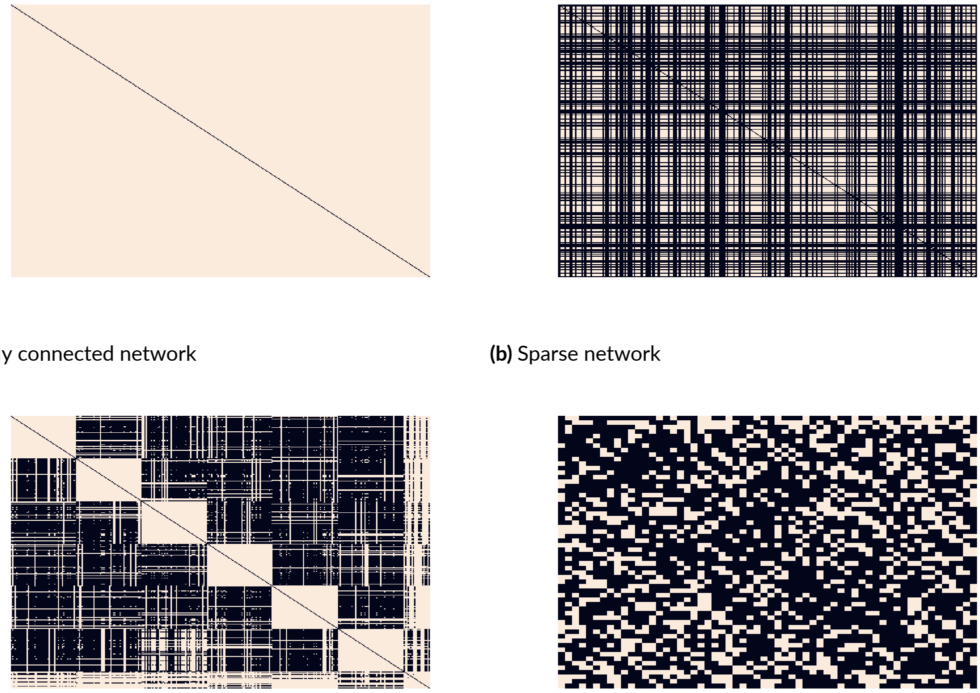
4.2 Network topology comparison

After conducting the same simulation steps with specific parameters for a fully connected, sparse, scale free and randomized topology the positive and negative interaction matrices are computed. The negative interactions matrix is created from the energy flux of the energy entering the cell J^{in} which can be obtained from the resource abundance and parameter dictionary. Then, the weight of the negative interactions network is proportional to the number of common resources detected. From here, by using the leakage fraction and the metabolic matrix, the energy exiting the cell J^{out} can be computed. The positive interactions matrix is then proportional to the number of common resources found between two species in the flux matrix of the J^{out} energy. By summing the positive and negative weight matrices, the combined weight matrix is obtained. Finally, the adjacency matrix is a binary matrix which is zero where there is no interaction detected and one where there is either a positive or negative interaction. It is important to note that this procedure creates the same adjacency matrix irrespective of the choice of condition for the resource vector used to calculate the J^{in} energy. However, the choice of condition affects the weights of the combined interaction. In this work, the first condition is used to compute the weight and adjacency matrices. The three adjacency matrices computed in this way are shown on Figure 4.2. In addition, the positive and negative interaction matrices are pictured on Figure A.2 and A.3 in the Appendix.

As it can be seen from figure 4.2 in the fully connected case only the diagonal contains zero values which means that there is a connection between all edges except for self-loops. In the sparse case, there are a lot more pairs of species without a connection between them. In the scale free network, the connections are concentrated mostly within the 6 specialist families, with other random connections as well. The generalist species seem to be connected to the majority of the remaining nodes and represent the nodes with the highest number of connections. In the randomized case, the number of species is significantly lower and the connections between the species do not follow a determined pattern.

In order to convert the adjacency matrices into graphs and analyze them the Python package networkx was used. Table 4.1 shows some of the main network properties of these graphs. The sparse graph is an unconnected with a total of 136 sub-graphs that comprise of the fully connected Giant Component (GC) with 185 nodes and the remaining nodes are isolated. As the number of edges reduces from the fully connected to the sparse network, the average shortest path length remains one, but the diameter increases slightly. The clustering coefficient and the average degree both decrease with the lower number of edges. The scale free network has more edges than the sparse network with a longer average shortest path length and a relatively high value of the clustering coefficient due to the closely connected specialised families.

This section explores the properties of the scale free network. The degree distribution is shown on Figure 4.3 a) and in log-log scale on b). The downward trend of a heavy-tailed distribution shows a lower probability for a given degree to appear in the network as the degree is increasing. This is the main property of a scale free network. The degrees for which the frequency is zero were excluded from the visualisation. The curve fit to the data in Figure 4.3 a). is done with a total error of:



(a) Fully connected network

(b) Sparse network

(c) Scale free network

(d) Randomized network

Figure 4.2: Adjacency matrices comparison
Dark pixels equal to zero and light pixels to one

$$R = 0.5486$$

and the following values for the power law parameters:

$$p(x) = Cx^{2.24}$$

where the constant C is equal to 4647796. From the γ exponent resulting in a value 2.9 which is in the interval $(2, 3)$ we can conclude ultra-small world behaviour in the network.

On Figure 4.3 c). the degree distribution data is compared with a theoretical power law distribution in the log-log scale, using the power law Python package. The degrees until the 120th degree are trimmed since the power law behaviour is expected with larger degrees.

Another property of the scale free networks is that the clustering coefficient decreases with higher degrees. This behavior shows that nodes with a lower degree tend to be clustered to-

Parameter	Fully Connected	Sparse	Scale free	Randomized
Number of edges	102080	34040	41759	1193*
Strongly / Weakly connected	T / T	F / F	T / T	T / T
Average Shortest Path Length	1.0	1.0*	1.59	1.66
Radius	1	1*	1	2
Diameter	1	1*	2	3
Clustering Coefficient	1.0	0.58	0.81	0.41
Average Degree	638.0	212.75	260.99	39.76
Average In Degree	319.0	106.37	130.49	19.88
Average Out Degree	319.0	106.37	130.49	19.88
Density	1.0	0.33	0.41	0.34
Center / Periphery	320 / 320	185 / 185*	4 / 316	54 / 6

Table 4.1: Network properties comparison

gether more closely, whereas the hubs do not share a lot of connections. Figure 4.4 shows the clustering coefficient with respect to the node degree both in linear and log-log scale.

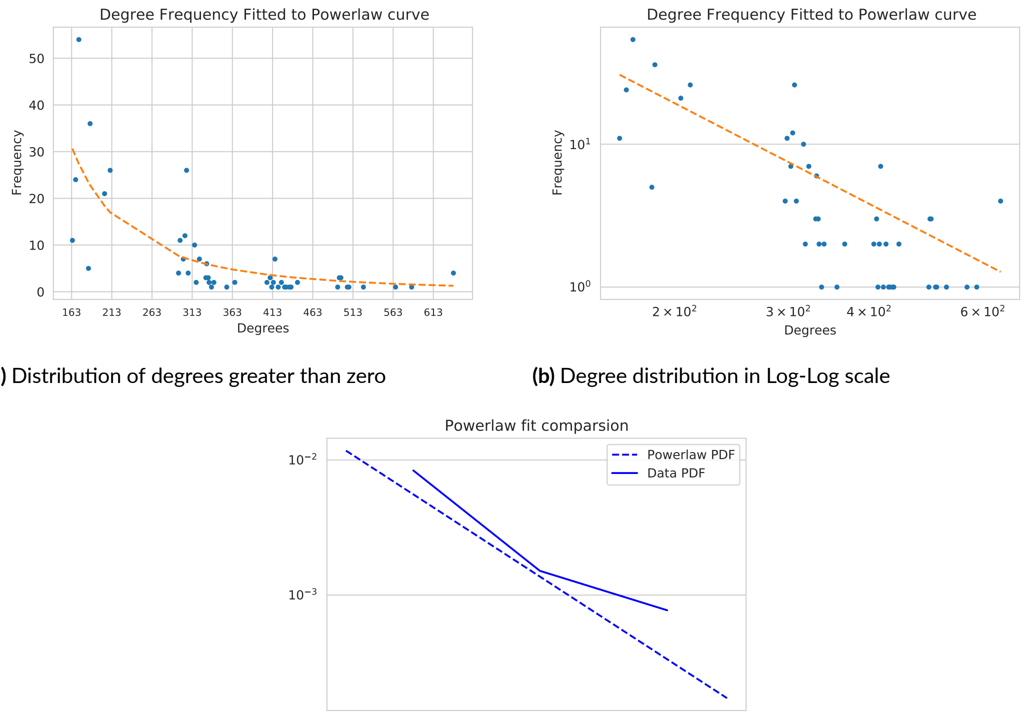
4.3 Relative abundances simulation

The species abundances obtained with the community simulator in Section 4.1 represent the amount of species truly present in the given condition and from them the ground truth interaction network is formed. Each of the simulations is a 320×60 (taxa × conditions) matrix. However, in order to properly benchmark methods for reverse engineering of the interactions, it is better to use data similar to the one normally detected with sequencing experiments. This is done by simulating relative abundances as a 16S rRNA count table using the metaSPARSim simulator [11]. Since the initial absolute abundances m_{ij} are simulated according to the Equation (4.3), the relative abundance Y_j is sampled from the following Mutivariate Hypergeometric (MGH) distribution:

