Transnational University Limburg is a unique collaboration of two universities in two countries: the University of Hasselt and Maastricht University.

UHASSELT
KNOWLEDGE IN ACTION

www.uhasselt.be
Universiteit Hasselt
Campus Hasselt:
Martelarenlaan 42 | 3500 Hasselt
Campus Diepenbeek:
Agoralaan Gebouw D | 3590 Diepenbeek

UHASSELT
KNOWLEDGE IN ACTION

Maastricht University

Faculty of Sciences
School for Information Technology
Master of Statistics and Data Science

Master's thesis

Effect of parameter choices of bioinformatic pipelines on abundance tables using TDA (Topological Data Analysis)

Mohammad Sazegar

Thesis presented in fulfillment of the requirements for the degree of Master of Statistics and Data Science, specialization Data Science

SUPERVISOR :
Prof. dr. ir. Jan AERTS

2023
2024

Faculty of Sciences
School for Information Technology
Master of Statistics and Data Science

Master's thesis

Effect of parameter choices of bioinformatic pipelines on abundance tables using TDA (Topological Data Analysis)

Mohammad Sazegar

Thesis presented in fulfillment of the requirements for the degree of Master of Statistics and Data Science, specialization Data Science

SUPERVISOR :
Prof. dr. ir. Jan AERTS

Faculty of Sciences
School for Information Technology
Master of Statistics and Data Science

Master's thesis

Effect of parameter choices of bioinformatic pipelines on abundance tables using TDA (Topological Data Analysis)

Mohammad Sazegar

Thesis presented in fulfillment of the requirements for the degree of Master of Statistics and Data Science, specialization Data Science

SUPERVISOR :
Prof. dr. ir. Jan AERTS

Abstract

The development of advanced sequencing technologies has transformed the field of microbial ecology, enabling researchers to explore microbial communities with incredible depth and precision. accuracy and reliability of the resulting amplicon sequence variant (ASV) tables are influenced by various settings in the bioinformatics pipelines, however, not all studies in the field give due consideration to the importance and potential effect of the parameters in the workflow. Understanding the settings is crucial because these settings can significantly affect the outcomes of microbial community analysis.

This study aims to answer this question using data visualization technique focusing on topological relations between pipelines and their corresponding set of parameters. A network graph was developed using ASV tables of dataset Mothur for the analysis. Parameters related to truncation are the most important one among the others.

Network is constructed out of smaller regions, namely three clusters (A,B, and C) and a tail are present in the network and two paths. A smooth transition of ASV values is present from the tail towards the main body. Also several patterns for different parameters pf the pipelines were observed.

In conclusion noticeable similar patterns were observed in both parameters and microbiome graphs and consequently ASV tables are sensitive to different processing pipeline parameters. Abundance of microbiomes not only varies from low to high values by choosing different set of parameters, but also could affect the presence of an ASV table as an outcome. Parameters related to truncation (“trunclen_f”, “trunclen_r” and “truncq”) were the significant ones resulting in lower ASV values in the tail for higher values of parameters.

Keywords: ASV tables, Biological pipeline, Data visualization, TDA (Topological Data Analysis), Abundance tables, DADA2

Abstract

Contents

List of Figures

List of Tables

1 Introduction

The development of advanced sequencing technologies has transformed the field of microbial ecology, enabling researchers to explore microbial communities with incredible depth and precision. Amplicon sequencing, particularly targeting the 16S rRNA gene, has become a standard approach for identifying microbial variety. The configuration of microbial community structure offers valuable insights into structure of natural ecosystems and also detailed relationship between the host and its bacterial inhabitants However, the accuracy and reliability of the resulting amplicon sequence variant (ASV) tables, which detail the presence and abundance of microbial species, are influenced by various settings in the bioinformatics pipelines used to process sequencing data. Not all studies in the field give due consideration to the importance and potential effect of these parameters in the workflow.

Understanding the settings within biological pipelines is crucial because these settings can significantly affect the outcomes of microbial community analysis. Misinterpretation of microbial variety and composition can arise from inappropriate settings, leading to incorrect ecological and clinical conclusions. Key settings that require careful consideration include quality filtering thresholds, chimera detection methods, sequence clustering algorithms, and taxonomic classification techniques. For instance, Edgar (2016) demonstrated that variations in chimera detection approaches could lead to substantial differences in the inferred microbial community structure. Similarly, Callahan et al. (2017) highlighted the impact of sequence denoising methods on the generation of ASVs, showing that different methods could produce varying levels of resolution in microbial variety analysis.

Several techniques have been used to investigate the effects of bioinformatics pipeline settings on ASV tables. These techniques can be broadly categorized into simulation studies, comparative analyses, and empirical evaluations. Simulation studies often use artificially created datasets to evaluate the performance of different pipeline configurations. Simulated data provide a controlled environment where the true composition of the microbial community is known, allowing for precise assessment of how different settings influence the accuracy of ASV tables. For instance, Prodan et al. (2020) used simulated data to compare the performance of various quality filtering and denoising methods, demonstrating that some methods were more robust to sequencing errors than others. Comparative analyses involve processing the same set of biological samples through different pipelines or settings to assess variability in the

resulting ASV tables. Bokulich et al. (2018) conducted a comprehensive comparison of commonly used pipelines such as QIIME 2, DADA2, and mothur, revealing significant discrepancies in the diversity metrics and taxonomic assignments produced by each pipeline. Empirical evaluations use real-world datasets to assess the performance of pipeline settings. These studies often focus on specific aspects of the pipelines, such as the impact of different sequence clustering thresholds or the effectiveness of various taxonomic classifiers. Schloss (2021) conducted an empirical study using environmental samples to evaluate the performance of different clustering algorithms, showing that more stringent clustering thresholds could reduce the number of spurious ASVs but might also overlook rare species.

Recent studies have provided valuable insights into the optimal configurations of bioinformatics pipelines for generating accurate ASV tables. Key findings include: Quality filtering is a critical step that can significantly affect downstream analyses. Nguyen et al. (2021) found that stringent quality filtering parameters improved the accuracy of ASV tables by removing low-quality reads that could introduce noise into the dataset. However, overly stringent filtering could also lead to the loss of genuine biological sequences. Effective chimera detection is essential for accurate microbial profiling. Edgar (2016) showed that different chimera detection algorithms, such as UCHIME and DADA2's built-in method, varied in their sensitivity and specificity, with some methods being more prone to false positives or negatives. Sequence denoising methods, which aim to correct sequencing errors and distinguish true biological sequences from artifacts, are crucial for generating high-resolution ASV tables. Callahan et al. (2017) demonstrated that DADA2's denoising approach outperformed traditional clustering methods by providing finer resolution and reducing the incidence of spurious ASVs. The choice of taxonomic classifier can impact the accuracy of taxonomic assignments. Bokulich et al. (2018) found that classifiers based on naive Bayesian approaches, such as the RDP classifier, generally performed well across various datasets, but their accuracy depended on the quality and completeness of the reference database used. Comprehensive comparisons of different bioinformatics pipelines revealed that no single pipeline consistently outperformed others across all metrics. Prodan et al. (2020) emphasized the importance of selecting pipeline settings tailored to the specific characteristics of the dataset and the research questions being addressed.

Prior work has offered important information on bioinformatics pipeline settings on ASV tables, including quality filtering, chimera detection, sequence denoising, and taxonomic classification. However, these investigations have not paid much

attention to the effects of parameter values in the various analysis packages including DADA2. In this study, our goal is to fill this gap by presenting a systematic analysis of the impact of different parameter settings in the DADA2 pipeline. Thus, by dissecting how various contexts affect the precision and the scope of the ASV tables produced by DADA2, we hope to advance the understanding of the microbial community analysis and offer practical recommendations that would improve the reliability of the subsequent investigations in microbial ecology.

Data visualization techniques play a pivotal role in elucidating patterns within high-dimensional data, particularly in fields such as microbiology where complex datasets are commonplace. By employing advanced visualization methods like network graphs, researchers can effectively depict intricate relationships and structures within microbial communities. For example, Smith et al. (2020) utilized network graphs to explore the co-occurrence patterns of microbial taxa across different environmental samples, revealing distinct ecological niches and potential symbiotic relationships among species. Such visual representations not only aid in understanding microbial community structures but also facilitate hypothesis generation and the formulation of testable predictions regarding ecological interactions.

Moreover, data visualization allows researchers to navigate and interpret complex datasets more intuitively, enabling the identification of outliers, clusters, and trends that might signify biological significance. In microbial ecology, where understanding community dynamics is crucial for ecosystem health and function, visualization techniques like principal component analysis (PCA, Jolliffe 2002) and multidimensional scaling (MDS, Borg et al., 2005), provide graphical representations that summarize variation across samples or experimental conditions. These techniques enable researchers to detect patterns of microbial diversity or community composition that are influenced by experimental variables or bioinformatics pipeline parameters (Jones et al., 2019). By integrating data visualization into their analyses, researchers can uncover hidden patterns and relationships within high-dimensional data, advancing our understanding of microbial ecosystems and their responses to environmental changes.

The main question of this study is to investigate how different choices in bioinformatic pipeline parameters influence the composition and structure of ASV tables using data visualization techniques, such as network graphs. Using network graphs to visualize interactions and associations among microbial groups based on their abundance patterns, researchers aim to understand how varying pipeline parameters impact community changes. This approach allows for identifying clusters, outliers, and connectivity patterns that may be affected by specific

parameter configurations. The study aims to uncover which settings lead to more accurate and informative ASV tables, thereby optimizing bioinformatics pipelines for robust microbial ecology research. Ultimately, by leveraging data visualization to explore these relationships, the research aims to enhance our understanding of microbial community responses to environmental changes and improve the reliability of microbial diversity assessments in various scientific applications.

2 Relevance, Stakeholders & Ethics

By assessing proper set of values for parameters involved in a pipeline, researchers can more precisely look into presence of different microbiomes and their abundance in the samples. In addition to that, rare microbiomes could be find with higher chances. This is crucial for identifying accurate and informative ASV tables and its potential to enhance public health outcomes and environmental management. Understanding how bioinformatics pipeline choices influence the accuracy of microbial diversity assessments is crucial for stakeholders such as healthcare professionals, biologists, statisticians, and environmental scientists. Accurate ASV tables facilitate early detection and monitoring of microbial communities in diverse ecosystems, aiding in the prediction and prevention of disease outbreaks and environmental disturbances.

3 Methodology

3.1 Dataset

The dataset is coming from Kozich JJ et al.(2013), Mothur MiSeq SOP project which aimed to demonstrate a standard operating procedure (SOP) for processing 16S rRNA gene sequences.

Dataset used is in this thesis is part of a bigger dataset representing microbiomes in fresh feces collected from mice for 365 days post weaning on a daily basis consisting of data belongs to a single mice (Female number 3) for nine time points at the early stage of the study (days 0 to 3 and 5 to 9) and ten time points at the middle of the study (days 141 to 150). Table 1 shows the name of the samples, ID and a brief explanation of it.

Table 1 List of samples in the dataset

ID	Sample Name	Explanation
1	F3D0_S188_L001	Female 3 on Day 0
2	F3D1_S189_L001	Female 3 on Day 1
3	F3D141_S207_L001	Female 3 on Day 141
4	F3D142_S208_L001	Female 3 on Day 142
5	F3D143_S209_L001	Female 3 on Day 143
6	F3D144_S210_L001	Female 3 on Day 144
7	F3D145_S211_L001	Female 3 on Day 145
8	F3D146_S212_L001	Female 3 on Day 146
9	F3D147_S213_L001	Female 3 on Day 147
10	F3D148_S214_L001	Female 3 on Day 148
11	F3D149_S215_L001	Female 3 on Day 149
12	F3D150_S216_L001	Female 3 on Day 150
13	F3D2_S190_L001	Female 3 on Day 2
14	F3D3_S191_L001	Female 3 on Day 3
15	F3D5_S193_L001	Female 3 on Day 5
16	F3D6_S194_L001	Female 3 on Day 6
17	F3D7_S195_L001	Female 3 on Day 7
18	F3D8_S196_L001	Female 3 on Day 8
19	F3D9_S197_L001	Female 3 on Day 9

To answer the main question ASV tables and the parameters used for generating them are needed. An ASV table is consisting of rRNA sequences as columns, different samples as rows, and abundance of each sequence in specific sample as the cell values. An example of the primary ASV tables is shown in Figure 1Figure 2. These ASV tables are provided to us by “Jannes Peeters” for which he used DADA2 package to transform the raw data of “MiSeq SOP” into ASV tables. This package uses several functions for filtering and correcting the errors in read and then identifying unique biological sequences and compiling them into ASV tables, providing a snapshot of microbial community composition in the sample.

	TACGGAGGATGCGAGCGTTA	TACGGAGGATGCGAGCGTTA	TACGGAGGATGCGAGCGTTA	TACGGAGGATGCGAGCGTTA	TACGGAGGATCCGAGCGTTA	TACGGAGGATGCGAGCGTTA	TACGGAGGATTCAAGCGTTAT	TACGGAGGATGCGAGCGTTA	TACGGAGGATGCGAGCGTTA	TACGTAGGTGGCAAGCGTTG	TACGGAGGATGCGAGCGTTA	TACGTAGGTGGCAAGCGTTA	TACGGAGGATGCGAGCGTTA	TACGTAGGTGGCAAGCGTTA	TACGTAGGGGCAAGCGTTA	TACGTAGGTGGCAAGCGTTA	TACGGAGGATGCGAGCGTTA	TACGTAGGTGGCAAGCGTTA	TACGTAGGTGGCAAGCGTTA	TACGTAGGGGCAAGCGTTA	TACGTAGGGGCAAGCGTTA
F3D0_S188_L001	603	355	479	483	167	502	197	296	176	22	232	55	112	100	86	75	70	46	345	47
F3D1_S189_L001	424	367	252	79	146	44	206	107	113	142	43	137	0	353	0	34	114	63	74	14
F3D141_S207_L001	475	383	361	548	200	350	345	266	151	220	158	12	71	37	110	46	6	52	58	93
F3D142_S208_L001	314	315	172	183	193	210	94	171	84	59	106	112	69	11	52	31	7	0	14	39
F3D143_S209_L001	244	188	228	253	138	264	89	166	76	98	118	45	64	11	41	20	0	17	14	66
F3D144_S210_L001	447	300	331	390	120	401	50	262	169	330	167	18	80	13	116	52	6	52	18	14
F3D145_S211_L001	666	522	570	622	333	503	138	422	258	425	269	25	141	15	129	114	5	44	15	15
F3D146_S212_L001	334	242	278	430	193	291	76	231	106	244	154	5	0	27	37	35	4	19	23	83

Figure 1 Example of an ASV table (raw rRNA as columns)

In each function several parameters used for fine-tuning and optimization to increase accuracy and reliability of downstream analysis in microbiome research. A pipeline in this context refers to a set of values for different parameters involved in the process of generating ASV tables using DADA2 package. In this study, 700 pipelines with 24 parameters were used to create ASV tables. List of parameters used in this process with a brief explanation of them is presented in Table 2. Values for these pipelines are generated randomly from predefined ranges after discussion with expertise in the field. These values explained in detail in section 4.1.1.

Table 2 List of the pipeline parameters

Parameter	Explanation
truncQ	This parameter specifies the minimum quality score required to retain a base during truncation (removal of low-quality bases from the end of reads).
truncLen_f	Length to which forward reads should be truncated.
truncLen_r	Length to which reverse reads should be truncated.
trimLeft_f	Number of bases to trim from the 5' end of forward reads.
trimLeft_r	Number of bases to trim from the 5' end of reverse reads.
trimRight_f	Number of bases to trim from the 3' end of forward reads.
trimRight_r	Number of bases to trim from the 3' end of reverse reads.
minLen	Minimum length a read must have after truncation and trimming to be retained.
minQ	Minimum acceptable quality score for a base.
maxEE_f	Maximum expected errors allowed in forward reads.
maxEE_r	Maximum expected errors allowed in reverse reads.
rm.phix	Whether to remove PhiX spike-in sequences from the data.
rm.lowcomplex	Whether to remove low-complexity sequences.
nbases	Number of Ns allowed in a read.
randomize	Whether to shuffle input sequences prior to processing.
max_consist	Maximum number of allowable erroneous base calls in a consistent region.
omega_c	Threshold for the fraction of errors to be expected in a consistent region.
selfConsist	Whether to use self-consistency algorithm for error modeling.
pool	Whether to pool samples for error rate estimation.
minOverlap	Minimum required overlap between forward and reverse reads for merging.
maxMismatch	Maximum number of mismatches allowed in the overlap region during merging.
chim.method	Method to use for chimera removal.
minBoot	Minimum bootstrap support required to retain a sequence as non-chimeric.
tryRC	Whether to try reverse complement when searching for chimeras.

Not all the pipelines resulted in an ASV table after using DADA2 package. In total, only 308 out of 700 pipelines used in this study resulted in a valid ASV table, while the remaining 392 pipelines failed to produce an output. The potential reasons for this are not part of the study, however, a small exploratory analysis was conducted to gain some insights into the possibilities. In addition to this information, an annotation table is also available in .csv format for each pipeline

providing information about the taxonomic system of rRNA sequences founded in each ASV table. Here is a list of files in the dataset:

- 700 ASV tables in .csv format, each for a single pipeline
- 700 annotation tables in .csv format, each for a single pipeline
- 1 parameters table in .csv format, set of parameter values for each pipeline

3.2 Procedure

To answer the main question, a network graph of pipelines needs to be generated. By overlaying information about microbes and parameters over this network graph, potential patterns and insights could be revealed. The network graph consists of 308 nodes, each representing one of the pipelines with an ASV table as outcome. An edge, connects a pair of nodes represents the similarity between ASV tables produced by those nodes (pipelines). For finding these edges, the distance between each pair of pipelines must be calculated using proper metric, this requires the presence of a common axis in the dataset space; in other words, similar columns must be present in different ASV tables.

After checking for common rRNA sequences among all ASV tables, it became clear that all of the sequences are unique. Therefore, it was not possible to calculate distances using the primary ASV tables because there wasn't any common axis among them. To overcome this issue, a preprocessing step was performed on the dataset to create appropriate ASV tables with similar columns across different pipelines. Section 3.4 explains this process in detail.

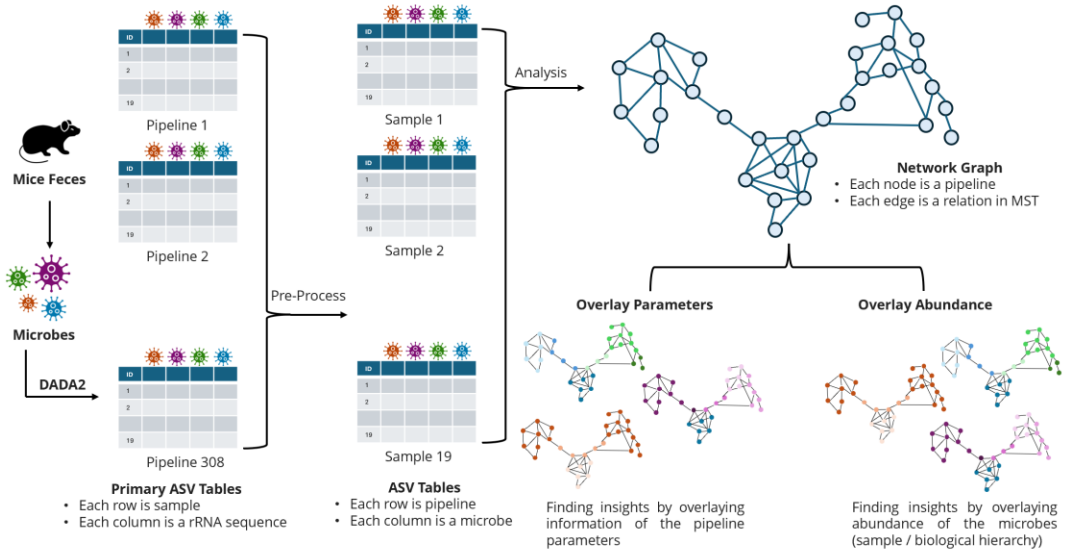


Figure 2 Procedure of the study

After calculating the distance between pairs of pipelines, a distance matrix is generated for construction of the network graph. In this study, Minimum Spanning Trees (MST) are used for representation of the network graphs. For each sample a tree was generated and combination of all of these MSTs resulted in the main network graph. Section 3.5.3 explains how different types of network graphs were developed for analyzing the dataset.