$$Y_j \sim MGH(n_j, m_j) \tag{4.6}$$

where n_j is the library size of the sample or the total number of mapped reads.

For all simulations, the library size is set to 1000 for every species and a relative abundance vector is sampled for each condition.



(a) Distribution of degrees greater than zero

(b) Degree distribution in Log-Log scale

(c) Data fitted to power law distribution starting from the 120th degree onwards

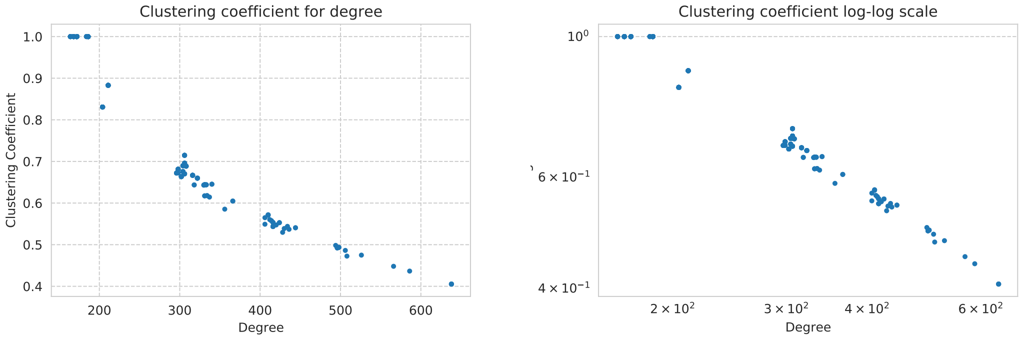
Figure 4.3: Scale free properties

4.4 Distribution comparison

Since sparisty is one of the main properties of 16S rRNA count data, in this section the distribution of relative abundances obtained are compared with the presets available in metaSPARSim. In order to compare the two distributions, a Kolmogorov Smirnov test is conducted [12]. The Kolmogorov Smirnov test is a non parametric test used for testing the similarity of a given sample with a known distribution in the one sample test or testing whether two samples belong to the same unknown distribution in the two sample test. The general formulation of the test hypothesis is:

$$\begin{aligned} H_0 &: F_1(t) = F_2(t) = \ldots = F_k(t) \; \forall t \in \mathbb{R} \\ H_1 &: F_i(t) \neq F_j(t) \text{ for at least one t} \in \mathbb{R} \text{ and } i \neq j \end{aligned} \tag{4.7}$$

The two-sided Kolmogorov-Smirnov test statistics is defined as follows:



(a) Clustering coefficient for degree

(b) Clustering coefficient distribution in Log-Log scale

Figure 4.4: Clustering coefficient in scale free network

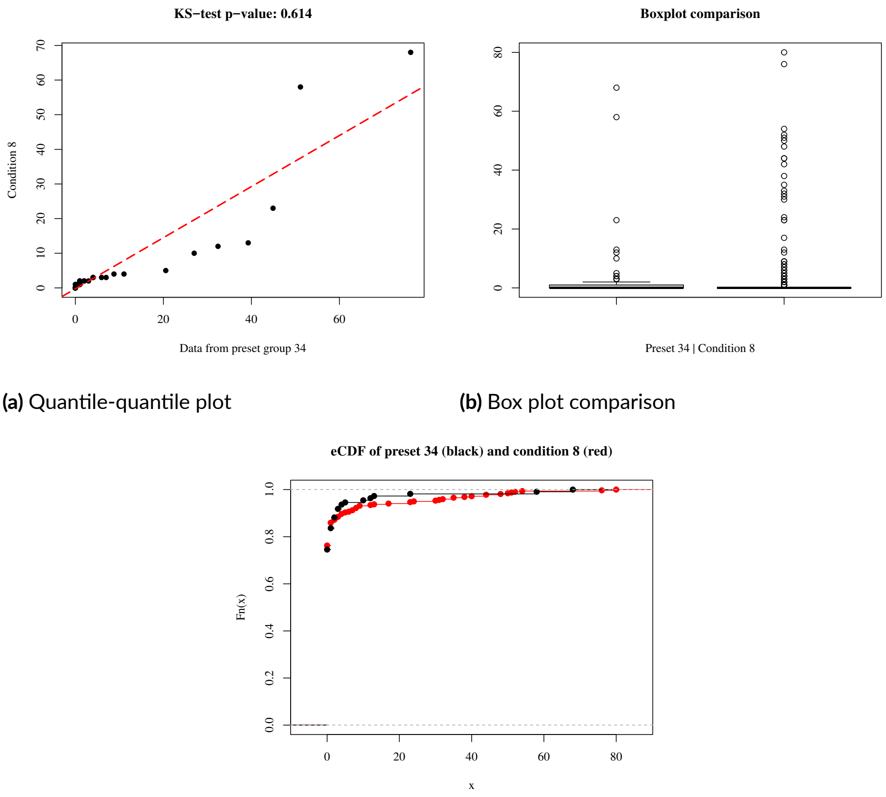
$$D_k = \max_{t,i,j} |\hat{F}_i(t) - \hat{F}_j(t)| \; i \neq j \; i,j \leq k \tag{4.8}$$

where $\hat{F}(t)$ is the empirical distribution function (eCDF) computed from the sample as:

$$\hat{F}_i(t) = \frac{1}{n} \cdot \text{[number of sample values lower than t]} \tag{4.9}$$

The value obtained with the Kolmogorov-Smirnov test statistics is the biggest difference in the empirical distributions between the samples. Even though this test is primarily for continuous distributions, the implementation in R supports discrete data as well [13]. The test is done on all possible pairs between the vectors of taxa for each condition and the abundances of the preset for each sample, with the number of conditions being 60 and 110 preset samples. The number of species between them is however different, with the number of simulated species being 320 while 3541 species are present in the preset.

The significance level is set to 0.05, and with higher values the alternative hypothesis is rejected, with the conclusion that the samples are most likely drawn from the same distribution. For each of the networks the following number of pairs with sufficient test significance are detected: 340 for the fully connected network, 245 for the sparse and 490 for the scale free. On Figure 4.5 plots of a pair from the scale free network simulation are shown to visualise the distribution comparison. The data is from the 34th sample in the preset and the 8th condition in the simulated data. The first plot is a quantile - quantile (Q-Q) plot which plots the quantiles of the given samples against each other. The red line is a fitted least squares regression line. The second plot shows a side by side box plot of the samples. It can be seen that even though



(a) Quantile-quantile plot

(b) Box plot comparison

(c) eCDF comparison

Figure 4.5: Distribution comparison between simulated data and preset

the preset sample is a lot bigger, most of the points are centered around the mean, whereas the simulated data shows a higher number of outliers. Since both samples are generally sparse, the mean values are close to each-other with 2.08 for the preset sample and 3.12 for the simulated condition. The third plot shows a comparison between the empirical probability functions of both samples used to compute the test statistic. The Kolmogorov-Smirnov test statistic can be sensitive, but it can be seen that the simulations generate sparse data in a similar range to the preset example.