At this point, a network graph is available for visual analytics. By overlaying information about the parameters (values of a single parameter for different pipelines) and also information of the abundances from different samples, potential patterns and insights can be identified.

By comparing patterns founded using pipeline parameters with patterns reveled in using abundance information, effect of parameters and their values will become clear. This entire process is illustrated in Figure 2.

3.3 Software

For preprocessing of the data, conducting exploratory data analysis, and calculating the network graph matrix (distance matrix), Python programming language and several libraries were used. Visualization of the graphs and graph statistics were performed using Gephi (Bastian 2019).

- Python 3.10.2
    - Matplotlib 3.5.0
    - Scipy 1.9.0
    - Pandas 1.5.5
    - Numpy 1.22.3
    - Seaborn 0.12.4
    -
- Gephi 0.10.1 202301172018

3.4 Preprocessing of ASV tables

Before beginning with topological data analysis (TDA), preprocessing of the existing ASV tables is necessary to transform them into suitable inputs for TDA. An initial issue, briefly mentioned in Section 3.2, was the lack of a common axis for calculating distance between different ASV tables. One proposed solution was to use BLAST to find nearly identical sequences and group them accordingly, however, each pipeline has an annotation table containing taxonomic system

information for the sequences found within it. By utilizing this annotation table, each rRNA sequence can be replaced with its corresponding taxonomic system.

It's important to note that not all rRNA sequences have a recognized corresponding category in taxonomic system in the annotation table for in Kingdom level and deeper, as some are still unknown in existing biological databases. Across different pipelines, this lack of information varies from 3% to 60%, with a median loss of approximately 21%. In this study, all unknown sequences are excluded from the analysis process. Figure 3 illustrates the distribution of available data among different pipelines.

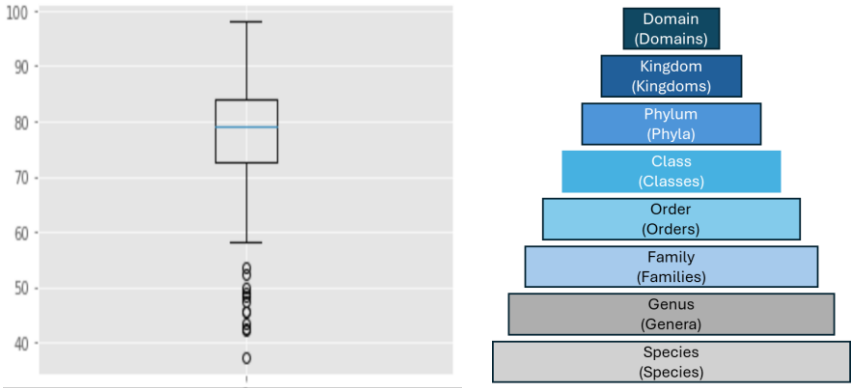


Figure 3 Boxplot for loss of information (left) Biological hierarchy (right)

Each annotation table provides information at 7 hierarchical levels: Kingdom, Phylum, Class, Order, Family, Genus, and Species, as shown in Figure 3. Exploring deeper into this hierarchy increases the number of instances, but at the same time increases loss of the information because not all sequences are labeled at deeper levels. Therefore, it is crucial to decide at which hierarchical level the rRNA sequences are going to be replaced with equivalent biological information. This decision requires balancing the trade-off between increasing instances (more axes in the data space for distance calculation) and minimizing the loss of information. To facilitate this decision, boxplots of available information were developed across different levels, as illustrated in Figure 4.

The top row of Figure 4 shows the amount of available data in the taxonomic system for each level among all pipelines, and the bottom row shows the number of available instances in each ASV table at that specific level. It is clear that all sequences have information at the Kingdom, Phylum, and Class levels. The loss of information starts at the Order level but is very minor (median almost 100%). The available information has a median of 91% at the Family level, 55% at the Genus

level, and a bit more than 6% at the Species level. Following this explanation, Species level is not a good choice for replacing rRNA sequences. Considering the number of instances at each level, boxplots show a median of 1 instance for Kingdom, 9 instances for Phylum, 12 instances for Class, 28 instances for Order, 36 instances for Family, and 52 instances for Genus. Considering these values, the Family level was chosen for creating the final ASV tables due to a good average number of instances (38) and a lower average loss of the information equal to 9%.

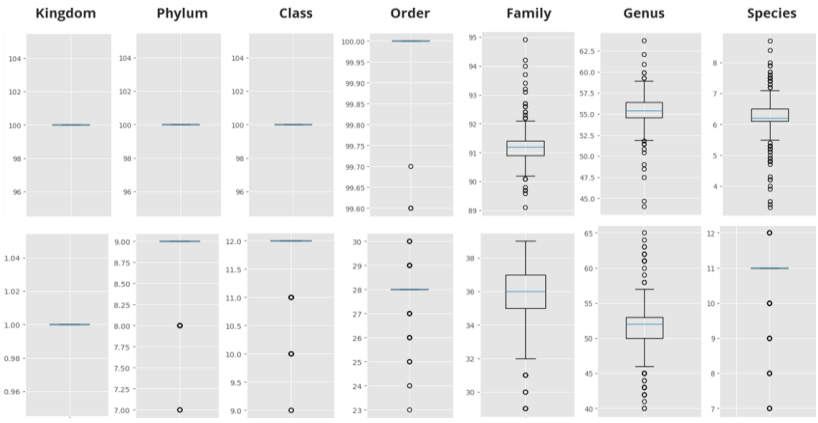


Figure 4 Loss of information (top) Number of instances in hierarchy (bottom)

The next step in the process is to change the shape of ASV tables from table per pipeline into table per sample for two reasons. First, study is querying for the effect of the parameters over ASV tables, so by checking this effect over different samples we can look for consistent patterns among them as the output of the similar samples should be close to each other. Secondly, to calculate the distance matrix, we need a table of abundance values of different pipelines for each axis in dataset space and in this case each access will be a Family. So the appropriate form of ASV table includes pipelines as rows, families as columns and abundance of each family in specified pipeline as cell value. It is possible that multi rRNA sequences belong to a similar Family, in this case, sum of the abundance of sequences belonging to that family is used.

The output of the preprocessing step includes 19 ASV tables, each corresponding to one of the samples in the dataset. In total, 35 unique families were identified, some of which are found only in certain subsets of samples. This count is lower than the median observed in Figure 4, primarily because it considers all 19 samples plus one Mock sample. Certain families appear exclusively in the Mock sample and not in the mouse samples. Removing the Mock sample leaves us with these 35 families, which represent columns of the final ASV tables. Each row in these tables corresponds to a pipeline, and the cell value is sum of the abundances for that family. These tables serve as input for the Topological Data Analysis (TDA) process.

To clarify how pipelines are identified, each pipeline is assigned a unique ID. This ID is sourced from the parameters .csv file and ranges from 1 to 700. As previously mentioned, only 308 pipelines result in ASV tables. Therefore, there will be 308 IDs within this range, but not necessarily spanning from 1 to 308.

3.5 Data visualization and Topological Data Analysis (TDA)

Data visualization plays a crucial role in biology by transforming abstract data into visual representations that are easier to interpret and analyze. Techniques such as heatmaps, scatter plots, and network diagrams allow researchers to explore relationships between genes, proteins, and metabolites across different experimental conditions. For instance, tools like Cytoscape enable the visualization of molecular networks, revealing interactions between biomolecules and their roles in cellular processes (Shannon et al., 2003). Moreover, advancements in interactive and three-dimensional visualization techniques enhance the exploration of spatial relationships in biological structures, such as protein folding and cellular localization, offering insights that aid in drug design and molecular engineering (Meyer et al., 2021).

Dimensionality reduction techniques, such as principal component analysis (PCA) and t-distributed stochastic neighbor embedding t-SNE (Maaten, 2014), are pivotal in reducing the complexity of biological datasets while preserving essential features. PCA identifies orthogonal components that explain the variance in data, making it useful for clustering similar samples or identifying outliers based on gene expression profiles (Wold et al., 1987). On the other hand, t-SNE is effective in visualizing high-dimensional data in lower-dimensional space, revealing clusters and patterns that may correspond to distinct biological states or cell types (Maaten and Hinton, 2008). These techniques are widely applied in single-cell RNA sequencing to explore cellular heterogeneity, identify rare cell populations, and understand developmental trajectories in tissues (Stuart and Satija, 2019).

Graph-based approaches have also gained prominence in biological research for modeling complex interactions and networks. Graph theory enables the representation of biological entities (nodes) and their relationships (edges) in various contexts, such as protein-protein interaction networks, metabolic pathways, and gene regulatory networks (Barabasi and Oltvai, 2004). Algorithms like community detection and centrality measures help identify modules of highly connected nodes or key regulatory elements within these networks, offering insights into disease mechanisms and potential therapeutic targets (Newman, 2006).

3.5.1 Minimum Spanning Trees (MSTs)

Topological data analysis (TDA) has revolutionized the study of biological systems by employing techniques such as minimum spanning trees (MST, Prime 1957) to unveil underlying structures and relationships in complex datasets. MSTs offer a straightforward yet powerful method to identify the most critical connections and hierarchies within biological networks, ranging from gene regulatory networks to ecological interactions (Gao et al., 2020). By emphasizing the shortest paths between nodes while connecting all vertices with minimal total edge weights, MSTs facilitate the identification of central nodes and clusters that play pivotal roles in biological processes and evolutionary dynamics. This approach has been instrumental in elucidating functional modules in protein interaction networks, identifying key genes in disease pathways, and understanding the resilience of ecosystems to environmental changes (Lee et al., 2023; Wang et al., 2022).

Moreover, MSTs contribute significantly to data visualization techniques in biology, offering intuitive representations of network structures and connectivity patterns. Visualizing MSTs as simplified graphs helps researchers to intuitively grasp the hierarchical organization and spatial relationships within biological systems, aiding in the identification of critical nodes, bottlenecks, and potential targets for intervention or study (Adams et al., 2022). This visualization approach not only enhances exploratory data analysis but also facilitates interdisciplinary collaborations by making complex biological phenomena accessible and interpretable across different fields of study.

Innovative applications of MSTs in conjunction with machine learning algorithms further extend their utility in predictive modeling and decision support in biology. Integrating MST-based insights with predictive analytics enables researchers to forecast disease trajectories, optimize drug discovery pipelines, and model ecological dynamics with greater accuracy and efficiency (Chen et al., 2021). As TDA continues to evolve, leveraging MSTs and similar techniques promises to unlock deeper insights into biological complexity, driving forward discoveries and innovations across diverse domains of biological research.

3.5.2 Bray- Curtis metric

The Bray-Curtis dissimilarity metric (Bray et al. 1957) is widely employed in biology to quantify compositional differences between samples, particularly in ecological and microbiological studies. This metric assesses dissimilarity based on species abundances or composition across multiple samples. It ranges from 0 (complete similarity) to 1 (complete dissimilarity), making it valuable for comparing community structures. To calculate Bray-Curtis dissimilarity between two

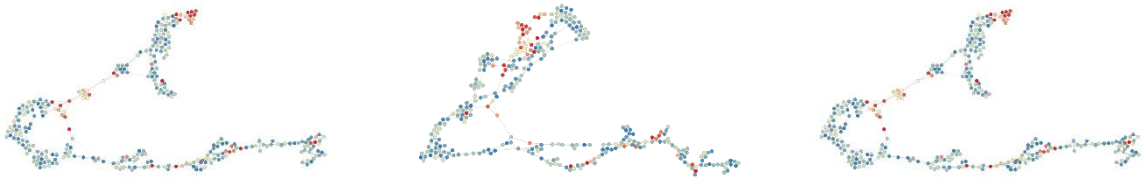
samples, one computes the sum of absolute differences in species abundances divided by the sum of total abundances in both samples. Mathematically, it is expressed as:

$$Bray - Curtis\ dissimilarity = \frac{\sum |A_i - B_i|}{\sum (A_i + B_i)}$$

where A_i and B_i represent the abundances of species (i) in samples A and B, respectively. This metric is preferred in ecology because it is sensitive to both species presence and abundance changes, providing a robust measure of community dissimilarity. In biological research, Bray-Curtis dissimilarity finds application in various contexts, including studying biodiversity patterns across habitats, comparing microbial community structures in different environments, and analyzing shifts in species composition over time. Its versatility and ability to handle large datasets make it a cornerstone in ecological research, aiding in understanding community dynamics, species interactions, and ecosystem responses to environmental changes (Krebs, 1999; Anderson, 2001; Legendre & Legendre, 2012).

3.5.3 Network graphs of pipelines

To generate these graphs, distance matrices were calculated using the SciPy library with Bray-Curtis metric. Subsequently, the Minimum Spanning Tree (MST) function from the same package was used to find paths connecting all pipelines (nodes) with a single edge, ensuring each node is visited exactly once. Thus, each graph consists of 308 nodes and 307 edges. It's important to note that MST graphs are not unique; other graphs with the same path length (distances in this case) may exist. Small variations in the distance matrix can result in different tree structures. Figure 5 provides examples of these trees for samples 1, 10, and 19. These graphs were created using ForceAtlas2 algorithm (Jacomy 2014) in Gephi. Colors used in these network graphs correspond to pipeline IDs and were only used for identification and comparison purposes among different MSTs.



Sample 1 (Day 0) Sample 10 (Day 148) Sample 19 (Day 9)

Figure 5 Example of MSTs for different samples

If we use a pipeline on similar samples, we expect to see a more or less similar network of graphs created by the distance matrices of those samples. By overlaying these individual MSTs on top of each other, a graph is created with the same number of nodes (308) but likely more edges, as some edges may only exist in the MST of one sample and not in others. Figure 6 shows a graph resulting from this process, visualized by ForceAtlas 2 algorithm in Gephi. The weight of each edge is equal to the sum of its presence in different MSTs, with a maximum weight equal to 19, equivalent to the number of samples.

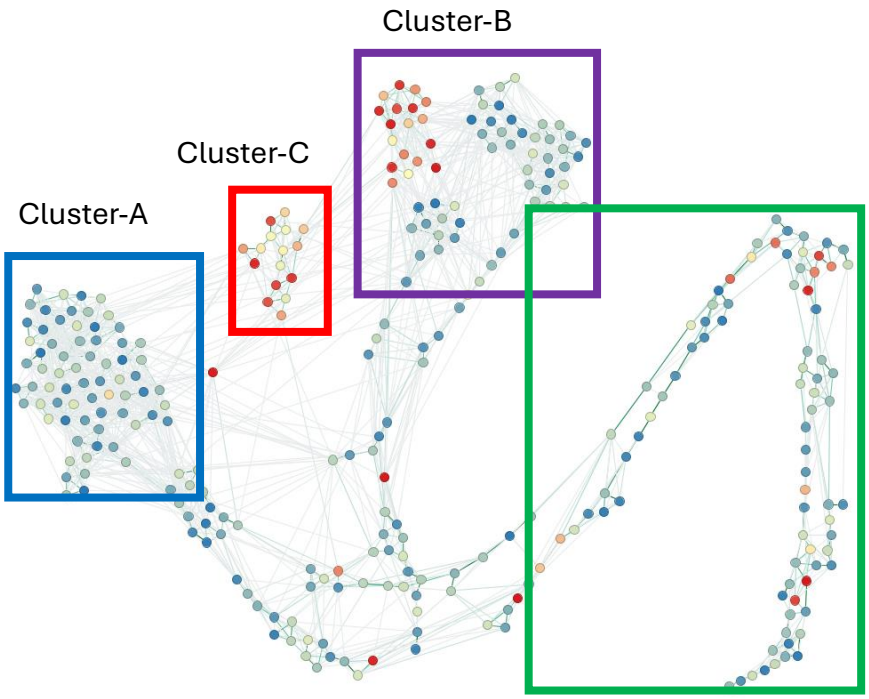


Figure 6 Network graph of pipelines

Clearly, the number of edges is greater, totaling 1698. The general form of the graph consists of a tail on the right side and a main body on the left. The main body itself comprises two clusters (cluster-A in blue rectangle and cluster-B in purple rectangle) and a smaller middle cluster (cluster-C), connected to the tail by two paths. Low weight indicates that an edge is not persistent across different samples, while higher weights correspond to edges consistently present in different MSTs. To examine the general shape and persistence of the edges, a filter is applied to this graph, each time filtering out all the edges with a weight equal to or higher than a minimum value.

Figure 7 shows how this graph changes by increasing the minimum threshold from 1 to 2. The general shape of the network graph remains the same; however, the number of edges is reduced to 821, or almost half of the original graph. This indicates that half of the connections appeared only once among different MSTs.

This network graph was considered the main graph for visual study and overlaying parameter/microbe information because it retains the general shape and connections while focusing on more persistent connections. Results derived from this graph are expected to be more robust and reliable.

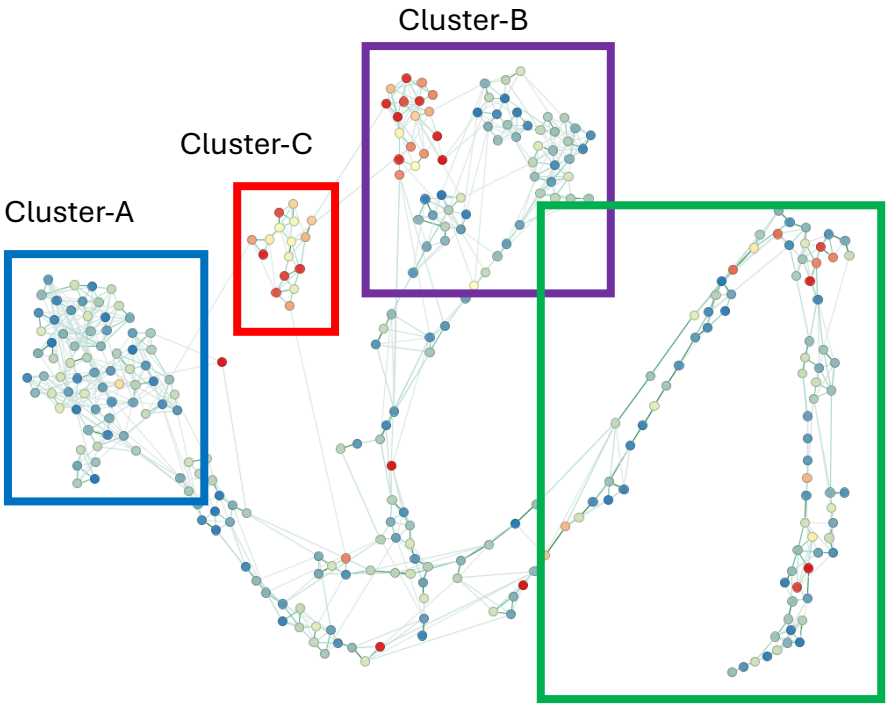


Figure 7 Network graph of pipelines (edges with weight ≥2)

3.5.4 Overlaying Parameters/Microbes Information

The final step is to investigate patterns visually by overlaying information about parameter values and microbes (families). Adjacent nodes in the network indicate that those nodes have similar ASV tables, so we hope to find corresponding patterns in the parameters . Each parameter's values will be overlaid on this network graph and colorized to reflect patterns, allowing to see if any patterns exist over clusters of pipelines.

Similarly, to see the effect of pipelines on abundance values, for each sample the abundance values of families (per family) will be overlaid to look for patterns.

4 Results

4.1 Exploratory data analysis

An statistical summary representation of ASV tables and pipeline parameters is necessary for the analysis of the pipeline network graph and making conclusions. In both cases, boxplots were used to visualize the distribution of the values across different families and pipelines. In addition to that, all the values were shown as jittered points over each box plot.

4.1.1 Pipeline parameters

As previously noted in section 3.1, only 308 out of 700 pipelines, each configured with different set of parameters, successfully generated ASV tables using DADA2. Figure 8 illustrates the distribution of 18 numerical parameters, colorized by the outcome. Red points signify the 392 pipelines that yielded no outcome, while blue points represent the 308 pipelines that produced an ASV table.