5
Reverse Engineering Methods

The data simulated in the manner explained in Chapter 4 is now used for testing different reverse engineering methods for recovering the ground truth adjacency matrix and comparing the obtained results. This chapter begins with the analysis obtained from the concentration and correlation matrices of the simulation as a first insight into the original interactions. However, there are many statistical methods based on different approaches designed to specifically solve the problems of correlation inference of compositional data. Many of them are created for metagenomic studies where the data obtained by experiment is spare and the number of features is several magnitudes higher than the number of samples. The feature correlation itself and the concentration matrix are sensitive to these constraints, and methods have been developed for a more accurate estimation of the true interactions. The following sections of this chapter are devoted to explaining the techniques used in each of the algorithms. Their accuracy on the simulated data is compared in Chapter 6.

5.1 Concentration matrix comparison

In the attempt to detect the interacting species, the general approach is to isolate the pairs of species (features) whose abundance changes in a similar way across the conditions (samples). The covariance matrix of the data is a measure of joint variability between any pair of features. The shape of these matrices is $N \times N$ where N is the number of species with every off-diagonal entry N_{ij} depicting the covariance between species i and j and the species variance on the diagonal. On the other hand, the concentration matrix is the matrix inverse of the covariance matrix. This matrix is can be also called *precision matrix*. Both terms are used in the text interchangeably. The values in these matrices are zero if the corresponding feature are conditionally independent and the sign shows positive or negative dependence. The methods generally try to estimate one of these matrices as close as possible when inferring the interactions from the data. In some cases, the covariance matrix is singular - not all columns of the matrix are linearly independent and therefore cannot be inverted. In this case, the Moore-Penrose pseudo-inverse approximation is used. For a given matrix A, the pseudo-inverse A^+ can be calculated by:

$$A^+ = VD^+U^T$$

where U, V and D is the singular value decomposition of A and D^+ is the transpose of the matrix D with reciprocal of all non zero elements.

By using the concentration matrix for estimating the adjacency matrix of the given network, the task can be seen as binary classification. The accuracy is high if the edges of the ground truth network are detected and the concentration matrix value for unconnected nodes is zero. However, the matrix contains positive and negative values for the relationship between the species, whereas the positive and negative interactions are combined in the adjacency matrix. Therefore, the values of the concentration matrix are converted to their absolute values in order to detect negative as well as positive interactions in the thresholding process.

The values of the concentration matrix are continuous, and a threshold needs to be chosen in order to separate them between interacting species and non-interacting species. The first approach is to set the threshold to zero, and any positive value would be detected as an interaction. However, there might be small relationship values detected in the matrix between non connected nodes of the network. Therefore, the model would get higher accuracy of detection with a higher threshold. This process is called thresholding and one way to find the optimal threshold is to plot the ROC (receiver operating characteristic) curve. The plot shows values for the TPR (true positive rate) and FPR (false positive rate) defined as:

$$TPR = \frac{TP}{TP + FN} \tag{5.1}$$

$$FPR = \frac{FP}{FP + TN} \tag{5.2}$$

where TP, FP, TN, FN are true positives, false positives, true negatives and false negatives respectively.

The curve is drawn by calculating true positives and false positives at every threshold from a previously sorted array of predictions. The amount of thresholds is chosen based on the variability of the values in the list. The diagonal line of the ROC curve shows a performance of a random classifier, and the classifier performance is evaluated by the AUC metric which is the two-dimensional area under the curve. The optimal threshold is chosen by the G-Mean (geometric mean) between specificity and sensitivity defined as:

$$G - Mean = \sqrt{(Sensitivity \cdot Specificity)} \tag{5.3}$$

where

$$Sensitivity = TPR \text{ and } Specificity = 1 - Sensitivity \tag{5.4}$$

Figure 5.1 shows the ROC curves for all three different networks estimated by the concentration matrix of the simulated data for that network. The values for the AUC are also added to the plot with the optimal thresholds shown as a black dot. It can be seen that the sparse network can be estimated from the concentration matrix with the highest accuracy, whereas the fully connected one with the lowest. Since the covariance matrices obtained from the simulations are often singular, the pseudo-inverse was used when computing the concentration matrix.

The AUC is a threshold insensitive metric, but for deeper comparison additional metrics can be computed at the optimal threshold. Table 5.1 shows values for the entries of the TP, FN, FP, TN, precision and recall for these networks computed at the optimal threshold, where the metrics are defined as:

$$Precision = \frac{TP}{TP + FP} \tag{5.5}$$

$$Recall = \frac{TP}{TP + FN} \tag{5.6}$$

These metrics are computed for the optimal threshold of each network and the results are shown on table 5.1. Precision is a measure of how many of the detected pairs truly form an interaction and recall measures how many of the true interactions are detected. The precision value is highest for the fully connected network. That is to be expected, since all species interact between each other in that case. However, the precision is high on the cost of low recall. In the scale free network, on the other hand, the values are both close to the random prediction. It

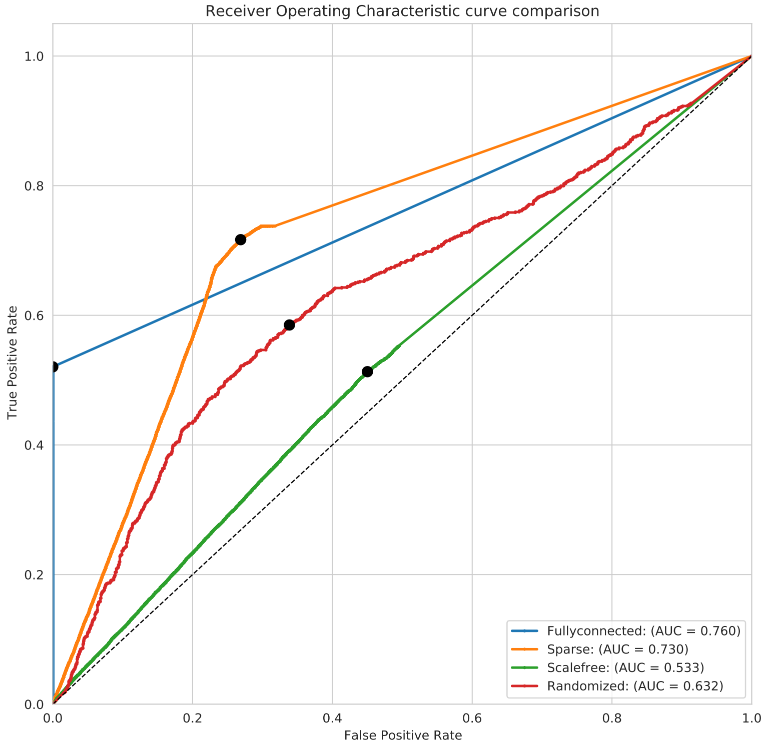


Figure 5.1: Comparison of ROC curves

is interesting to see that even though the sparse network has the highest AUC, these values are low at the optimal threshold. Since the randomized simulation has a lower number of species (60) than the rest, the exact values of the confusion matrix (TP, TN, FP, FN) should not be compared.

In a similar way as the ROC curve, the precision-recall curve shows the values for precision and recall for different thresholds. The area under the precision-recall curve (AUPR) is a common metric used for evaluating the quality of classification. This metric is often adjusted to account for unbalanced data. This is done by calculating the baseline-adjusted $AUPR_{BA}$ computed in the following way:

$$AUPR_{BA} = \frac{AUPR - AUPR_{Random}}{1 - AUPR_{Random}} \tag{5.7}$$

Metric	Fully Connected	Sparse Network	Scale Free Network	Randomized
TP	52415	13384	22128	504
FN	49665	20656	19621	689
FP	4171	27868	31217	1008
TN	148	40492	29424	1399
Precision	$99,67\%$	32.44%	41.42%	33.33%
Recall	51.35%	39.32%	53.01%	42.25%

Table 5.1: Metrics comparison obtained by concentration matrix inference

where AUC_{Random} is the value of the metric obtained with a random classifier.

Figure 5.2 shows the comparison of the precision - recall curves among all four types of network topologies. The best threshold between both metrics is shown as a black dot. Along with the curve, a baseline curve showing the performance of a random classifier for the given data is shown with a dashed line. In addition, the values for the baseline-corrected area AUPR are highlighted. It can be seen that even though the fully connected network can be approximated from the concentration matrix the most accurately, the random classifier value is also very high due to the large imbalance in the ground truth. Among all, the sparse network shows the best accuracy based on threshold invariant metrics (AUC and baseline-adjusted AUPR) and is then used to compare the different classifiers. In the subsequent sections, each of the methods used to obtain the results is explained.