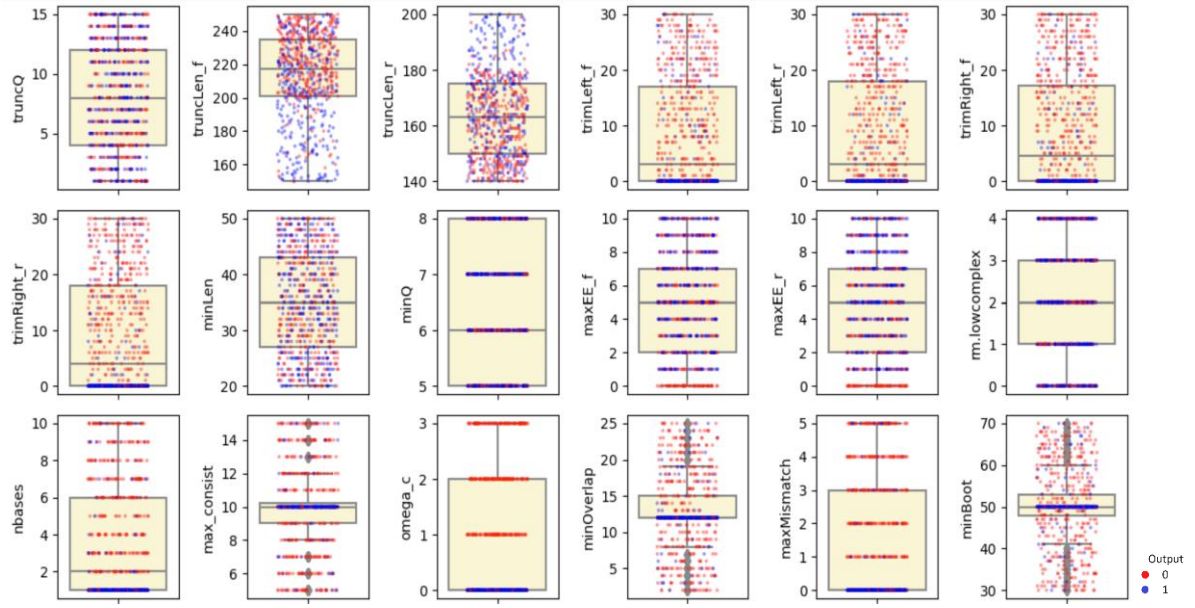


Figure 8 Boxplots of numerical parameters

For some parameters, most pipelines with outcomes are concentrated around a specific value among the other values. By controlling the ratio of pipelines without outcomes but with the same parameter value, the influence of the parameter on producing results can be examined. Table 3 shows a list of those parameters, value with high concentration, total number of pipelines, number of pipelines with outcome and number of pipelines without outcome for that value. Last column additionally added to consider the ratio of pipelines with other values but resulted in an outcome, so this column is equal to difference of 4th column and 308.

Table 3 Parameters with concentrated values resulted in an outcome (ASV table)

Parameter	Value	Total	With Outcome	Without Outcome	With Other Values & Outcome (out of 308)
trimLeft_f	0	316	255 (80.70%)	61 (19.30%)	53 (17.21%)
trimLeft _r	0	311	253 (81.35%)	58 (18.65%)	55 (17.86%)
trimRight_f	0	307	253 (82.41%)	54 (17.59%)	55 (17.86%)
trimRight _r	0	307	253 (82.41%)	54 (17.59%)	55 (17.86%)
nbases	10E+8	343	263 (76.68%)	80 (23.32%)	45 (14.61%)
Max_consist	10	334	257 (76.95%)	77 (23.05%)	51 (16.56%)
omega_c	0	394	308 (78.17%)	86 (21.83%)	0 (0.00%)
minOverlap	12	306	255 (83.33%)	51 (16.67%)	53 (17.21%)
maxMismatch	0	368	264 (71.74%)	104 (28.26%)	44 (14.29%)
minBoot	50	309	253 (81.88%)	56 (18.12%)	55 (17.86%)

By looking at last column, almost 83% of pipelines with outcome have a value equal to 2nd column, except for "omega_c" for which all the pipelines with outcome having a value equal to 0. Additionally, by looking at 4th and 5th column, 77% to 83% of pipelines with specified values are resulted in ASV tables for all parameters, for "maxMismatch" this ratio is 71.74%.

These parameters could be the potential parameters for answering to the question "What is the relation between set of parameters and presence of an ASV table as outcome? ", because violation from specified values in 2nd column mostly resulted in a pipeline with no outcome, however, for making solid conclusions permutation of values for these parameters and possible correlation between them must be investigated.

Interestingly, majority of the pipelines without an outcome have "truncLen_f" with a value more than 200 and "trunLen_r" is mostly less than 180 for these pipelines. For pipelines with an outcome "truncLen_r" spreads over a range from 150 to 250 and "truncLen_r" over 140 to 200. Deeper investigation in pipelines with outcome for these parameters could be resulted in finding some patterns.

Remaining parameters ("trunQ", "truncLen_f", "truncLen_r", "minLen", "minQ", "maxEE_f", "maxEE_r", and "rm.lowcomplex") having almost uniformly distributed values among different pipelines with an outcome.

Figure 9 illustrates distribution of values for categorical parameters. For four Boolean parameters ("randomize", "selfConsist", "pool", and "tryRC"), the majority of pipelines with outcomes have a False value (represented as 0 in boxplots). Only

about 10% of pipelines with these parameters set to True result in an outcome. In contrast, for pipelines without outcomes, the distribution is more balanced, with approximately half having False and half having True values. For “rm.phix” the behavior between piplines with and without an outcome is almost the same. Majority of pipelines with per-sampled and pooled values for “chim.method” don’t have an outcome, in contrast, for 64.37% of pipelines with “chim.methid” equal to consensus an outcome is available.

Similar to what mentioned about ten parameters in Table 3, all the categorical parameters except for “rm.phix” could be potentially used to answering “What is the relation between set of parameters and presence of an ASV table as outcome?”.

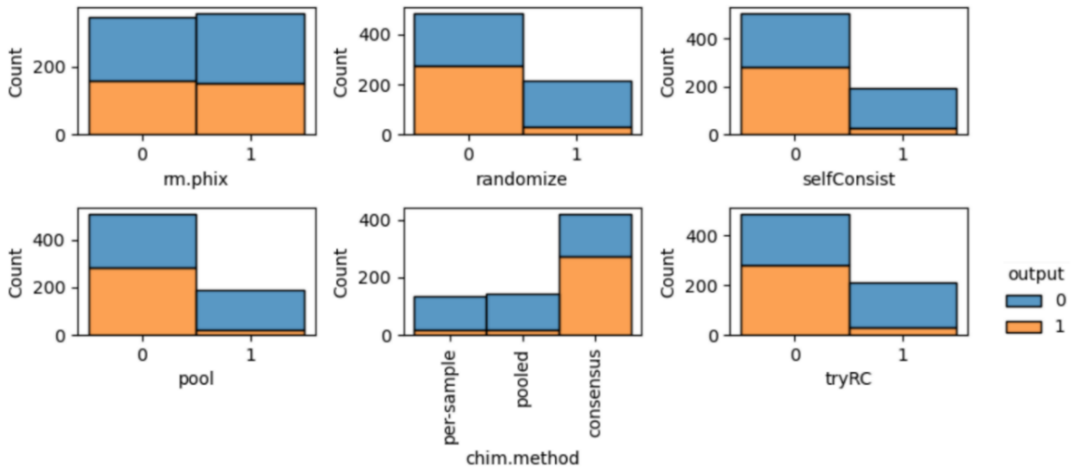


Figure 9 Boxplots of categorical parameters

4.1.2 ASV tables

For ASV tables, boxplots of abundances were generated across different families for all pipelines, in order to examine insights out of the network graph of pipelines easier and more precisely (all boxplots in appendix 8.1). Based on these box plots families were categorized in four groups, each category considers a certain maximum threshold for the value of abundance in ASV tables. Table 4 summarizes these categories and a list of families belonging to them.

Table 4 List of families for different abundance threshholds

Threshold <5	Threshold <50	Threshold <500	Threshold >500

Actinomycetaceae	Akkermansiaceae	Acholeplasmataceae	Bacteroidaceae
Anaerofustaceae	Butyricicoccaceae	Anaerovoracaceae	Lachnospiraceae
Atopobiaceae	Christensenellaceae	Bifidobacteriaceae	Lactobacillaceae
Bacillaceae	Clostridiaceae*	Erysipelotrichaceae	Muribaculaceae
Defluviitaleaceae	Eggerthellaceae	Oscillospiraceae*	
Deinococcaceae	Enterobacteriaceae	Peptococcaceae	
Erysipelatoclostridiaceae	Mitochondria	Rikenellaceae	
Listeriaceae	Pseudomonadaceae	Ruminococcaceae	
Monoglobaceae	Saccharimonadaceae		
Moraxellaceae	[Eubacterium]		
Neisseriaceae			
Staphylococcaceae			
Streptococcaceae			

*Families with star only have a single value more than threshold for one of the samples

In microbial ecology, determining if a small abundance value in an ASV table represents a true biological presence or just noise is a concern. Different thresholds have been proposed to address this issue. Callahan et al. (2016) suggests threshold of one, while Needham et al. (2017) propose values of two or three. In this study, the first category is defined with a threshold of 5 to serve as a reference for future analysis in case of unexpected results in the patterns found in network graph of pipelines. Categories with higher threshold were designated to provide additional insights, including considerations for longitudinal effects or sudden family appearances, potentially coming from factors like disease although pipelines expected to behave independent to these effects.

In addition to the value of the cell in the ASV table for each family, another important factor is the presence of the microbe in different samples. In Table 5, all families are colorized based on their number of occurrences in various samples. Twelve families out of thirty-five exist in all samples, while nine families appear less than four times in different samples. Among the families with an occurrence threshold of less than five, “Erysipelotrichaceae,” “Monoglobaceae,” and “Streptococcaceae” appear frequently (more than ten times) in different samples. Therefore, any pattern found in the network graph of pipelines could potentially be used to examine the presence of these families.

The table also shows that the number of families in different samples varies between 19 and 29, with an average of 22.73 (almost 23) families per sample. A family like “Akkermansiaceae” only shows up in the beginning of the study (days 0 to 9), and another family like “Neisseriaceae” only shoes up at the middle of the study (days 141 to 150).

Table 5 Presence of the families among different samples (during time)

Sample	1	2	13	14	15	16	17	18	19	3	4	5	6	7	8	9	10	11	12	
Day	0	1	2	3	5	6	7	8	9	141	142	143	144	145	146	147	148	149	150	Total
[Eubacterium	1	0	1	0	0	0	0	0	0	1	0	0	0	1	0	0	1	1	1	7
Acholeplasmataceae	1	1	1	1	1	1	1	1	1	0	1	0	0	1	1	1	1	1	0	15
Actinomycetaceae	0	0	0	0	0	0	0	0	0	1	1	0	0	1	0	0	0	0	0	3
Akkermansiaceae	1	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	3
Anaerofustaceae	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	1	0	0	3
Anaerovoracaceae	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	19
Atopobiaceae	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	1	1	0	0	4
Bacillaceae	0	0	0	0	0	0	0	0	0	1	1	0	0	1	0	0	0	0	0	3
Bacteroidaceae	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	19
Bifidobacteriaceae	1	0	1	1	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	13
Butyricicoccaceae	1	1	1	0	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	17
Christensenellaceae	1	0	1	0	0	1	0	1	1	1	1	1	1	1	0	1	1	1	1	14
Clostridiaceae	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	19
Defluviitaleaceae	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	0	1	7
Deinococcaceae	0	0	0	0	0	0	0	0	0	1	1	1	0	1	0	0	0	0	0	4
Eggerthellaceae	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	19
Enterobacteriaceae	1	1	1	1	1	1	1	1	1	1	0	0	1	1	1	1	1	1	1	17
Erysipelatoclostridiaceae	1	0	0	1	0	0	1	1	1	1	1	0	0	0	1	1	1	0	0	10
Erysipelotrichaceae	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	18
Lachnospiraceae	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	19
Lactobacillaceae	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	19
Listeriaceae	0	0	0	0	0	1	0	0	0	0	1	1	0	0	0	0	0	0	0	3
Mitochondria	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	1	0	0	10
Monoglobaceae	1	1	1	1	0	0	1	0	0	1	1	0	0	1	1	1	1	1	0	12
Moraxellaceae	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	2
Muribaculaceae	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	19
Neisseriaceae	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0	0	0	0	0	3
Oscillospiraceae	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	19
Peptococcaceae	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	19
Pseudomonadaceae	1	1	1	1	1	1	1	1	1	0	1	1	1	1	0	0	1	1	1	16
Rikenellaceae	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	19
Ruminococcaceae	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	19
Saccharimonadaceae	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	19
Staphylococcaceae	0	0	1	0	0	0	0	1	0	1	1	0	1	1	0	0	0	0	0	6
Streptococcaceae	1	1	1	1	1	1	1	1	1	1	0	1	0	1	0	0	1	1	0	14
Total	25	19	25	21	19	22	22	22	21	26	25	21	23	29	20	23	27	22	20	

17 ≤ Occurance ≤ 19
10 ≤ Occurance ≤ 16
5 ≤ Occurance ≤ 9
2 ≤ Occurance ≤ 4

0 Family exist in the sample
1 Family doesn't exist in the sample

Families colored in green are consistent across all samples over time, being present in at least seventeen out of nineteen samples. Figure 10 shows a boxplot of these families belonging to categories with thresholds of 500 and 5000 (3rd and 4th columns in Table 4) spread over different pipelines. For seven families, a peak value is present on day 2 in these plots (D002-S12). Similarly, some peaks exist on the right side for families "Bacteroidaceae," "Lactobacillaceae," "Muribaculaceae," and "Peptococcaceae" on days 148, 149, and 150. Although this should not be due to related to values of parameters in pipelines, in the next section we will check by looking for a similar color gradient in these samples compare with other samples.

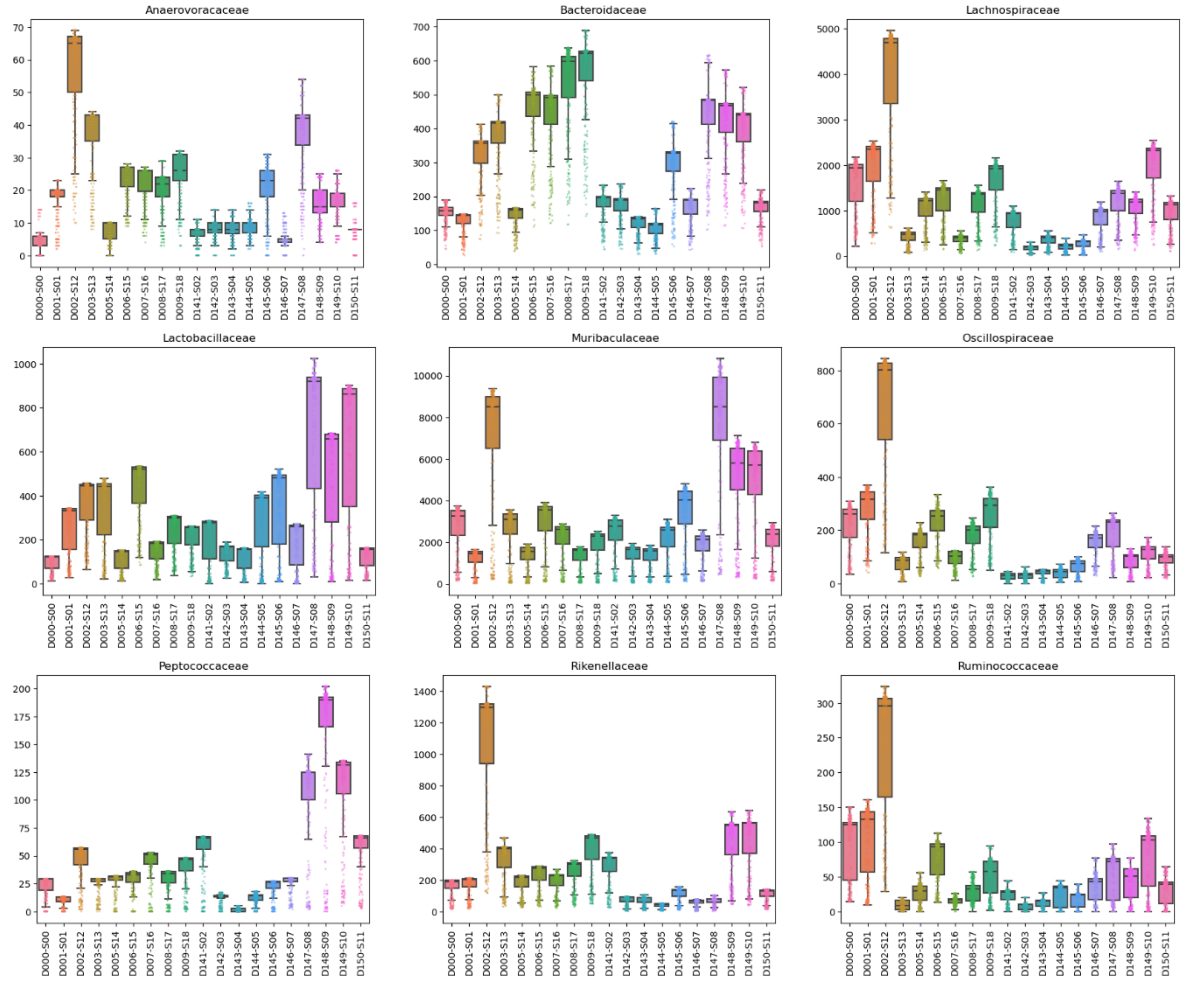


Figure 10 Boxplots of abundances of persistent families

4.2 Network graph analysis

Before looking for patterns, a statistical summary of the graph were provided to figure out the important topological nodes and structures. Figure 11 illustrates histogram of node's degree of the network graph of pipelines. Smaller degree refers to persistent pipelines, the reason behind this is a node with similar connections in MST of samples will result in a node with the same edges but with a high value for weight, on the other hand if a node connected differently to other nodes among different MSTs, this node will turn into a node with more connections but with lower weight for each. Histogram is close to normal distribution with a skewness to the right. Average degree is 11.026 and 24 nodes have a degree equal or less than 4 which are the consistent ones (connected to the same node in different MSTs of different samples) and 25 nodes have a degree equal or more than 19 which are unstable (connected to different nodes in different MSTs of different samples)

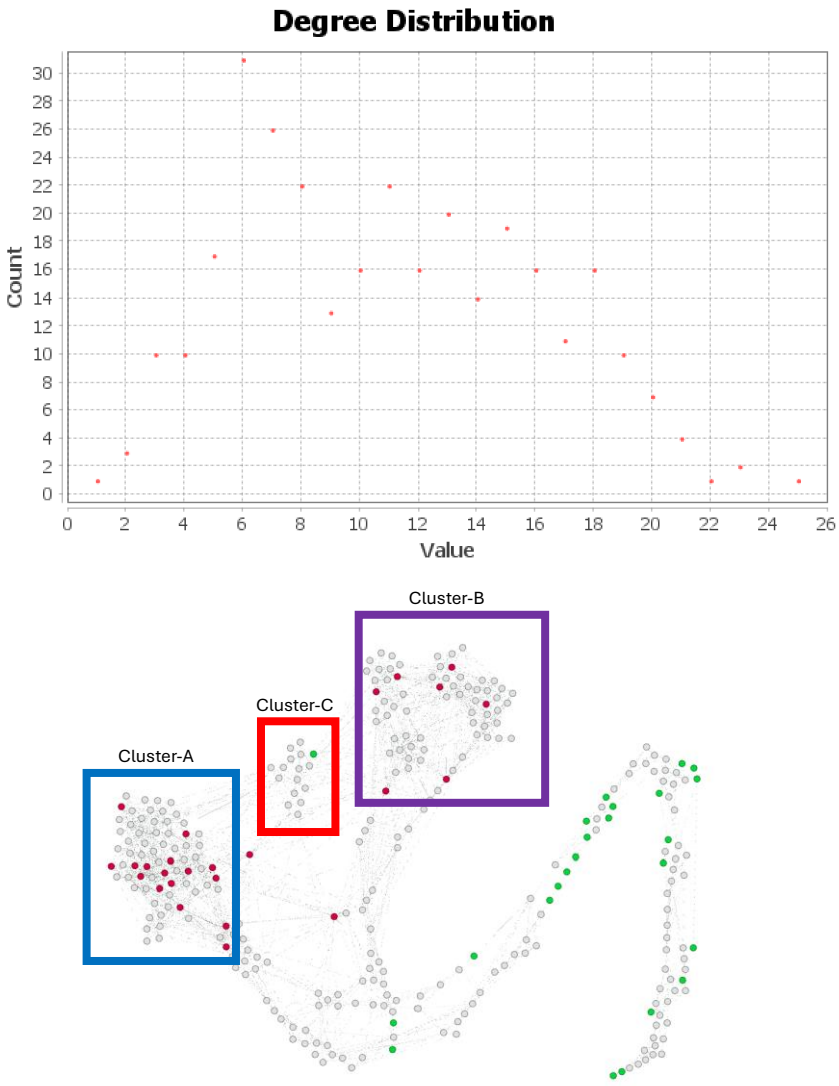


Figure 11 Degree histogram (top) Potential structures in the network (bottom)

Red points are located in clusters A and B (Figure 11), presenting an unstable connection between pair of nodes. To check for the existence of these two clusters, a filter was applied to the graph for nodes with degrees between 13 and 25 (inclusive). All 121 nodes filtered in this range are located in clusters A and B in the body of the graph. The edges connecting these nodes show sparse connections, which is expected. To examine inter-clusters connectivity, edges were filtered for weights equal to or greater than 2. Figure 12 shows that clusters A and B become separated by applying this filter, indicating that the only possible consistent connectivity between these two clusters are through the middle cluster, C.

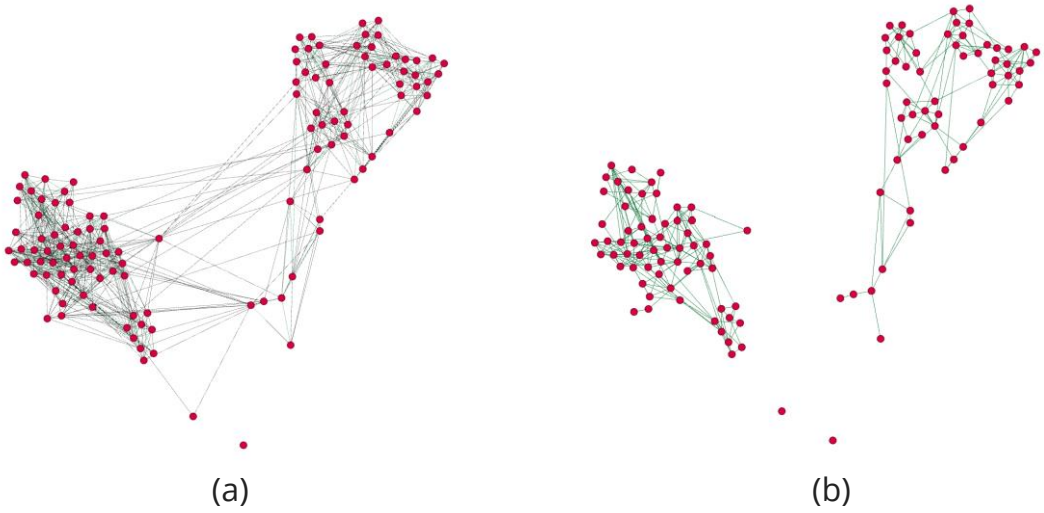


Figure 12 (a) Clusters A and B (b) Clusters A and B (edge weight ≥2)

After adding cluster C to the graph and examining inter-cluster connectivity, for edges with weights equal to or greater than 3, cluster C is only connected to cluster B (see Figure 13), and for edges with weights equal to or greater than 4, all connections between clusters A and B disappear illustrating a stronger connectivity between clusters C and B in compare with clusters C and A (two edges with weight equal to 4 versus one edge with weight equal to 3)

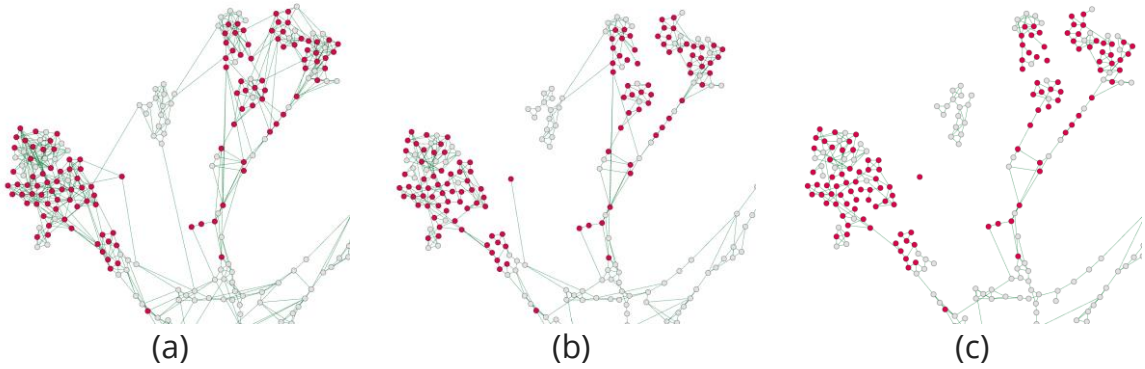


Figure 13 Clusters A,B and C (a) edge weight ≥2 (b) edge weight ≥3 (c) edge weight ≥4

Finally, two separate paths from the main body to the tails is present in the graph with edges filtered for weight equal or greater than 2 (Figure 14). In addition to clusters in the main body and tail, these two paths could be considered as potential structures in finding patterns for overlaid parameters / microbes.

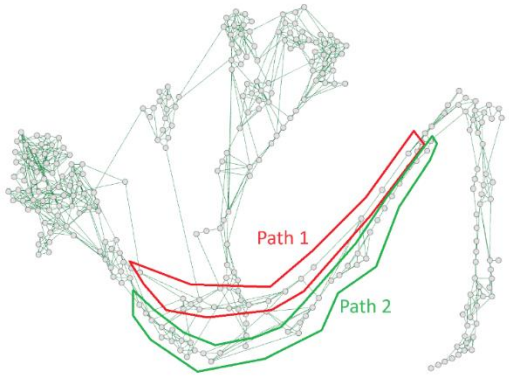


Figure 14 Network structures - Path 1 and Path 2

4.2.1 Overlaying parameters information

To find any effect of the pipeline parameters on abundance tables (ASV tables), the first step is to explore parameters overlaid on top of the network graph of the pipelines, focusing on the identified structures from the previous section (clusters A, B, and C, tail, path 1, and path 2). For this purpose, all 24 parameters are colorized based on their values on top of the network graph. From Table 5, we know that 10 numerical parameters are mainly concentrated around a certain value in the network graph of pipelines. For these parameters, we are only checking if any pattern exists for other values. Except for "omega_c," which is equal to 0 for all pipelines, other parameters are illustrated in Figure 16. This figure clearly shows that most of the pipelines with values different from the mentioned value in the table are concentrated in cluster C and part of cluster B. The stronger connection between these two clusters found in Figure 13 is mostly due to the similarity in the values of these parameters. Several pipelines with different values are also spread over the network graph in the connecting path of the clusters to the tail and the tail itself. These pipelines are common across all parameters, meaning values other than what mentioned in Table 3, are gathered in these pipelines. This pattern is labeled as pattern-1.

The color pallet which used for colorizing of the graphs (Figure 15) in this section and next section is considering the minimum to maximum values within each parameter and does not reflect a constant values everywhere. Red is always at the minimum side of the range and blue refers to the maximum values. For range of values refer to 8.18.1.

Figure 15 Default color palette

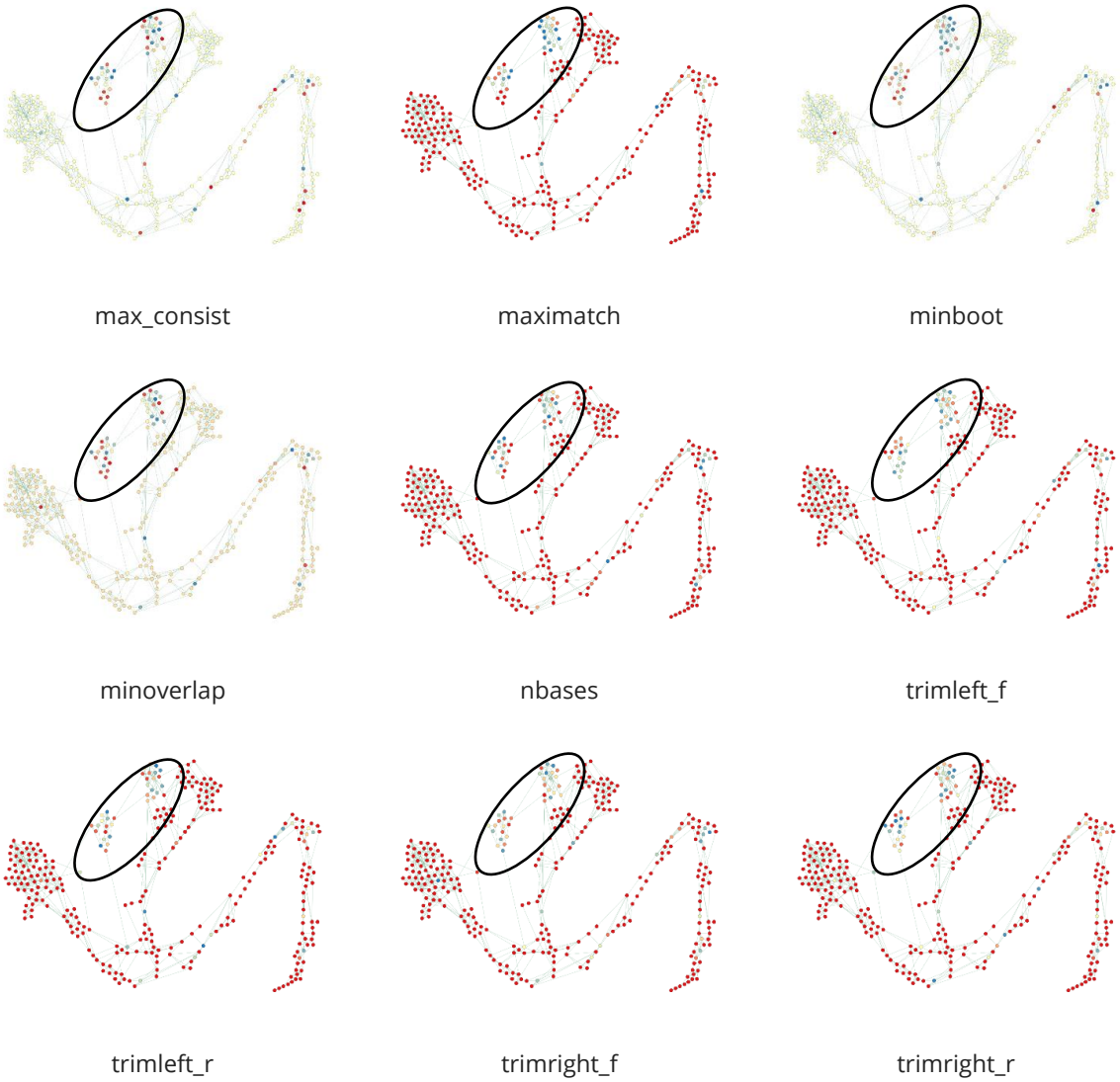


Figure 16 Overlaid numerical parameters on the network graph – Pattern-1

Categorical parameters with similar situation (mostly equal to False) are also following the same pattern. Figure 17 shows these parameters overlaid on the network graph.

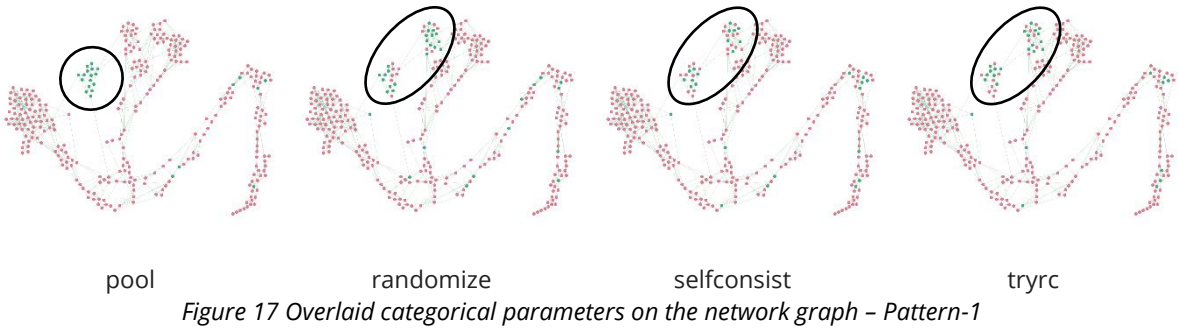


Figure 17 Overlaid categorical parameters on the network graph – Pattern-1

Among the remaining parameter, four of them distributed over the whole graph without notifying a considerable pattern or concentration of values (colors) in a specific location.“rm.phix” is categorical (Boolean) and “rm.lowercomplex”, “minlen” and “minq” are numerical parameters. (Figure 18)

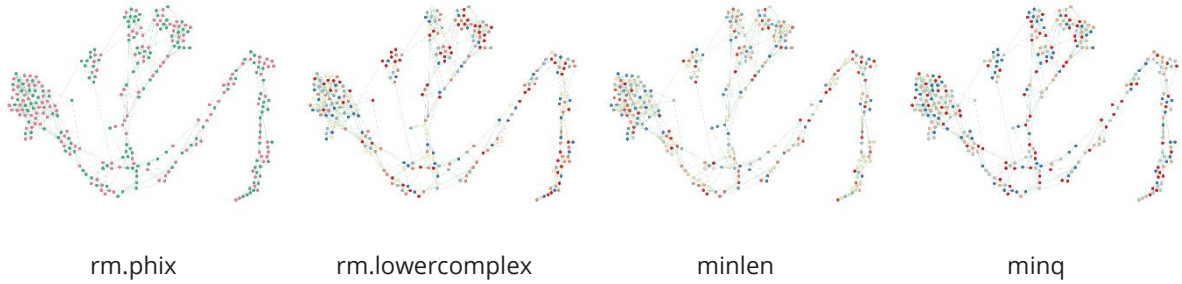


Figure 18 Overlaid parameters on the network graph (distributed uniformly)

By visualizing the last five parameters, different pattern revealed for each of them. For “maxee_f” lower values are concentrated in path 2, however, for maxee_r lower values are concentrated in path 1. In “trunclen_r “ lower values located in clusters C and B and also half of the tail closer to the main body, while higher values are in cluster A and end of the tail.

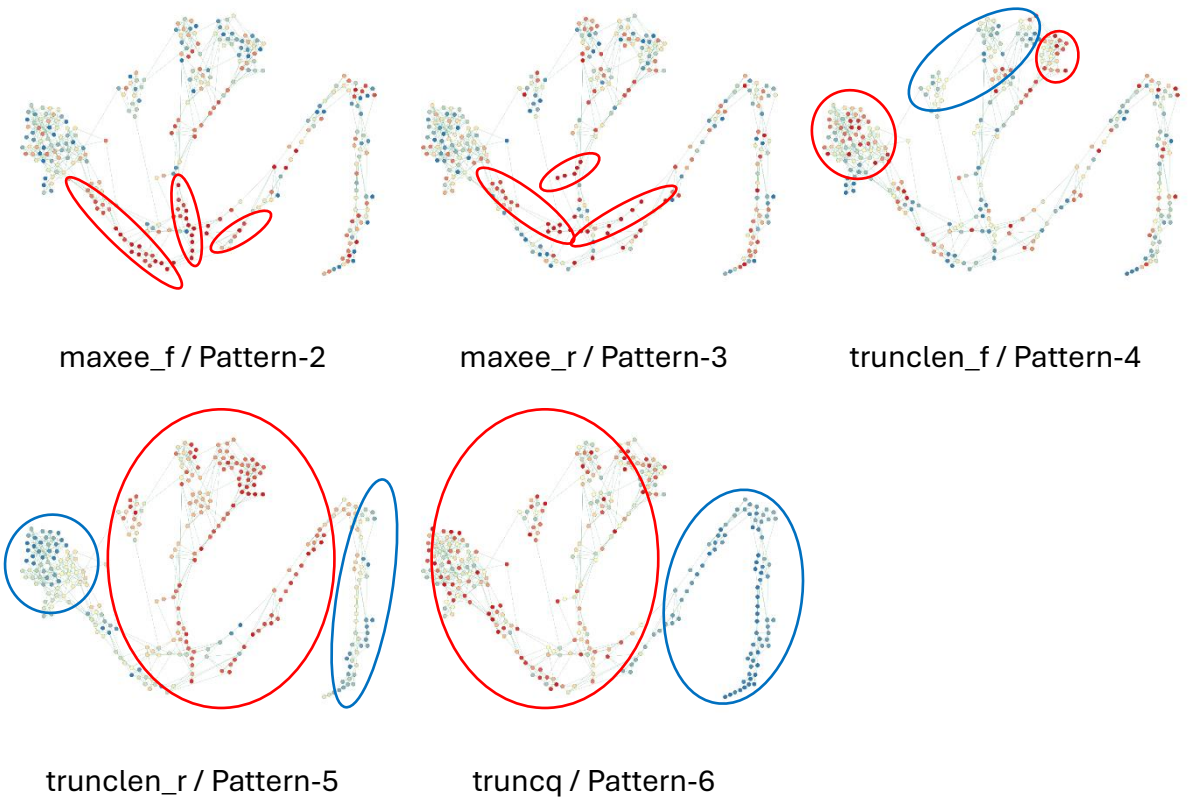


Figure 19 Overlaid parameters on the network graph – patterns (2) to (6)

In "truncq" higher values are completely in the tail and lower values in the main body and finaly for "trunclen_f" a concentration of lower values on clusters A and B except for the region in cluster B which discussed earlier in Figure 17. These patterns are important for relating effects of the parameters to ASV tables. (Figure 19)

4.2.2 Overlaying microbes (families) information

In order to analyze all families properly, the values of ASV tables (abundance values) for each family are overlaid on the network graph of pipelines for each sample. The aim of this exploration is to find any existing patterns, check if the pattern remains consistent within the family across different samples, and, in case of inconsistencies, determine their frequency and identify any emerging patterns. Also to check if pipelines behave differently for different families of the microbes.

To achieve this goal, families are sorted based on the thresholds used in the table and their presence over time (across samples). The idea is that ASV tables with higher values will create a more detailed and clearer gradient of colors when overlaid on the network graph of pipelines. When the values of the ASV tables are lower, patterns could vary significantly because the range from minimum to maximum is not wide enough to vary smoothly. By checking for the presence of patterns in these graphs, potential relationships between parameters and ASV tables will be identified.

Starting by "Bacteroidaceae" ,"Lachnospiraceae", "Lactobacillaceae", and "Muribaculaceae", a consistent pattern of the colors found for all of them. Figure 20 illustrates an example of this pattern using sample 8 (Day 147) for "Bacteroidaceae" family. Interestingly lower values of the ASV tables appears at tail of the network graph. End of the tail is consisting of the lowest values and it increases by going towards the main body. Among the labeled patterns, this pattern is very close to pattern-5.

Moderate values of ASV tables are located around the position in which the main body is connected to the tail, including path 1 and path 2 from structures found in section 4.2. Although different patterns were found for these two paths in section 4.2 (pattern-2) and pattern-3), ASV table found by the pipelines belonging to these two structures are showing very similar values with a slightly smaller values in path 2 (pattern-2).