5.2 Graphical Lasso

One of the methods for estimation of the precision matrix is Graphical Lasso[14]. It is used for sparse graph estimation by applying a L^1 (lasso) regularization penalty to the covariance matrix. In the multivariate Gaussian distribution, for a given sample $X \sim N(0, \Sigma)$ the model for the precision matrix $\hat{\Theta} = \Sigma^{-1}$ estimate is defined as:

$$\hat{\Theta} = \text{argmin}_{\Theta \geq 0} \left(tr(S\Theta) - \log\det(\Theta) + \lambda \sum_{j \neq k} |\Theta_{jk}| \right) \tag{5.8}$$

where S is the sample covariance matrix and λ is the regularization parameter.

The graphical lasso algorithm cycles trough all of the variables and fits a modified lasso regression to each in every run. The procedure is repeated until convergence is achieved.

In practice before running the graphical lasso solver, a shrinking transformation on the em-

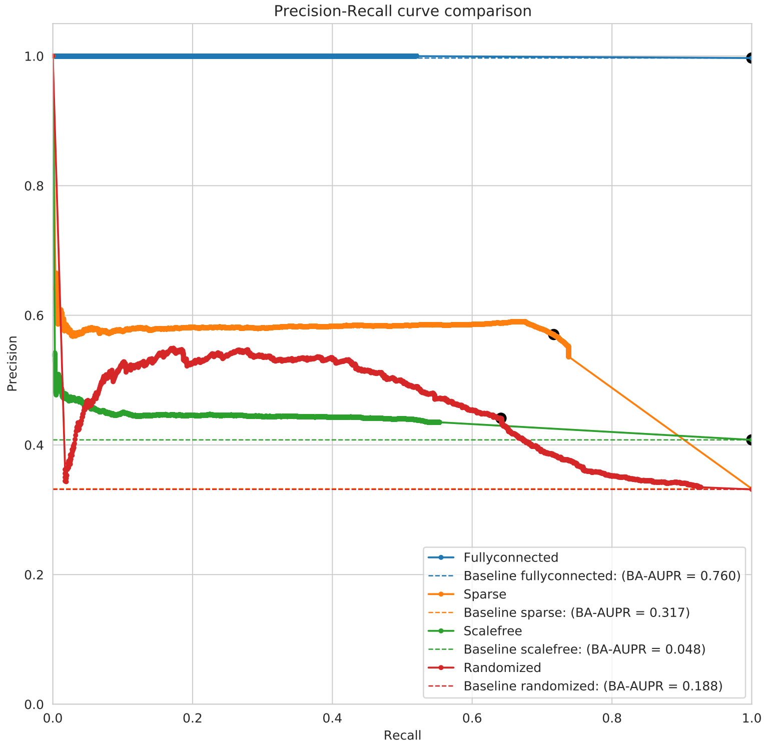


Figure 5.2: Precision - recall curves comparison

pirical covariance applied. The amount of required shrinkage depends on the conditioning of the original data and it is regulated with the parameter α in the following formulation of the shrinkage:

$$\Sigma_{shrunk} = (1-\alpha)\hat{\Sigma} + \alpha\frac{Tr(\hat{\Sigma})}{p}I_p \tag{5.9}$$

where $\hat{\Sigma}$ is the empirical covariance and p is the number of features.

The value used for the parameter α for the shrinkage of the empirical covariance is 0.6 and the regularization parameter λ for the model is set to 0.1. The results obtained with this method based on the AUC show random classification performance for all three networks. At the optimal threshold, the precision values are very high for the sparse and the fully connected network, but at the expense of very low recall values. In fact, this method obtains the highest precision for the sparse network. This method is best if the desired output is a sparse graph with only high precision.

5.3 SparCC

Another approach for inferring the correlation tailored to microbial data is SparCC [15]. It takes into account the sparse distribution of sequencing data and the compositional effects of the data obtained by normalizing the count matrix into fractions of sample abundances. This produces features that are no longer independent and whose feature space is the D-dimensional unit simplex, where p is the number of samples:

$$\mathbb{S}^p = \left\{ x | x_i > 0, \sum_{i=1}^{p} x_i = 1 \right\} \tag{5.10}$$

One of the problems with compositional data is the constant sum constraint (CSC) which removes independence between the variables. In other words, the proportion of a given species depends on the change of abundance in the others. In order to deal with the effects of compositional data, log-ratio transformations are commonly used. In particular, the transformation applied in SparCC is the additive log-ratio transformation **alr** $: S^p \longrightarrow \mathbb{R}^{p-1}$ defined as:

$$alr(x) = \log \frac{x_i}{x_j} = \log x_i - \log x_j \tag{5.11}$$

where x_i and x_j are the fractions of abundance for species i and j.

Alternatively, the centered log-ratio transformation **clr** $: S^p \longrightarrow U, U \subset \mathbb{R}^p$ is commonly used defined as:

$$clr(x) = \log \frac{x_i}{g(x)} = \log x_i - \log g(x) \tag{5.12}$$

where $g(x) = [\prod_{i=1}^{p} x_i]^{\frac{1}{p}}$ is the geometric mean of the vector x.

The isometric log-ratio (ilr) can be used as well [16]. However, these transformations require pre-processing of the zero values. The zero counts occur when the species has not been detected experimentally, or when it is not present in the sample at all. However it is imposible to distinguish between the both cases. In SparCC, a small pseudo-count is added to the zero counts in the original matrix before the transformations. Additionally, the SparCC iterative scheme checks for variables for which no positive correlations can be estimated and removes them. The algorithm converges when no new correlated pairs are identified and in each iteration the basis (absolute abundance) covariance matrix is estimated using the values of the variation matrix T of the log-ratio transformed fractions.

The results of this method are better than a random classifier only for the fully connected network, with an AUC value of 100. Since the absolute correlation estimate from SparCC is used and the values on the diagonal are removed due to the expected constrain of a graph without self-loops, any threshold set on the estimated adjacency matrix successfully detects an interaction between the off diagonal elements. However, this method is primarily designed for inference on sparse networks with low component variability.

5.4 CCrepe

The next method is Compositionality Corrected by Renormalization and Permutation or CCrepe implemented in R [17]. The main idea in the method is creating an expected (null) distribution of the data by iteratively permuting each feature and re-normalizing the samples based on its the previous sum. This is done in order to remove the dependency among the features of compositional data. After the final iteration, a similarity measure is computed between the features which can be Spearman correlation or N-dimensional checkerboard score (NC-score). The results reported are using the default Spearmen correlation metric, since the NC-score metric produced almost identical values. After this, during a bootstraping step the method iterates over subsets of the samples and creates an alternative distribution of the distance metric used. In the end the two resulting distributions are compared with a pooled-variance Z-test. In addition the data is pre-processed and all of the zero count samples are removed.

Since CCrepe is designed for the analysis of compositional data, the counts are first transformed into fractions by adding a pseudo-count of 0.5 to all values and normalizing the rows. The results shown in Table ?? are not higher than a random classifier for the scale free network, but show higher values for the sparse and fully connected case and it is one of the two best performing methods for the sparse network based on the AUC without taking into account the precision and covariance matrices. Taking computation time into account, CCrepe is also the one requiring most CPU time until convergence.

5.5 SPIEC-EASI

Another precision matrix estimation method is SPIEC-EASI (SParse InversE Covariance Estimation for Ecological Association Inference) [18] implemented in R. The algorithm has two different approaches to estimate the precision matrix of the absolute abundance data. The first approach follows the Meinshausen and Buhlmann (MB) method and it is used for neighbourhood selection in sparse high-dimensional graphs. The second approach is based on graphical lasso. Before running the methods however, the data is transformed using the centered log ratio transformation shown at Equation (5.12). In this work, only the MB approach is used in the experiments due to slow convergence of the graphical lasso approach (5.8).

The MB neighbourhood selection method approaches the task by solving p regularized linear regression problems. For each node v_i a convex problem is solved defined as:

$$\hat{\beta^{i\lambda}} = \text{argmin}_{\beta\in\mathbb{R}^{p-1}} \left(\frac{1}{n}||Z^i - Z^{-i}\beta||^2 + \lambda||\beta||_1\right) \qquad (5.13)$$

where $Z \in \mathbb{R}^{N\times p}$ is the matrix of log-ratio transformed abundances and Z^i is the ith column. This formulation is a least-square fit for the the value of β_j for every relationship between the nodes i and j. The regularization parameter is λ tuning the L^1 norm of the vector β calculated as the sum of its absolute values. For higher values of λ the coefficients tend to absolute zero. An edge between the nodes i and j is detected where at least one of the coefficients $\hat{\beta_j^{i,\lambda}}$ or $\hat{\beta_i^{j,\lambda}}$ is positive. In case where both of them are detected, the weight is computed as the average.