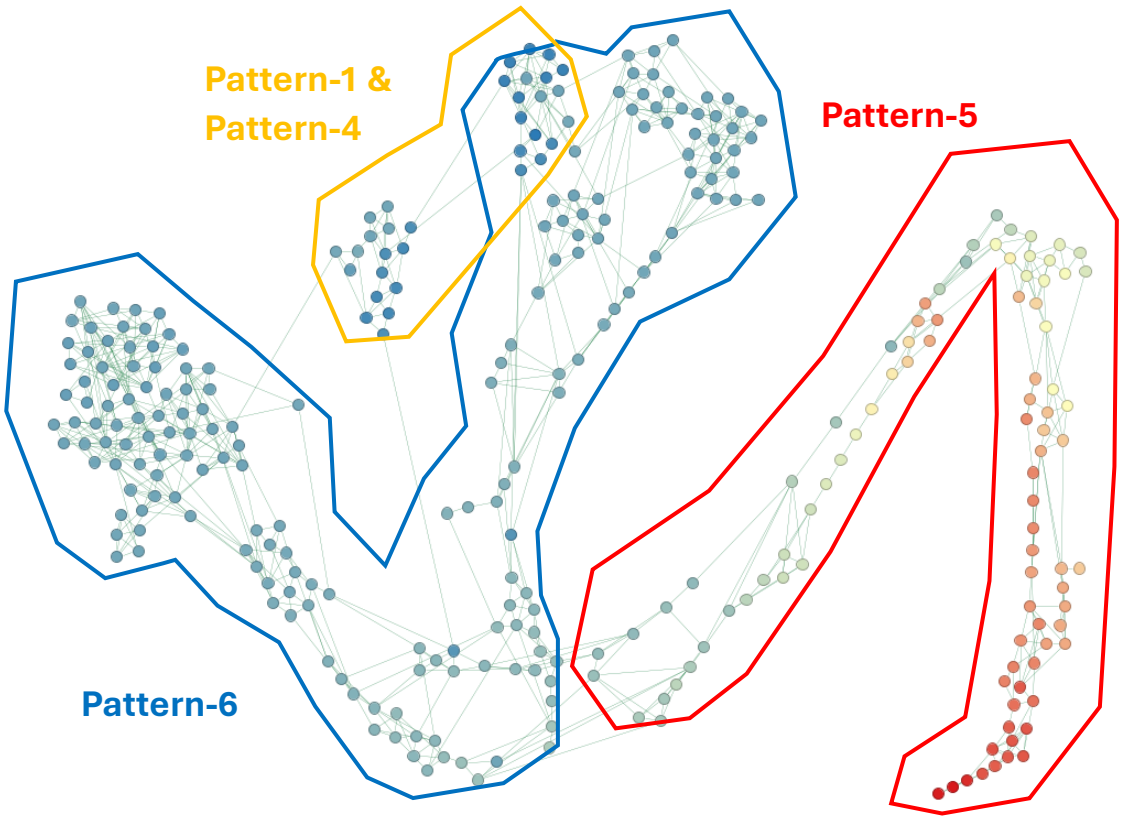
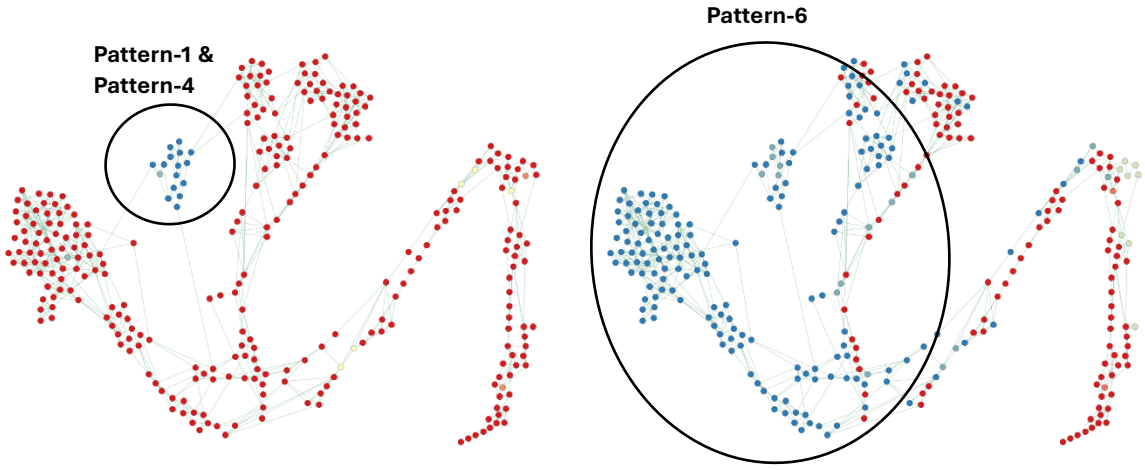


Figure 20 Family abundance on the network graph (Sample 8 Bacteroidaceae)

Higher values are present in clusters A and B with a smooth transition from position of moderate values to these positions. Left side of the pattern-6 is very similar to what we see for these higher values. Maximum values are located in cluster C, for which pattern-1 was found in section 4.2, however, this pattern is similar in all fourteen parameters explained in the same section and it's hard to say directly how this region is affected by different parameters. In addition to that pattern-4 also shows a similar behavior for the pipelines located in cluster C and a region of cluster B.

The transition of colors are smoothly and no exception appears in the transition from tail towards the main body. In general pipelines closer to the tail generate ASV tables with lower values in compare with pipelines located in the main body.

For moderate values of ASV table (50 < threshold <500) and for all frequencies (from 10 to 19) such as , same behavior was observed, however, for some samples exceptions with a dominant pattern appears. This exceptions doesn't change the logic behind what observed for the previous families. (Figure 21)



(Day 141) (Day 142)

Figure 21Example of exceptions for "Christensenellaceae"

As families with low values of ASV tables and low frequencies are appearing time to time, patterns for the pipelines succeed in finding these values are important and those parameters could be used potentially for founding microbes with low frequencies (“Staphylococcaceae”, “Atopobiaceae”, “Deinococcaceae”, “Actinomycetaceae”, “Anaerofustaceae”, “Bacillaceae”, “Listeriaceae”, “Neisseriaceae”, “Moraxellaceae”). Even for these families, the logic remains the same. The dominant pattern for these families by far is pattern-1 and pattern-4, specifically cluster C. In these families which instances of microbes are shows up rarely, cluster C is the main group of pipelines succeed in founding these instances.

5 Discussion

In this study by analyzing values of parameters involved in converting raw rRNA sequences into ASV tables, several main patterns found yields how these parameters affect the values of ASV tables. Concentration of minimum values at the end of the tail and maximum values in cluster C is the most important finding. Pipelines connection the end of the tail to main body are finding more and more instances of microbes by moving from tail towards the main body. The location in which tails connect to the main body is mostly including moderate values and by moving towards clusters A and B, values increases. This pattern exists among all the families and no exception found against it.

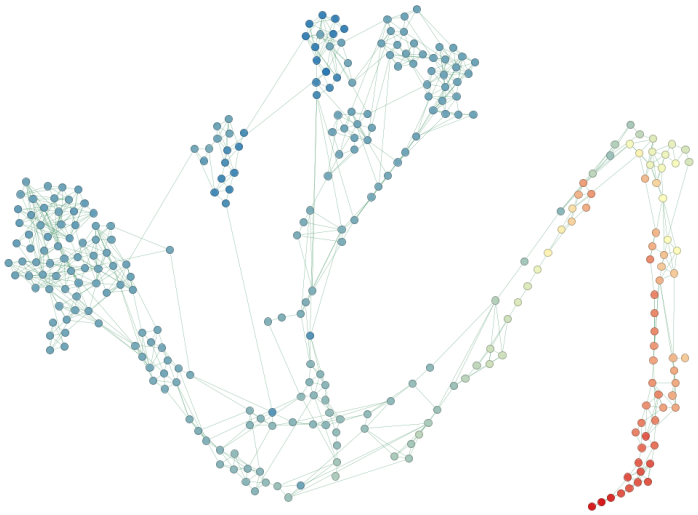


Figure 22 Main pattern in ASV tables

The main parameters which creates this effect are “trunclen_f”, “trunclen_r” and “truncq”. In addition, parameters which yielding pattern-1 are important for the families with very small values in ASV tables as these instances are mostly found by cluster C. It’s hard to say which parameter out of many involved in this pattern (Figure 16 and Figure 17) is the important one, as all of them are concentrated on this region.

Based on these findings, truncation is the most important factor in increasing the value of ASV tables. High values of “trunq” are located in tail and moderate and low values in the main body, for “trunclen_r” pattern is more interesting, high values are in the tail and left side of the main body, namely cluster A, finally in “trunclen_f” a concentration of higher values in cluster C and lower values in clusters B and C is present. To increase the values in ASV tables we should lower the values in truncation, exceptionally effect of higher values of “trunclen_f”in combination with parameters of pattern-1 resulting in high values in ASV tables.

“rm.phix”, “rm.lowercomplex”, “minlen” and “minq” could be used with different values without effecting the results significantly. “maxee_f” and “maxee_r” are making two paths potentially due algorithm behind DADA2 but resulting in very close results in ASV tables.

5.1 Drawbacks of the methodology and Ideas for future studies

This study aimed to find the effects of the pipeline parameters on transforming laboratory work to available form in data analysis. Insights from this study is coming from a solid foundation of observing noticeable patterns, however, in order to claim them as fact, statistical tests must be done.

First of all, although this study provided very good insights about how parameters combination effect ASV table, it didn’t answer to the question “What is the relation between set of parameters and presence of an ASV table as outcome? ”. Studying such an effect could be used in a two-step process of choosing appropriate values for the parameters, first step insures presence of an output and second step looks for values with proper level of abundance or setting up a pipeline to find rare type of the families.

Secondly, using MSTs for generating the network graph of the pipelines is one of the simplest methods among many more. More complex methods such as Mapper could be used in generating the network graph. Also more advanced techniques such as Persistent Homology or Mapper have more potential in finding more precise patterns. These algorithms although are proper for datasets with much bigger amount of information. Bray-Curtis is also used without normalization as most of the values ranged under 500 in ASV tables, it worth it to look for the same effect after normalization of the values in ASV tables.

Moreover in this study missing values are excluded from the analysis. Effect of these values could adjust the behavior of the patterns or resulted in finding new ones. This loss information is coming from two sources, first unknown sequences, secondly lack of information in deeper level of the hierarchy. Studying the same effect in different levels of the taxonomic system and over other dataset could show if the same patterns appears again or not. Loss of information at the start point of this study was one of the aspects which accepted to simplify the process.

Finally, a data visualization approach was used in this study, by defining statistical tests for parameters resulting in patterns, and specifically parameters related to truncation, significance of the effect of these parameters could be examined. In this way insights will turn into conclusions with statistical support and could be used as guidelines by the biologists in the analysis process of the raw data.

6 Conclusion

In conclusion noticeable similar patterns were observed in both parameters and microbiome graphs and consequently ASV table are sensitive to different processing pipeline parameters. Abundance of microbiomes not only varies from low to high values by choosing different set of parameters, but also could affect the presence of an ASV table as an outcome.

Network is constructed out of smaller regions, namely three clusters (A,B, and C) and a tail are present in the network. Two paths exist from the tail towards the clusters which representing forward and reverse read of the data in DADA2 algorithm. These paths however behave similarly in the values of the ASV tables.

In network graph of the pipelines a smooth transition of ASV values is present from the tail towards the main body, with minimum values located at the end of tail. Parameters related to truncation are the most important one among the others. To get higher values in ASV tables, lower values were used in these parameters. Fourteen parameters are concentrated over certain values mostly resulted in ASV tables without an outcome, however, an outcome comes out of these pipelines, it’s always resulting in maximum values for ASV tables. Other parameters distributed over the whole graph without any significant pattern in the results.

7 References

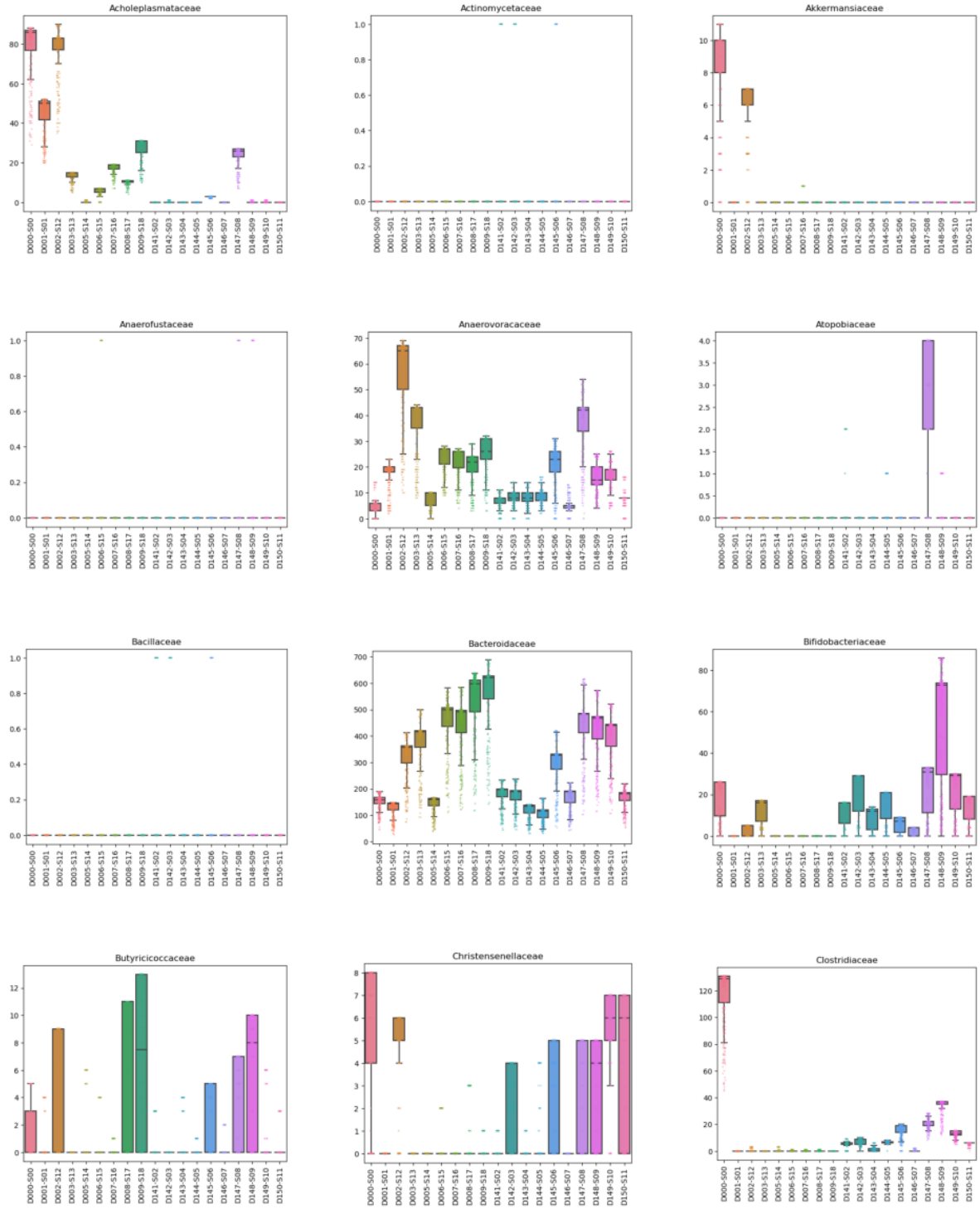
- Adams, A. et al. "Machine Learning and Topological Data Analysis for Predictive Modeling in Biology." Journal of Computational Biology, 2022.
- Anderson, M. J. (2001). A new method for non-parametric multivariate analysis of variance. Austral Ecology, 26(1), 32-46.
- Barabasi, A. L. and Oltvai, Z. N. "Network Biology: Understanding the Cell's Functional Organization." Nature Reviews Genetics, 2004.
- Bastian, M., Heymann, S., & Jacomy, M. (2009). Gephi: An open source software for exploring and manipulating networks. International AAAI Conference on Weblogs and Social Media. Retrieved from
- Bokulich, N. A., Dillon, M. R., Bolyen, E., Kaehler, B. D., Huttley, G. A., & Caporaso, J. G. (2018). q2-sample-classifier: machine-learning tools for microbiome classification and regression. Journal of Open Research Software, 6(1), 11.
- Borg, I., & Groenen, P. J. F. (2005). Modern multidimensional scaling: Theory and applications (2nd ed.). Springer.
- Bray, J. R., & Curtis, J. T. (1957). An ordination of the upland forest communities of southern Wisconsin. Ecological Monographs, 27(4), 325-349.
- Callahan BJ, McMurdie PJ, Rosen MJ, Han AW, Johnson AJA, Holmes SP. DADA2: High-resolution sample inference from Illumina amplicon data. Nat Meth. 2016;13:581–583. pmid:27214047
- Callahan, B. J., McMurdie, P. J., & Holmes, S. P. (2017). Exact sequence variants should replace operational taxonomic units in marker-gene data analysis. The ISME Journal, 11(12), 2639-2643.
- Chen, J. et al. "Predictive Modeling of Biological Systems Using Minimum Spanning Trees and Machine Learning." Bioinformatics, 2021.
- Edgar, R. C. (2016). UCHIME2: improved chimera prediction for amplicon sequencing. bioRxiv. DOI: 10.1101/074252.
- Gao, X. et al. "Minimum Spanning Tree and Its Applications in Data Analysis." IEEE Access, 2020.
- Jacomy, M., Venturini, T., Heymann, S., & Bastian, M. (2014). ForceAtlas2, a continuous graph layout algorithm for handy network visualization designed for the Gephi software. PLOS ONE, 9(6), e98679.
- Jolliffe, I. T. (2002). Principal component analysis (2nd ed.). Springer.
- Jones MB, Highlander SK, Anderson EL, Li W, Dayrit M, Klitgord N, et al. (2019) Library preparation methodology can influence genomic and functional predictions in human microbiome research. Proceedings of the National Academy of Sciences of the United States of America, 116(47): 23268-23275
- Kozich JJ, Westcott SL, Baxter NT, Highlander SK, Schloss PD. (2013): Development of a dual-index sequencing strategy and curation pipeline for analyzing amplicon sequence data on the MiSeq Illumina sequencing platform. Applied and Environmental Microbiology. 79(17):5112-20.

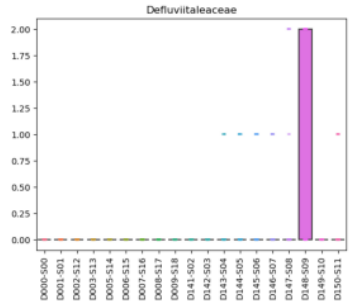
References

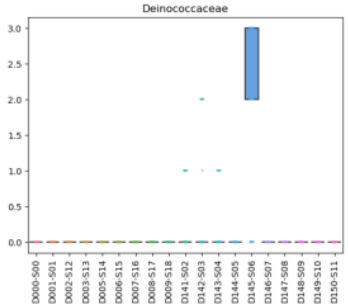
- Krebs, C. J. (1999). Ecological Methodology. Benjamin Cummings.
- Lee, C. et al. "Machine Learning Techniques for Improving Minimum Spanning Tree Construction in Topological Data Analysis." Pattern Recognition Letters, 2023.
- Legendre, P., & Legendre, L. (2012). Numerical Ecology (3rd ed.). Elsevier.
- Maaten, L. van der, & Hinton, G. (2008). Visualizing data using t-SNE. Journal of Machine Learning Research, 9, 2579-2605. Retrieved from
- Maaten, L. van der, and Hinton, G. "Visualizing Data using t-SNE." Journal of Machine Learning Research, 2008.
- Meyer, M. et al. "Interactive Visualization for Protein Folding." IEEE Transactions on Visualization and Computer Graphics, 2021.
- Newman, M. E. J. "Modularity and Community Structure in Networks." Proceedings of the National Academy of Sciences, 2006.
- Nguyen, N. H., Song, Z., Bates, S. T., Branco, S., Tedersoo, L., Menke, J., ... & Kennedy, P. G. (2021). FUNGuild: An open annotation tool for parsing fungal community datasets by ecological guild. Fungal Ecology, 20, 241-248.
- Prim, R. C. (1957). Shortest connection networks and some generalizations. Bell System Technical Journal, 36(6), 1389-1401.
- Prodan, A., Tremaroli, V., Brolin, H., Zwinderman, A. H., Nieuwdorp, M., & Levin, E. (2020). Comparing bioinformatic pipelines for microbial 16S rRNA amplicon sequencing. PLoS ONE, 15(1), e0227434.
- Schloss, P. D. (2021). Reintroducing mothur: 10 years later. Applied and Environmental Microbiology, 87(12), e00901-21.
- Shannon, P. et al. "Cytoscape: A Software Environment for Integrated Models of Biomolecular Interaction Networks." Genome Research, 2003.
- Smith AB, Johnson CL, Davenport ER (2020) Comparative analysis of bioinformatic pipelines to investigate the effects of sequencing depth and other experimental parameters on microbiome diversity. Frontiers in Microbiology, 11: 586.
- Stuart, T. and Satija, R. "Integrative Single-Cell RNA-Seq Analysis." Nature Reviews Genetics, 2019.
- Wang, Y. et al. "Applications of Minimum Spanning Trees in Biomedical Data Analysis." BMC Bioinformatics, 2022.
- Wold, H. et al. "Principal Component Analysis." Chemometrics and Intelligent Laboratory Systems, 1987.