The matrix obtained this way depends on the λ parameter, so in order to find the optimal λ a model selection scheme named Stability Approach to Regularization Selection (StARS) is used. It works by sub-sampling the data in each iteration and estimating the edges for various levels of λ. The value for which the incidence matrix is the most stable is chosen as a regularization parameter.

Based on the AUC in the results, this model shows random classification performance for all three networks with lower recall at the optimal threshold than the others. It is most similar to the graphical lasso prediction, with lower precision values.

5.6 Coocur

Coocur is a probabilistic model for species co-ocurence [19]. It highlights the advantage against randomization probability algorithms since id does not try to estimate the null distribution of the data which may be prone to randomization errors. Instead, the probability that two given species 1 and 2 are both found at j samples is given by the following equation:

$$p_j = \frac{C(N, j) \cdot C(N - j, N_2 - j) \cdot C(N - N_2, N_1 - j)}{C(N, N_2) \cdot C(N, N_1)} \tag{5.14}$$

where the numerator is the product between the number of ways the species can be arranged in j samples, and the number of ways that they can be arranged in the remaining samples. The denominator counts the total number of ways the species can occur in the samples.

The experiments on the simulated data are run using the relative abundances. For each network, there are 51040 pair combinations, for the 320 species in 60 samples. However, the method removes the pairs of species where the expected co-occurence is less than 1. For the fully connected network the number of removed pairs is is 36873 pairs or (72.24%), for the sparse network 38107 pairs or (74.66%) and for the scale-free case 38122 pairs or (74.69%). The high number of undetected interactions can lead to poor classification results. Since the p-values in the model estimation matrix are already positive and co-occurences are only calculated for the different species, the pre-processing step of taking the absolute values of the predictions and removing the diagonal elements did not change the results.

5.7 Minerva

Implemented in R, this method is part of the class of statistics called Maximal Information-based Nonparametric Exploration (MINE) statistics and uses the Maximal Information Coefficient (MIC) for identifying the relationships [20]. The method is based on maximizing the Mutual Information (I) between every pair of species X and Y in a iterative fashion. The mutual information between two random variables is defined as:

$$I(X, Y) = H(X) - H(X|Y) \tag{5.15}$$

where $H(X)$ is the entropy of X computed as:

$$H(X) = -\sum_x p_x(x) \log p_x(x) \tag{5.16}$$

and $H(X|Y)$ is the conditional entropy:

$$H(X|Y) = \sum_y p_y(y) \left[- \sum_x p_{x|y}(x|y) \log \left(p_{x|y}(x|y) \right) \right] \qquad (5.17)$$

The mutual information is a measure of how the uncertainty (entropy) of one random variable changes when introducing information about another. High mutual information corresponds to a large reduce in uncertainty and suggests a significant interaction between the features. This method tries to spot significant interaction areas on the scatter-plot drawn between any pair of features. The plot is then iteratively divided into girds of different resolution and the grid which maximizes the mutual information between the variables is chosen. After normalizing the results to account for the sample size, the MIC is obtained for each interaction pair.

The final matrix has positive entries and with no self-loops. From the results shown in **??** it can be seen that this model obtains the highest AUC for the sparse network, apart from the precision matrix. It also shows high results for the fully connected network experiment, second only to SparCC.

5.8 gCoda

The final method used in this study is gCoda [21]. It is also based on a L^1 (lasso) regularization as previous methods. One of the assumptions in this method is that the log ratios of the abundances follow a multivariate normal distribution. The model tries to estimate the inverse covariance matrix through an MM (Majorization - Minimization) optimization algorithm. This is set by defining a majorizing function for the estimate whose minimization is represented as a Graphical Lasso model. This function is minimized iteratively until convergence, aiming to detect the most sparse inverse covariance consistent with the data. The regularization parameter λ is computed by choosing the value for which the EBIC (Extended Bayesian Information Criteria) is the lowest for the number of edges in the network. A lower EBIC represents a model with lower variance.

6

Methods Comparison

This chapter contains commentary on the results obtained by testing the methods explained in Chapter 5 using the simulated data matrices from Chapter 4. Each inferred matrix from a given method is compared with the corresponding ground truth adjacency matrix. There are two sections in this chapter comparing results obtained by the sparse and the randomized network topology.

From the performance of the concentration matrix as a method, explained in the first section of Chapter 5, it can be seen that the sparse network topology is estimated most accurately among them, without accounting for the fully connected network since it not a natural behaviour. For this reason in Section 6.1 10 instances of the sparse topology have been simulated, using the parameters shown in Table A.1, with 320 species and 60 conditions. Different accuracy metrics are computed and the mean and variation among the 10 networks is displayed. Since the randomized network scenario consists of 60 species in 200 simulations, which means that the number of samples is bigger than the number of features, 10 separate networks have been tested for this scenario as well. The results are shown in Section 6.2.

Since in these performance evaluations, the negative and positive interactions are combined in order to classify between the presence and absence of a given edge, in Section 6.3 a comparison on the ability of the networks to detect positive and negative interactions is shown.

6.1 Sparse Network simulations

This section describes the results obtained from the 10 sparse network topology simulations. Before the metrics are computed, there are few pre-processing steps of the inferred data. The inferred data can contain negative and positive predictions in a different scale depending on the method. In order to have comparable data, each feature of the matrix is scaled based on its maximum absolute value. This allows for the data to fall in a selected range [-1, 1] without scaling the 0 values and losing information of the sparsity. Since here methods are tested on both positive and negative interactions, only the absolute value of the matrix is considered. Then, due to the network topology constraint, no self loops are allowed because only inter-species interactions are investigated. Thus, the values on the diagonal of the matrix are removed. If any species have not been detected in any of the simulations of absolute abundance (generation of abundance as it is in nature), they are removed from the comparison. In these sparse experiments, all species were detected in at least one simulation. The values of the matrix and the corresponding ground truth are then flattened and sorted in order to computed the metrics.

6 different methods have been tested, along with the concentration matrix, which serves as a baseline for accuracy. The mean values obtained for the AUPR, baseline-adjusted AUPR, AUC and RMSD along with the standard deviation are shown on Table 6.1. The bar-plots for these values are shown on Figure 6.1.

It can be seen that the best performing methods are *Cooccur* and *Minerva*, with average AUC of 76.1% and 79.03% respectively. Minerva outperforms Cooccur in all of the other metrics except for the RMSD and is therefore the best classifier for the sparse scenario.

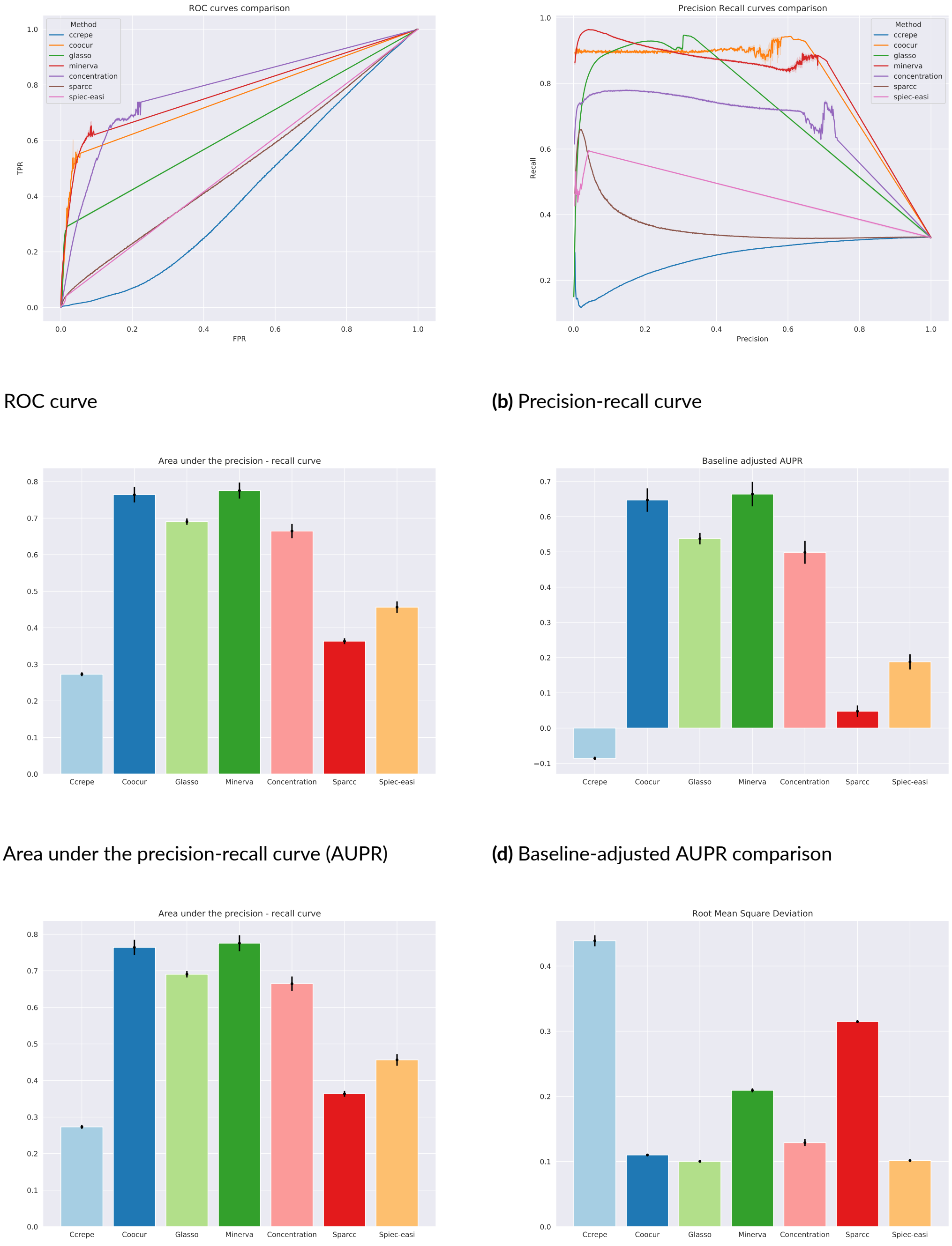
Most of the other methods are under-performing with accuracy lower than the concentration matrix prediction or even lower than the random baseline - like CCrepe. Graphical Lasso, however is a method whose prediction accuracy seems to be very close to the one of the concentration matrix.

Method	AUPR	BA-AUPR	AUC	RMSD
Concentration Matrix	66.47% (±1.98)	49.88% (±3.24)	77.22% (±1.53)	0.13
Graphical Lasso	69.05% (±0.87)	18.79% (±2.17)	51.23% (±0.28)	0.1
SparCC	36.08% (±0.2)	4.25% (±0.31)	50.45% (±0.34)	0.31
CCrepe	27.3% (±0.51)	−8.61% (±0.48)	41.6% (±0.78)	0.44
SPIEC-EASI	45.64% (±1.58)	18.79% (±2.17)	51.23% (±0.28)	0.1
Cooccur	76.54% (±2.1)	64.72% (±3.34)	76.1% (±2.05)	0.11
Minerva	77.54% (±2.19)	49.88% (±3.46)	79.03% (±1.84)	0.21

Table 6.1: Methods comparison for sparse topology - table

It should be noted that the values for the RMSD metric are computed with respect to the ground truth weight matrix instead of the adjacency matrix. Some methods like Graphical Lasso and SPIEC-EASI obtain the lowest error among the methods. This could suggest that even if not all true interactions are detected by the methods in the thresholding process, they could infer the weights among the connections more accurately. The inference of the strength of the interactions is another problem worth exploring, but this work focuses on the inference of the existing edges and the underlying adjacency matrix.

Along with the bar-plots shown on Figure 6.1, a comparison between the ROC and Precision - Recall curves is shown. On the ROC plot, it can be seen that the best performing methods have a similar behaviour, Coocur and Minerva, close to the concentration matrix. It is also visible that the curve for most of the methods have is steep in the beginning, for low FPR rates. The partial AUC is another useful metric that shows the importance of the performance of a classifier when the FPR rate is controlled. For a default threshold of FPR (maximum value for FPR) of 0.1, the partial AUC metric for Coocur and Minerva is over 70%, over 60% for Graphical Lasso and the concentration matrix and around 50% for the rest. The precision-recall curve also shows a distinction between the high and low performing methods.



(a) ROC curve

(b) Precision-recall curve

(c) Area under the precision-recall curve (AUPR)

(d) Baseline-adjusted AUPR comparison

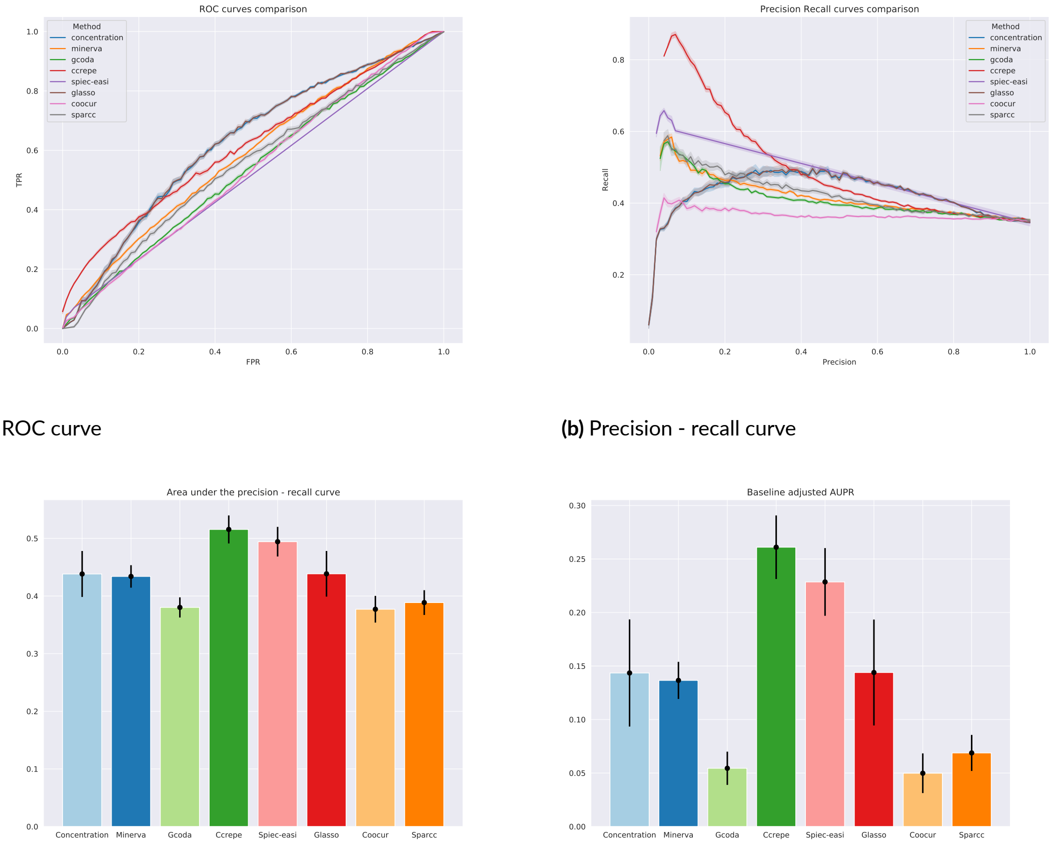
(e) AUC comparison

(f) RMSD comparison

Figure 6.1: Methods comparison for sparse network topology - figures

6.2 Randomized Network Simulation

Similarly to the sparse network simulation, for the randomized topology 10 different methods were simulated. The pre-processing of the data is done in the same way, with the difference that in these experiments some species had absolute data abundance equal to zero in each simulation and are therefore removed. In addition, the *gCoda* method is tested only in these experiment, since it was not computationally possible to test it in the sparse scenario.