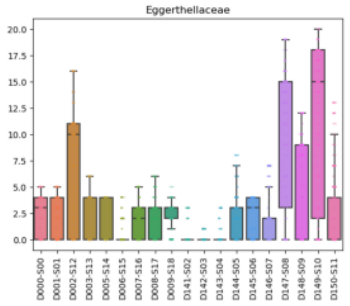
8 Appendices

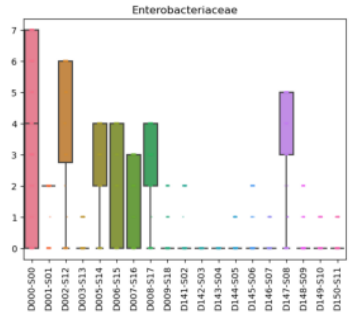
8.1 Boxplots of ASV tables (bundance in all pipelines)

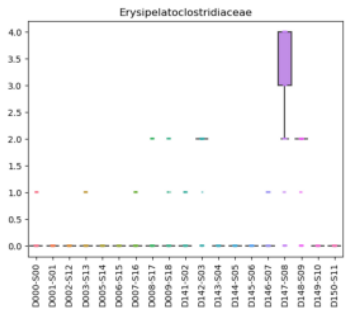


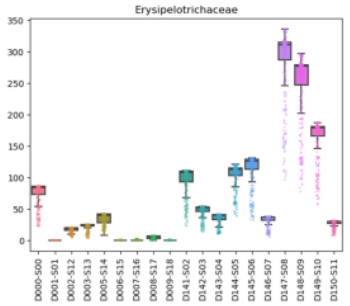

Defluviitaleaceae

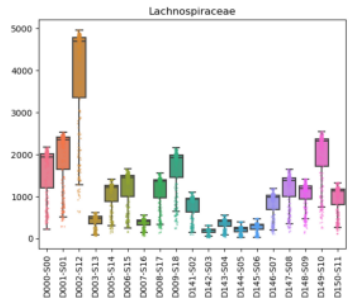

Deinococcaceae

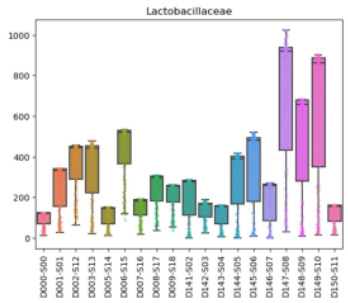

Eggerthellaceae

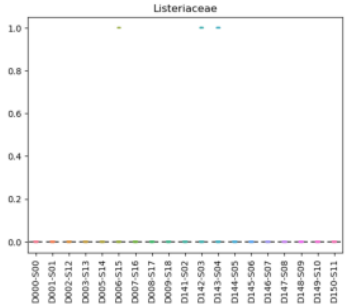

Enterobacteriaceae

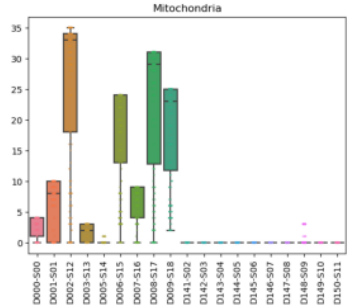

Erysipelatoclostridiaceae

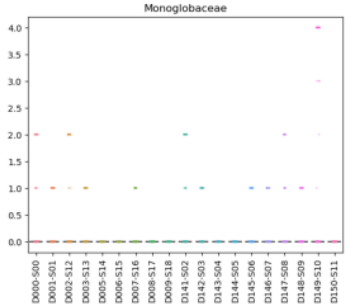

Erysipelotrichaceae

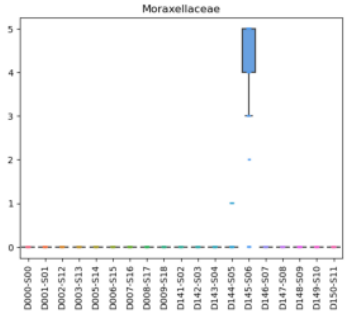

Lachnospiraceae

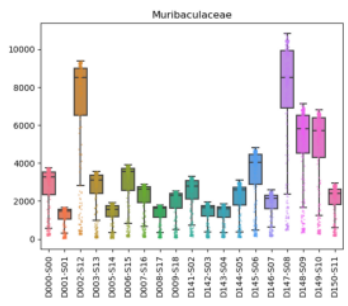

Lactobacillaceae

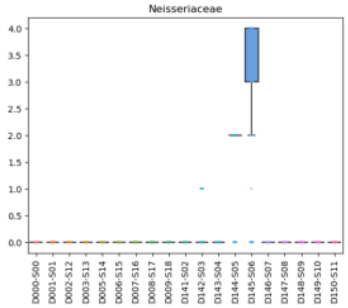

Listeriaceae

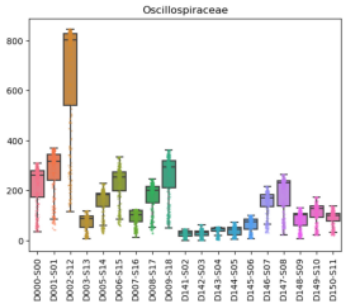

Mitochondria

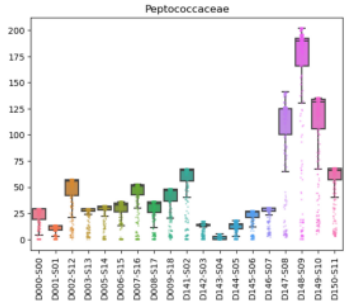

Monoglobaceae

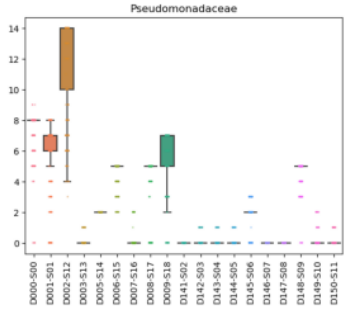

Moraxellaceae

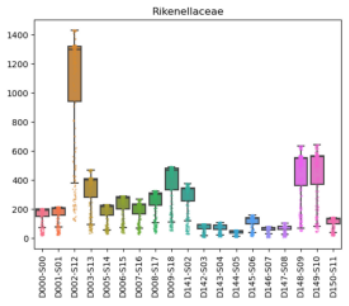

Muribaculaceae

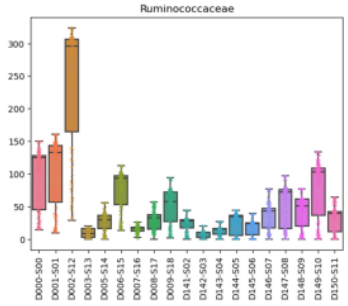

Neisseriaceae

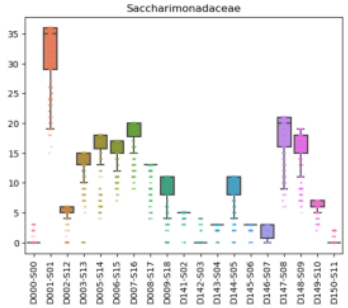

Oscillospiraceae

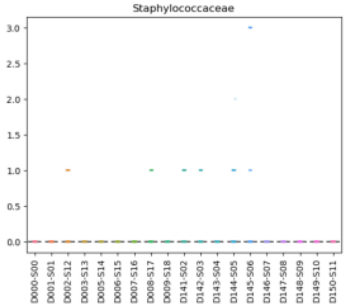

Peptococcaceae

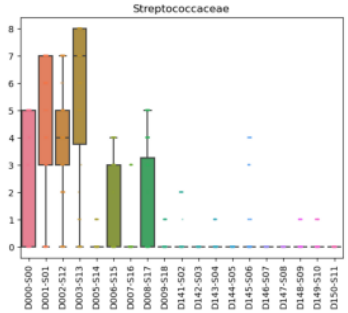

Pseudomonadaceae

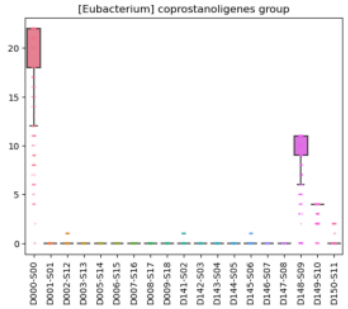

Rikenellaceae


Ruminococcaceae

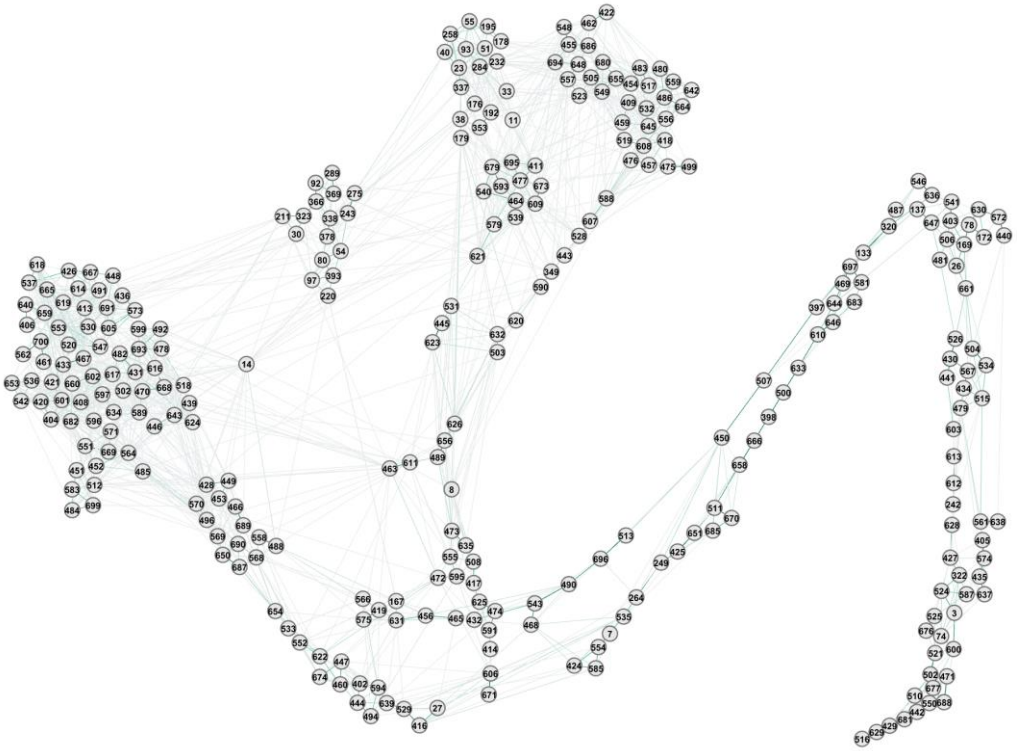

Saccharimonadaceae


Staphylococcaceae

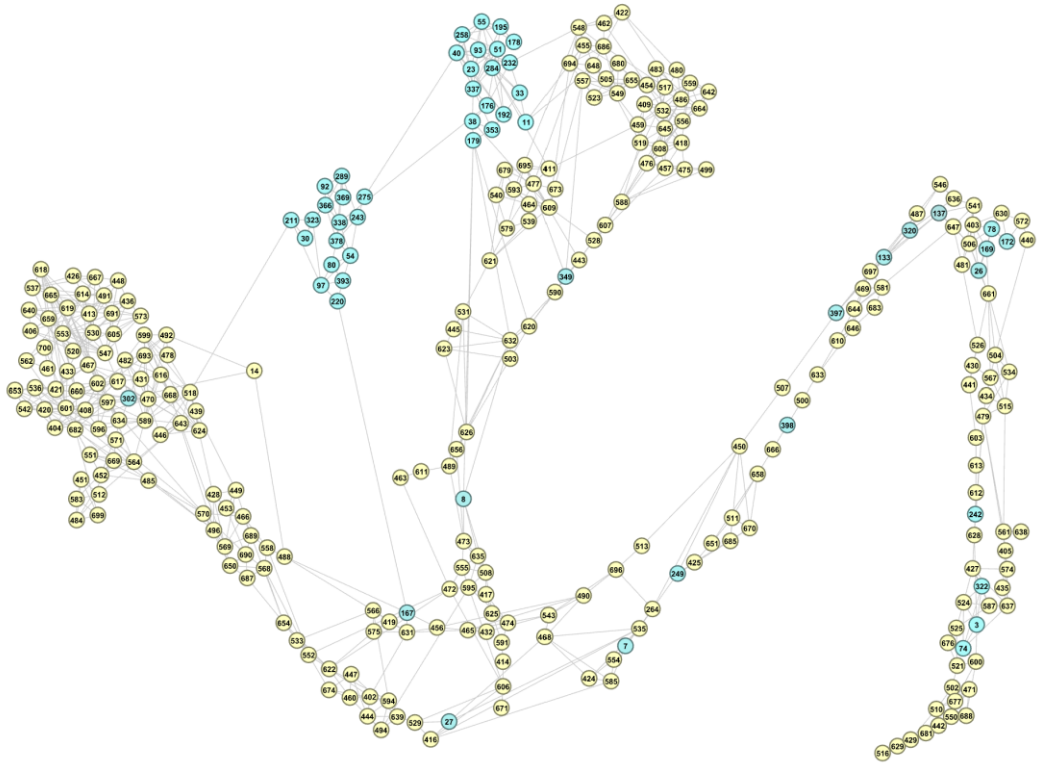

Streptococcaceae


[Eubacterium] coprostanoligenes group

8.2 IDs of the pipelines in network graph



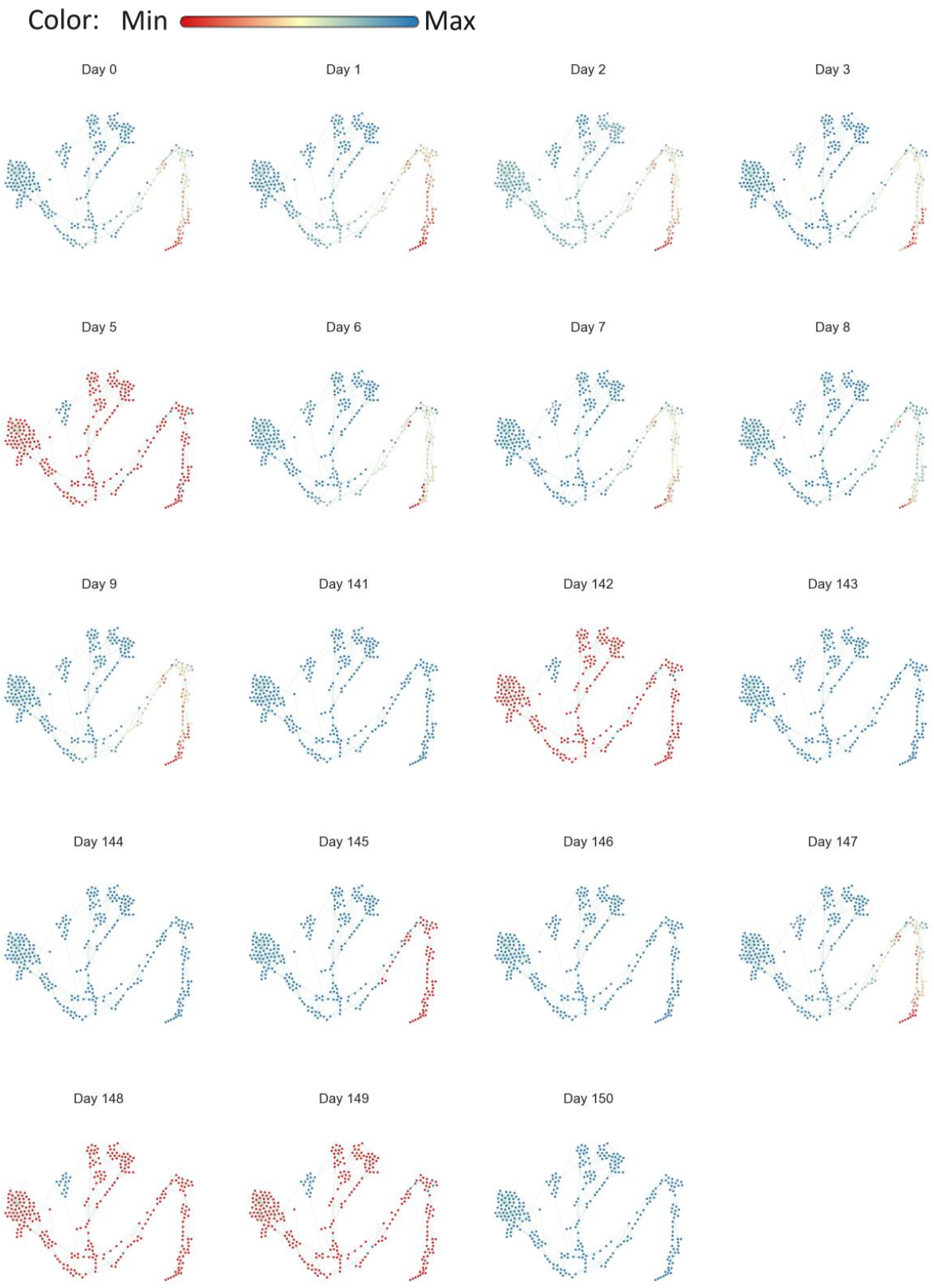
(a) IDs of pipelines in the network graph



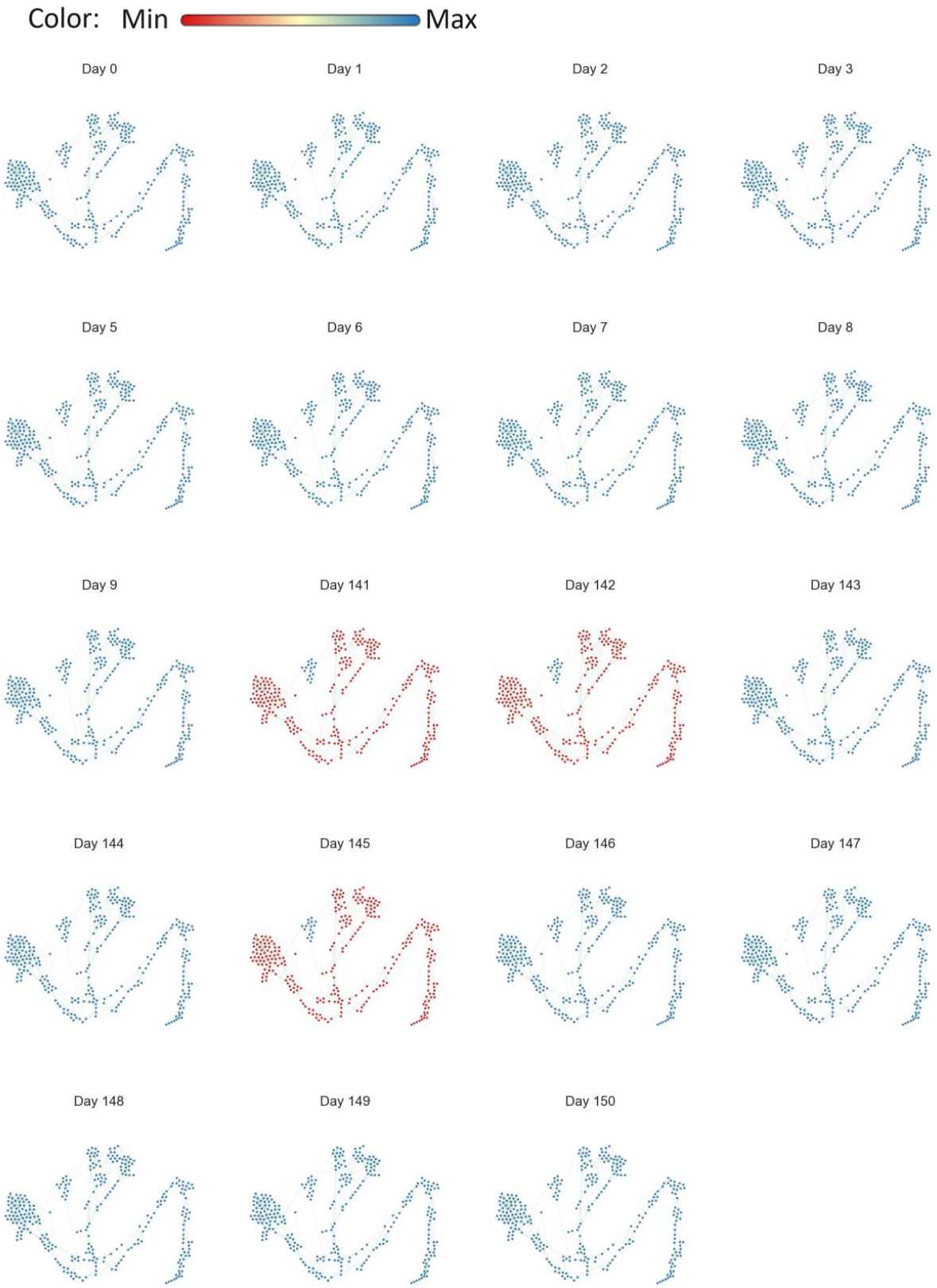
(b) IDs of pipelines and specified pipelines with a pattern in blue