(a) ROC curve

(b) Precision - recall curve

(c) Area under the precision-recall curve

(d) Baseline adjusted AUPR

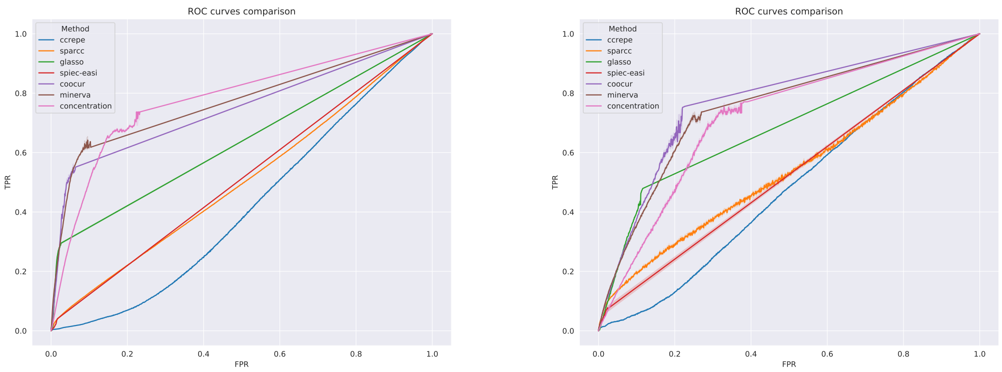
Figure 6.2: Methods comparison for randomized network topology

Figure 6.2 shows the results obtained for each method, the mean among the 10 simulations and the standard deviation. It can be seen that the results are more variable when compared to the sparse network scenario and the accuracy is lower. Based on the ROC curve, Graphical Lasso is the only one outperforming the concentration matrix, with CCrepe following the performance. However, when looking at the precision - recall curve, even though the overall accuracy among the methods is low, SPIEC-EASI and CCrepe have higher values for both AUPR and baseline-adjusted AUPR.

6.3 Positive and Negative interactions

This section shows a comparison between the detection of positive and negative interactions between the methods. All method inferred matrices are compared with ground truth matrices for species competition (negative interactions) or species cooperation (positive interactions). Figure 6.3 shows the comparison for the sparse network simulations and Figure 6.4 for the randomized scenario.

For the sparse network simulation there is not a significant difference between the detection of positive and negative interactions. The AUC values are slightly higher for the positive interactions for the best performing methods: Cooccur and Minerva, but for the concentration matrix as well. For all of the remaining methods, there is a slight increase in the negative interaction AUC. In general, the values for the positive interactions are less variable, but they show very low performance on other metrics such as AUPR.

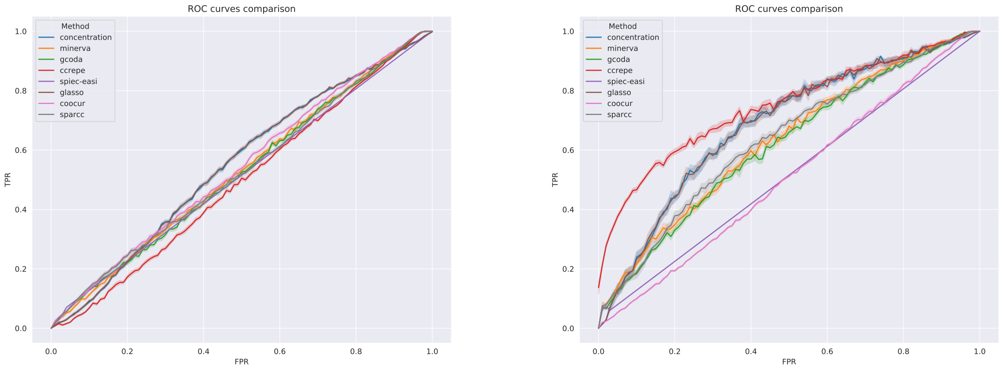


(a) Positive interactions

(b) ROC curve for negative interactions

Figure 6.3: Positive and negative interactions for sparse network

From the comparison on Figure 6.4 it can be seen that the performance of almost all methods is better for the negative interactions retrieval in the randomized scenario. Here, the best performing based on all of the metrics was CCrepe, but it can be argued that this is only because of the detection of the negative interactions. The AUC for this method is 74.9% for the negative interactions and only 50.65% for the positive case. The rest of the methods show higher AUC values for the negative interactions as well.



(a) ROC curve for positive interactions

(b) ROC curve for negative interactions

Figure 6.4: Positive and negative interactions for randomized network

7 Conclusion

The microbial organisms play a crucial role in all natural processes in the world from controlling the state of their environment to influencing disease risks in host organisms. Understanding the details of their interactions can lead to diverse solution of issues from ecosystem preservation to drug development. The aim of this thesis on reverse engineering networks is to test the performance of specialized methods for inferring these relationships from experiment data. Through simulating both ground truth and experiment data, 7 different methods have been tested on different network topologies. Depending on the network, different performance was recorded, suggesting that the methods are sensitive to the type of data. Many of them assume sparse conditions and high dimensional data and they performed better in this scenario, represented by the sparse network. The highest AUC value was obtained by Minerva of 79%, but this is lower than the literature reported performance of all of the methods. In a more general sense, the results are consistent to the findings reported in [22]. The authors have tested a different set of reverse engineering methods with a chosen network topology of the data and report lower accuracy for interaction detection than expected and higher accuracy for negative interactions detection when compared with the positive ones. This shows that this field of research remains open to alternative solutions that can more accurately detect species relationships. Given the sensitivity of the data, the accuracy reported here is not enough to detect species interaction with high certainty, but can be a starting point of further exploration.

References

[1] K. J. Locey and J. T. Lennon, "Scaling laws predict global microbial diversity," *Proceedings of the National Academy of Sciences*, vol. 113, no. 21, pp. 5970–5975, 2016. [Online]. Available: https://www.pnas.org/doi/abs/10.1073/pnas.1521291113

[2] J. Durack and S. V. Lynch, "The gut microbiome: Relationships with disease and opportunities for therapy," *J. Exp. Med.*, vol. 216, no. 1, pp. 20–40, Jan. 2019.

[3] J. C. Wooley, A. Godzik, and I. Friedberg, "A primer on metagenomics," *PLOS Computational Biology*, vol. 6, no. 2, pp. 1–13, 02 2010. [Online]. Available: https://doi.org/10.1371/journal.pcbi.1000667

[4] F. Sanger, S. Nicklen, and A. R. Coulson, "Dna sequencing with chain-terminating inhibitors," *Proceedings of the National Academy of Sciences*, vol. 74, no. 12, pp. 5463–5467, 1977. [Online]. Available: https://www.pnas.org/doi/abs/10.1073/pnas.74.12.5463

[5] L. Bragg and G. W. Tyson, "Metagenomics using next-generation sequencing," in *Environmental microbiology*. Springer, 2014, pp. 183–201.

[6] A. Kamble, S. Sawant, and H. Singh, "16S ribosomal RNA gene-based metagenomics: A review," *Biomedical Research Journal*, vol. 7, no. 1, pp. 5–11, 2020. [Online]. Available: https://www.brjnmims.org/article.asp?issn=2349-3666;year=2020;volume=7;issue=1;spage=5;epage=11;aulast=Kamble;t=6

[7] T. S. Tshikantwa, M. W. Ullah, F. He, and G. Yang, "Current trends and potential applications of microbial interactions for human welfare," *Frontiers in Microbiology*, vol. 9, 2018. [Online]. Available: https://www.frontiersin.org/articles/10.3389/fmicb.2018.01156

[8] R. Balakrishnan and K. Ranganathan, *A textbook of graph theory*. Springer Science & Business Media, 2012.

[9] A.-L. Barabási and M. Pósfai, *Network science*. Cambridge: Cambridge University Press, 2016. [Online]. Available: http://barabasi.com/networksciencebook/

[10] R. Marsland, P. Mehta, W. Cui, and J. Goldford, "The community simulator: A python package for microbial ecology," *bioRxiv*, 2020.

[11] I. Patuzzi, G. Baruzzo, C. Losasso, A. Ricci, and B. Di Camillo, "metasparsim: a 16s rrna gene sequencing count data simulator," *BMC Bioinformatics*, vol. 20, no. 9, p. 416, Nov 2019. [Online]. Available: https://doi.org/10.1186/s12859-019-2882-6

[12] G. Schröer and D. Trenkler, "Exact and randomization distributions of kolmogorov-smirnov tests two or three samples," *Computational Statistics Data Analysis*, vol. 20, no. 2, pp. 185–202, 1995. [Online]. Available: https://www.sciencedirect.com/science/article/pii/016794739400040P

[13] T. Arnold and J. Emerson, "Nonparametric goodness-of-fit tests for discrete null distributions," *R Journal*, vol. 3, 12 2011.

[14] J. Friedman, T. Hastie, and R. Tibshirani, "Sparse inverse covariance estimation with the graphical lasso," *Biostatistics*, vol. 9, no. 3, pp. 432–441, 12 2007. [Online]. Available: https://doi.org/10.1093/biostatistics/kxm045

[15] J. Friedman and E. J. Alm, "Inferring correlation networks from genomic survey data," *PLOS Computational Biology*, vol. 8, no. 9, pp. 1–11, 09 2012. [Online]. Available: https://doi.org/10.1371/journal.pcbi.1002687

[16] J. J. Egozcue, V. Pawlowsky-Glahn, G. Mateu-Figueras, and C. Barceló-Vidal, "Isometric logratio transformations for compositional data analysis," *Mathematical Geology*, vol. 35, no. 3, pp. 279–300, Apr 2003. [Online]. Available: https://doi.org/10.1023/A:1023818214614

[17] C. B. Emma Schwager and G. Weingart, *ccrepe: ccrepe*$_{a}nd_{n}c.score, 2022, rpackageversion 1.32.0$.

[18] Z. D. Kurtz, C. L. Müller, E. R. Miraldi, D. R. Littman, M. J. Blaser, and R. A. Bonneau, "Sparse and compositionally robust inference of microbial ecological networks," *PLOS Computational Biology*, vol. 11, no. 5, pp. 1–25, 05 2015. [Online]. Available: https://doi.org/10.1371/journal.pcbi.1004226

[19] J. Veech, "A probabilistic model for analysing species co-occurrence," *Global Ecology and Biogeography*, vol. 22, 02 2013.

[20] D. N. Reshef, Y. A. Reshef, H. K. Finucane, S. R. Grossman, G. McVean, P. J. Turnbaugh, E. S. Lander, M. Mitzenmacher, and P. C. Sabeti, "Detecting novel associations in large data sets," pp. 1518–1524, Dec. 2011.

[21] H. Fang, C. Huang, H. Zhao, and M. Deng, "GCoda: Conditional dependence network inference for compositional data," *J. Comput. Biol.*, vol. 24, no. 7, pp. 699–708, Jul. 2017.

[22] H. Hirano and K. Takemoto, "Difficulty in inferring microbial community structure based on co-occurrence network approaches," *BMC Bioinformatics*, vol. 20, no. 1, p. 329, Jun 2019. [Online]. Available: https://doi.org/10.1186/s12859-019-2915-1

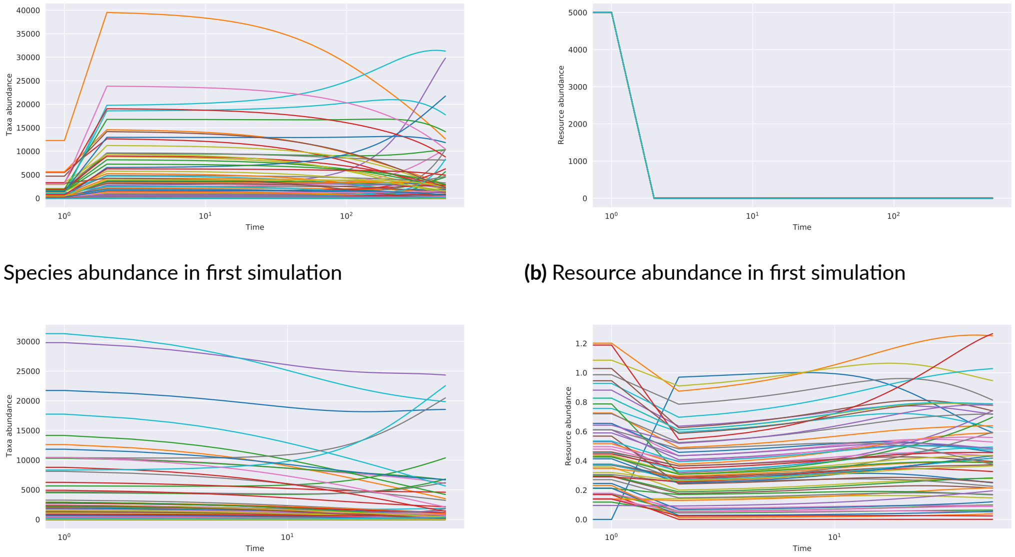
A

Appendix

This chapter contains additional tables and figures in the order of reference in the main text.

Parameter	Description	Value	Network
SA	Number of specialized species	300	Fully connected
		300	Sparse
		6 * 50	Scale free
MA	Number of resources	60	Fully connected
		60	Sparse
		6 * 10	Scale free
fs	Fraction of conversion to same resource	0.45	Fully connected
		0.99999	Sparse
		0.99999	Scale free
fw	Fraction of conversion to waste resource	0.45	Fully connected
		0.000001	Sparse
		0.000001	Scale free
muc	Mean sum of consumption rates	7	Fully connected
		1	Sparse
		7	Scale free
nwells	Number of wells	1	
Sgen	Number of generalist species	20	
S	Total number of species	320	
supply	Resource supply (external, self-renewing or off)	external	
g	Conversion factor for species	1	
m	Maintenance cost for species	1	
sparsity	Effective sparsity on the metabolic matrix	0.2	
regulation	Metabolic regulation (energy or independent)	independent	
response	Functional response (type I, type II or type III) [linear, saturating Monod or Hill/sigmoid-like]	type I	
l	Leakage fraction	0.8	
w	Energy density for resources	1	
tau	Turnover rate for externally supplied resources	1	
q	Preference strength for specialist families	0.9	
c0	Sum of background consumption rates	0.0	
c1	Specific consumption rate	1	
sampling	Choice of sampling distribution	Binary	

Table A.1: Parameters used for simulations



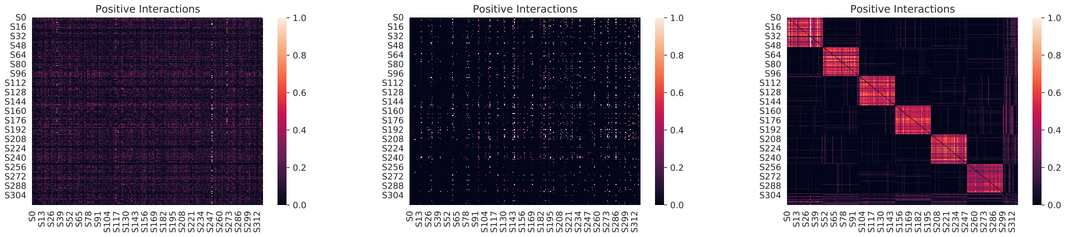
(a) Species abundance in first simulation

(b) Resource abundance in first simulation

(c) Species abundance in second simulation (I condition)

(d) Resource abundance in second simulation (I condition)

Figure A.1: Species and resource simulated abundances

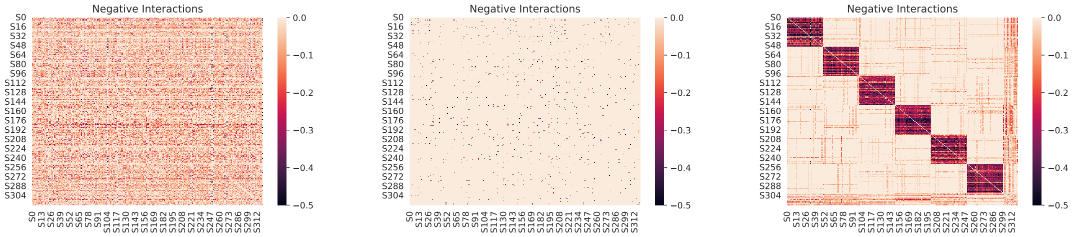


(a) Fully connected network

(b) Sparse network

(c) Scale free network

Figure A.2: Positive weight matrices comparison



(a) Fully connected network

(b) Sparse network

(c) Scale free network

Figure A.3: Negative weight matrices comparison

Acknowledgments

This thesis is done in collaboration with the systems biology research group at the University of Padova. I would like to thank my supervisor prof. Barbara Di Camillo for her guidance and support during the process. Furthermore, I would like to thank Ada Rosatto, Marco Capellato and Giacomo Baruzzo for their help and supervision.