8.3 Network graph of pipelines by family

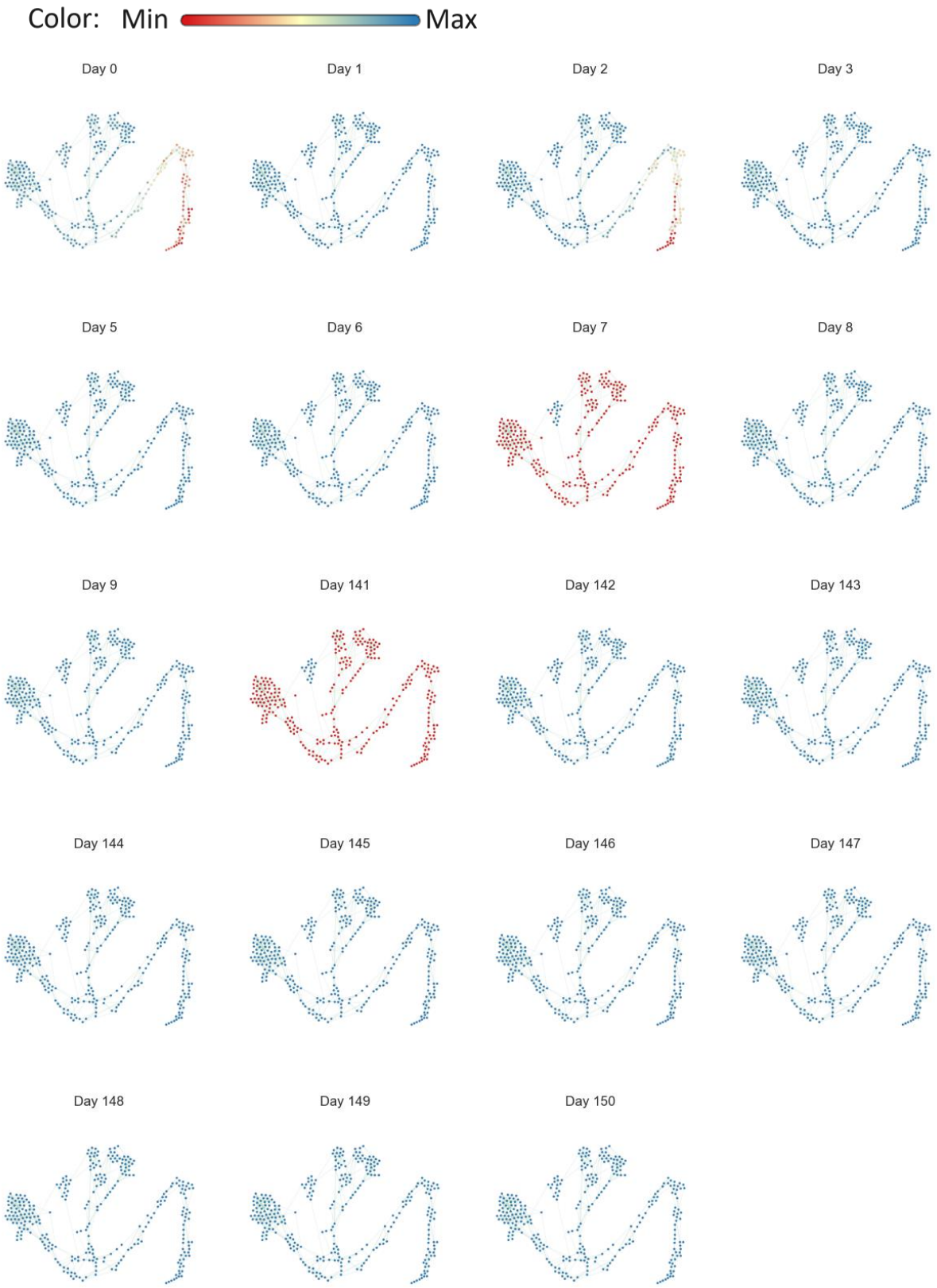
8.3.1 Acholeplasmataceae



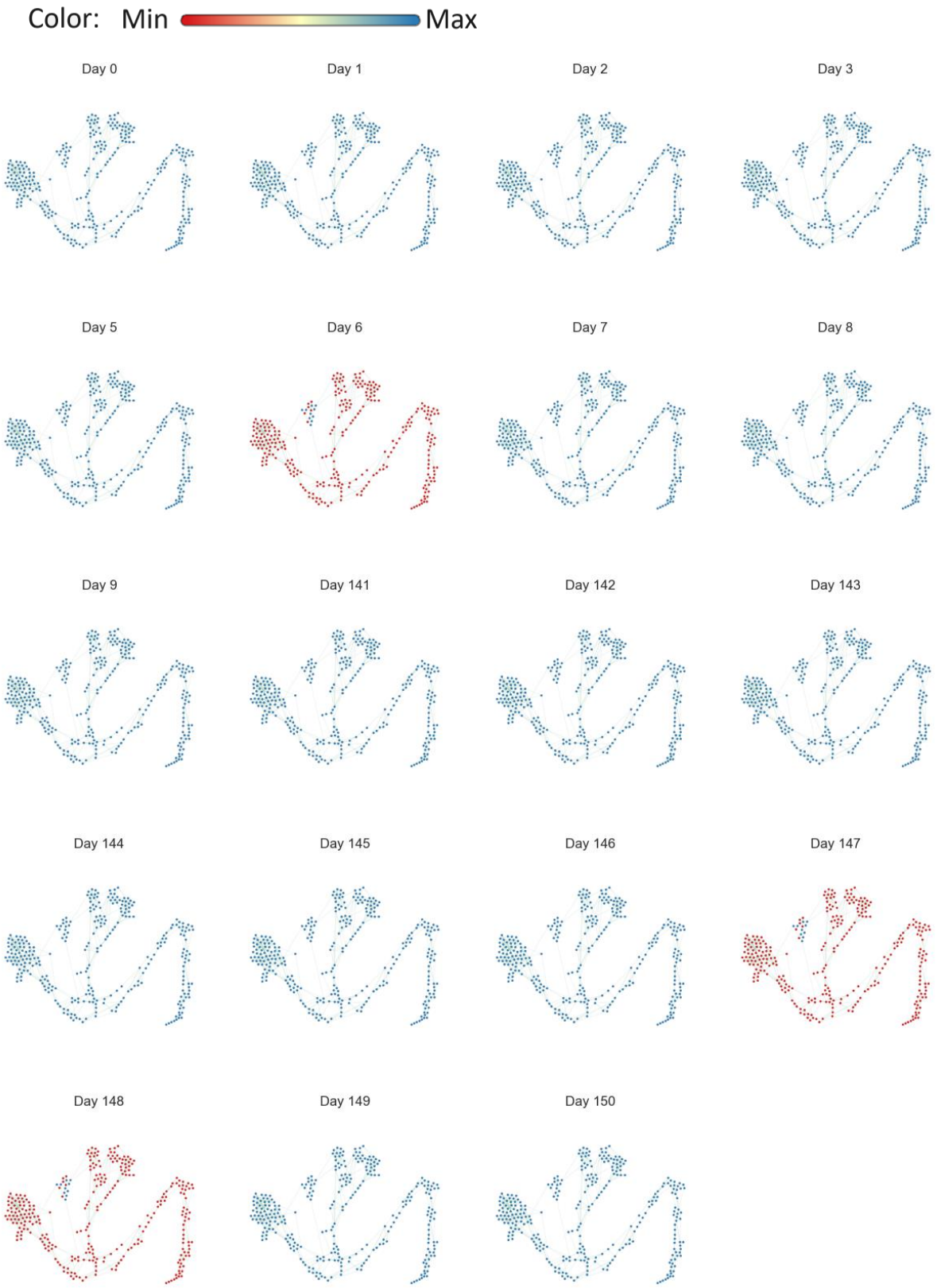
8.3.2 Actinomycetaceae



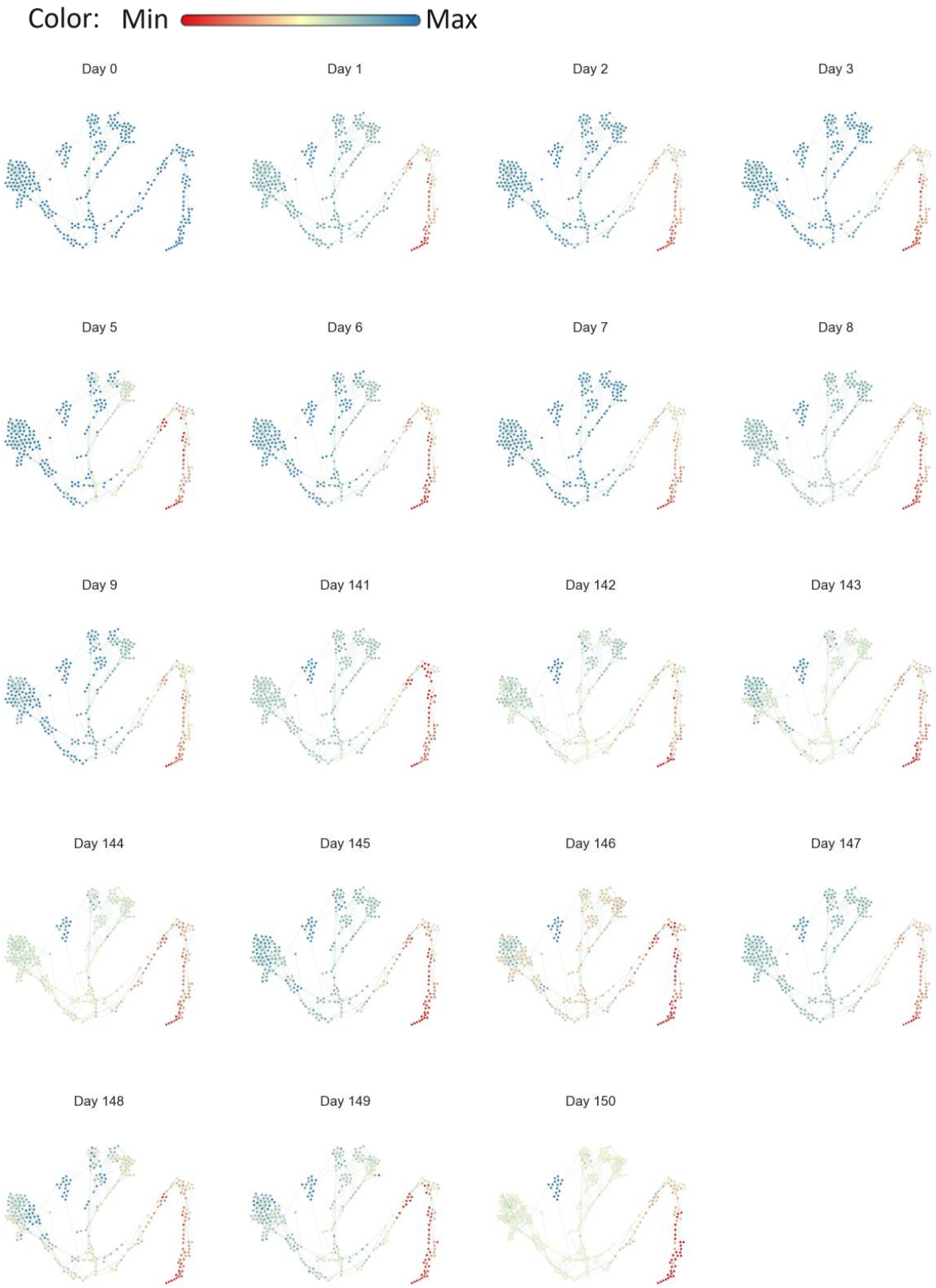
8.3.3 Akkermansiaceae



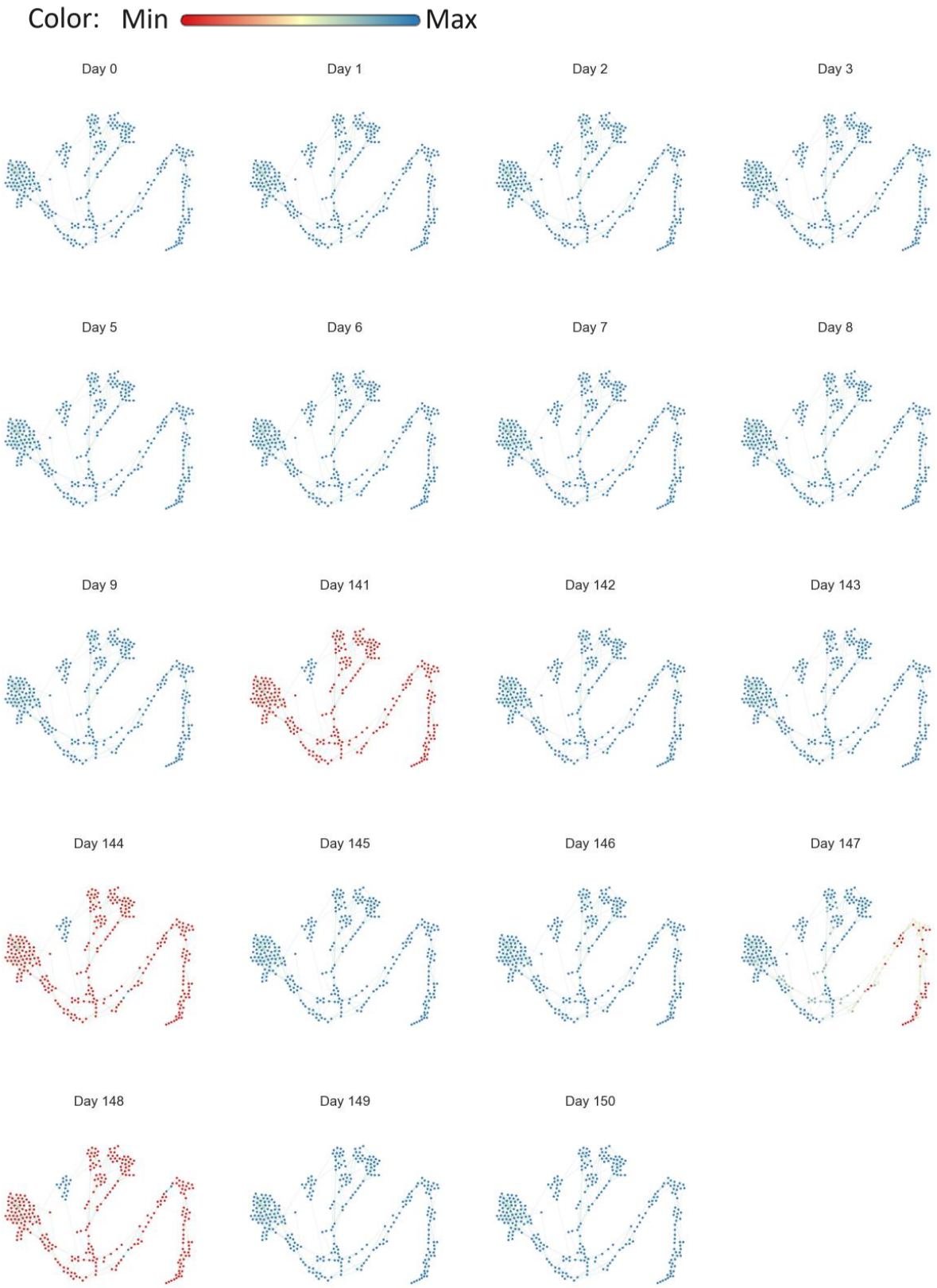
8.3.4 Anaerofustaceae



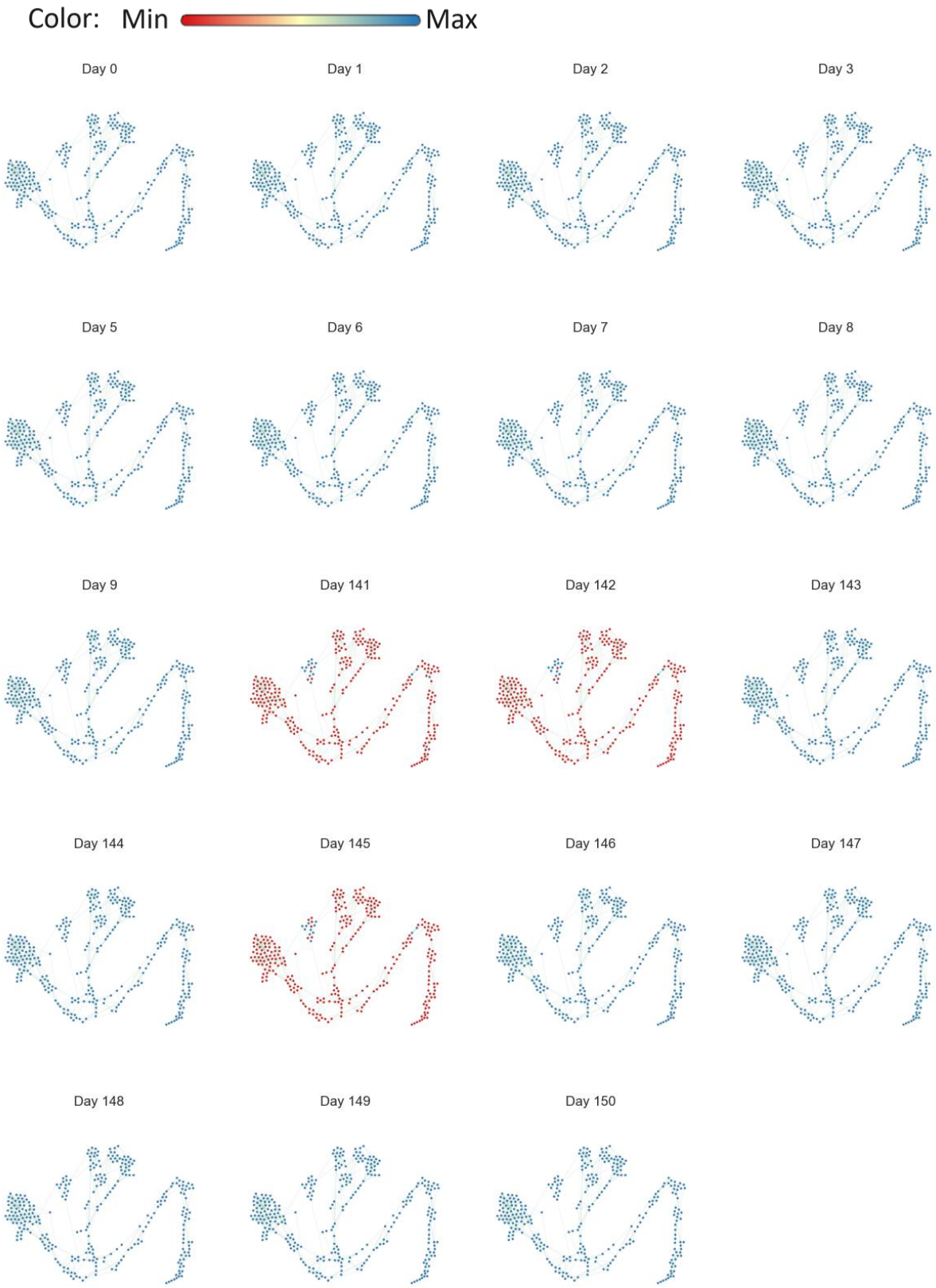
8.3.5 Anaerovoracaceae



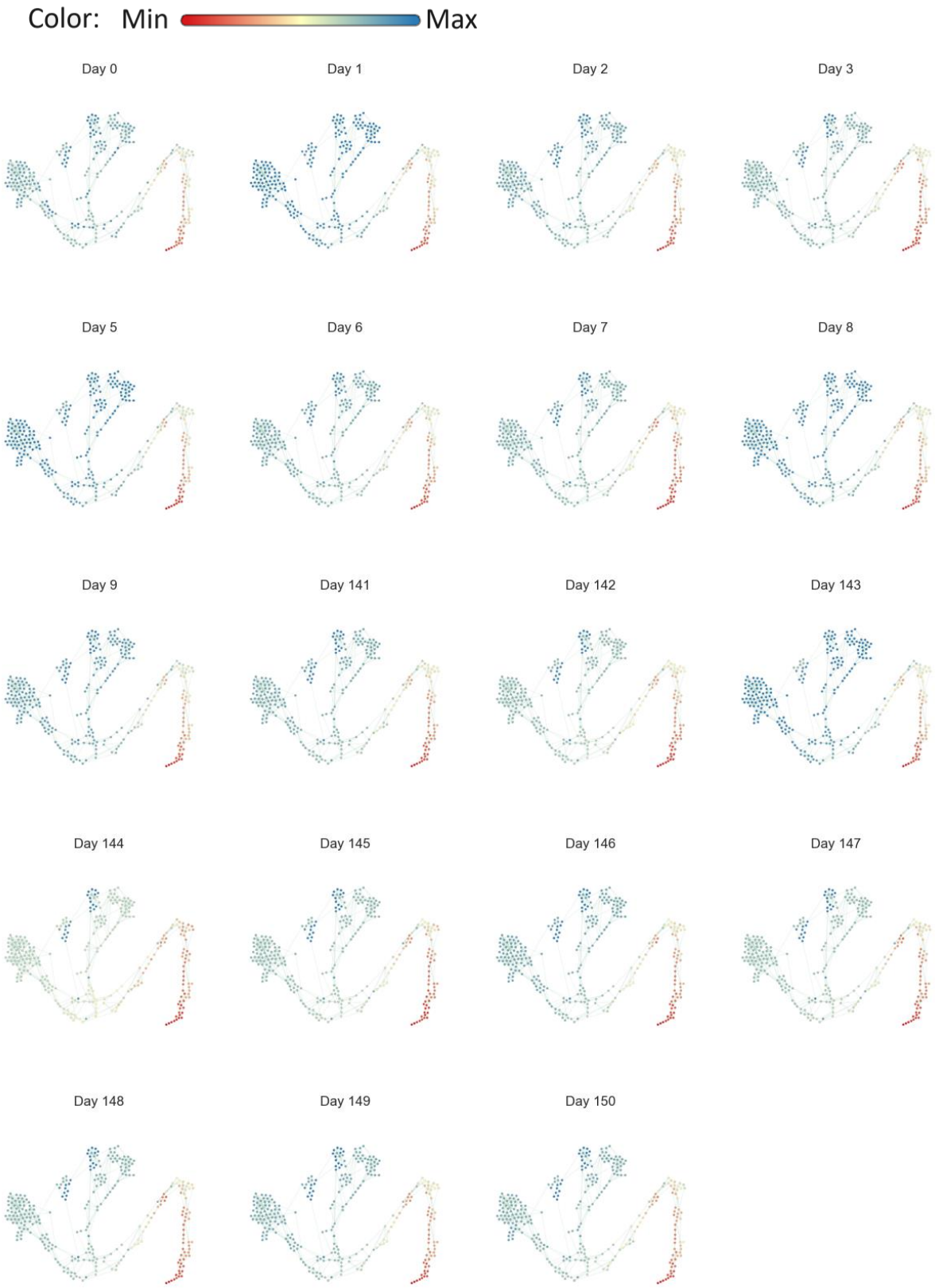
8.3.6 Atopobiaceae



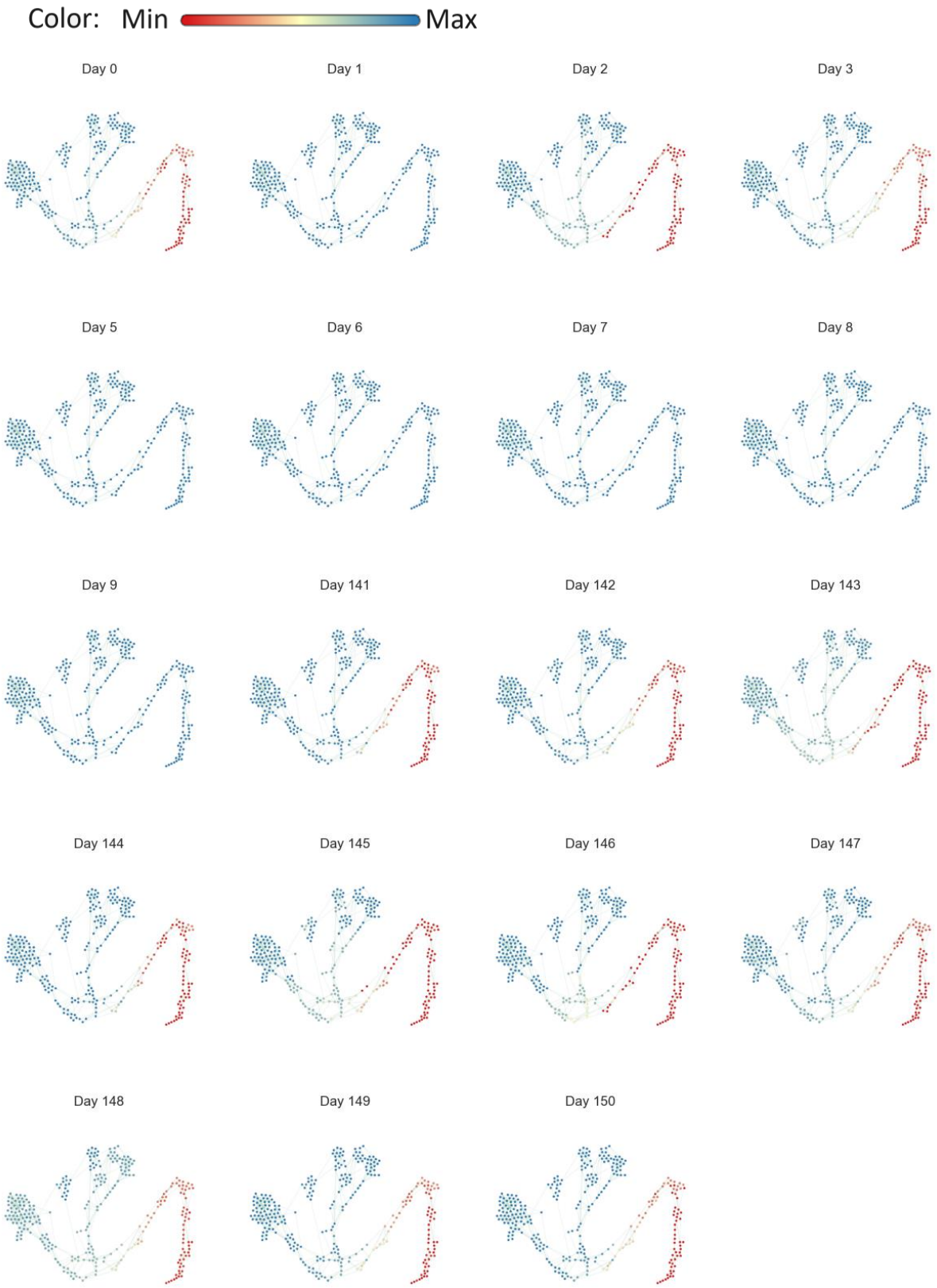
8.3.7 Bacillaceae



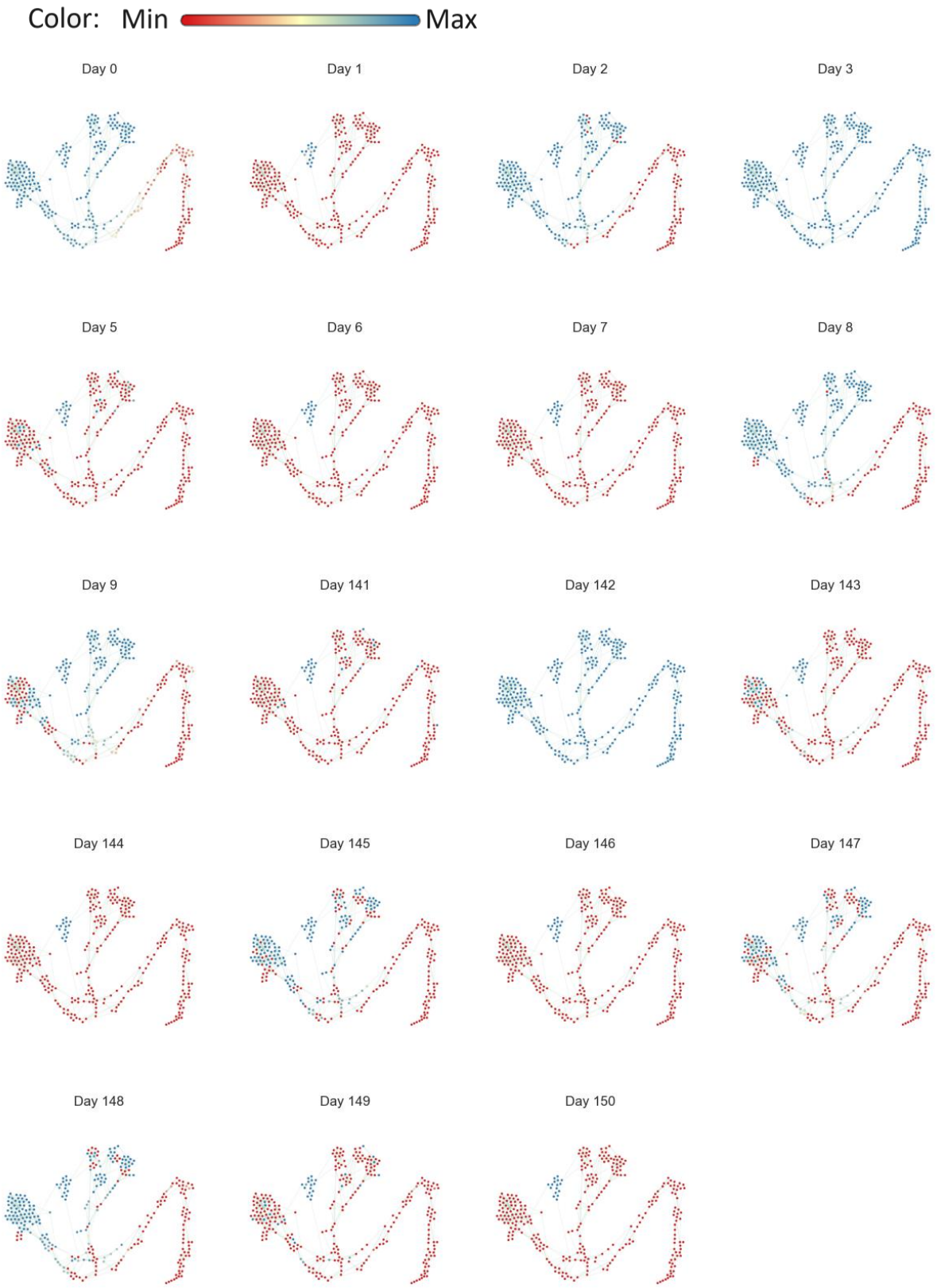
8.3.8 Bacteroidaceae



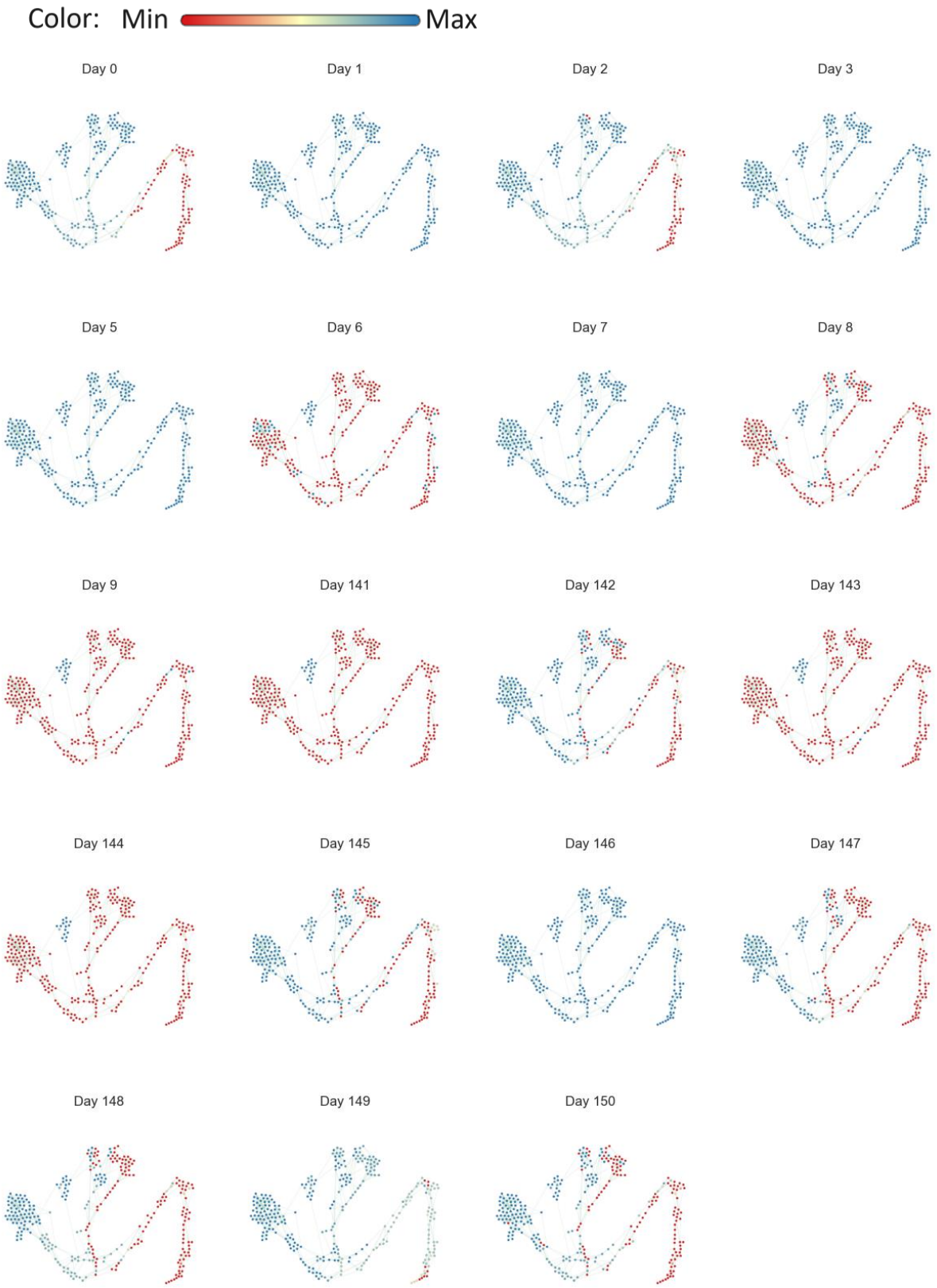
8.3.9 Bifidobacteriaceae



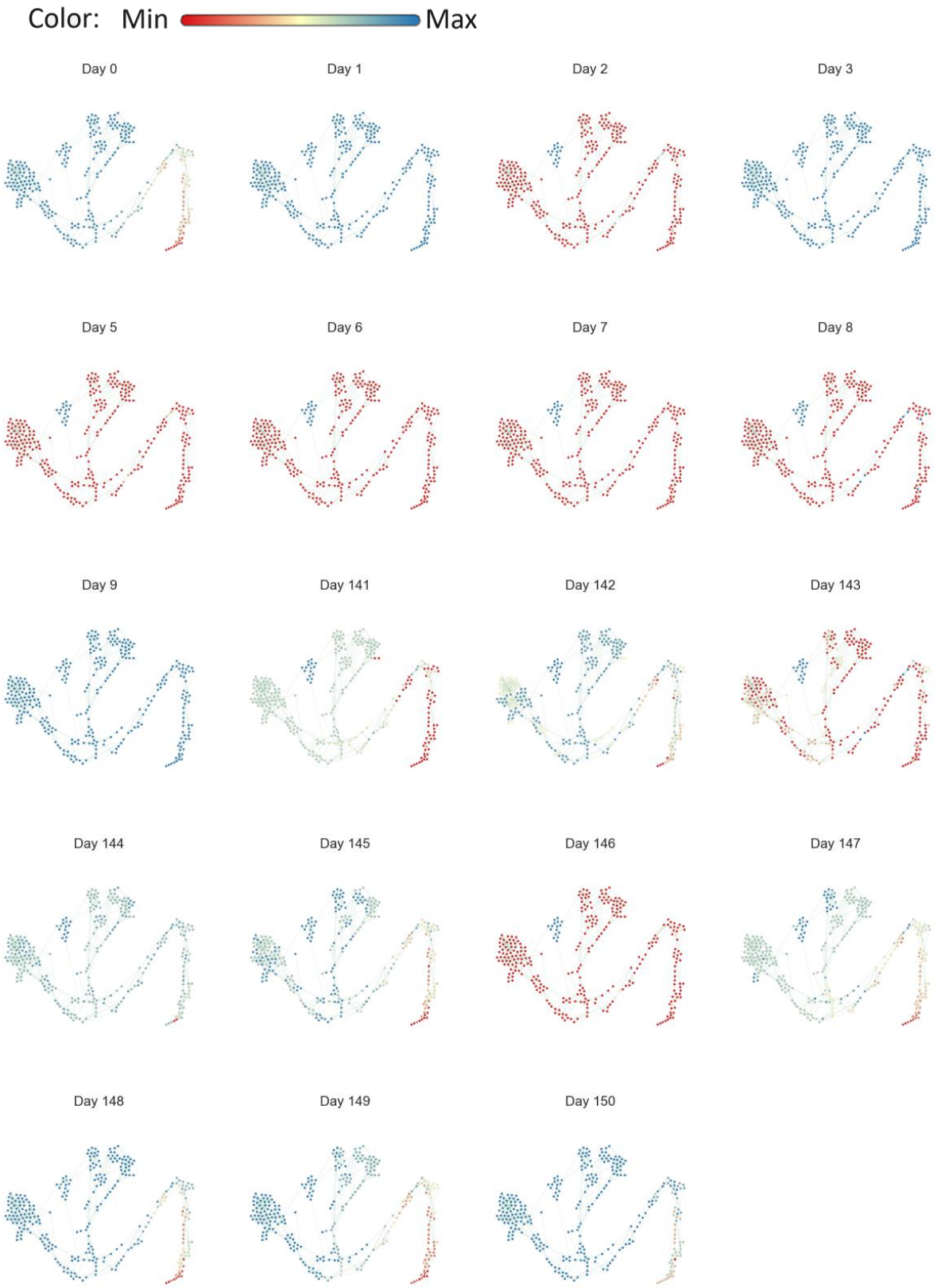
8.3.10 Butyricicoccaceae



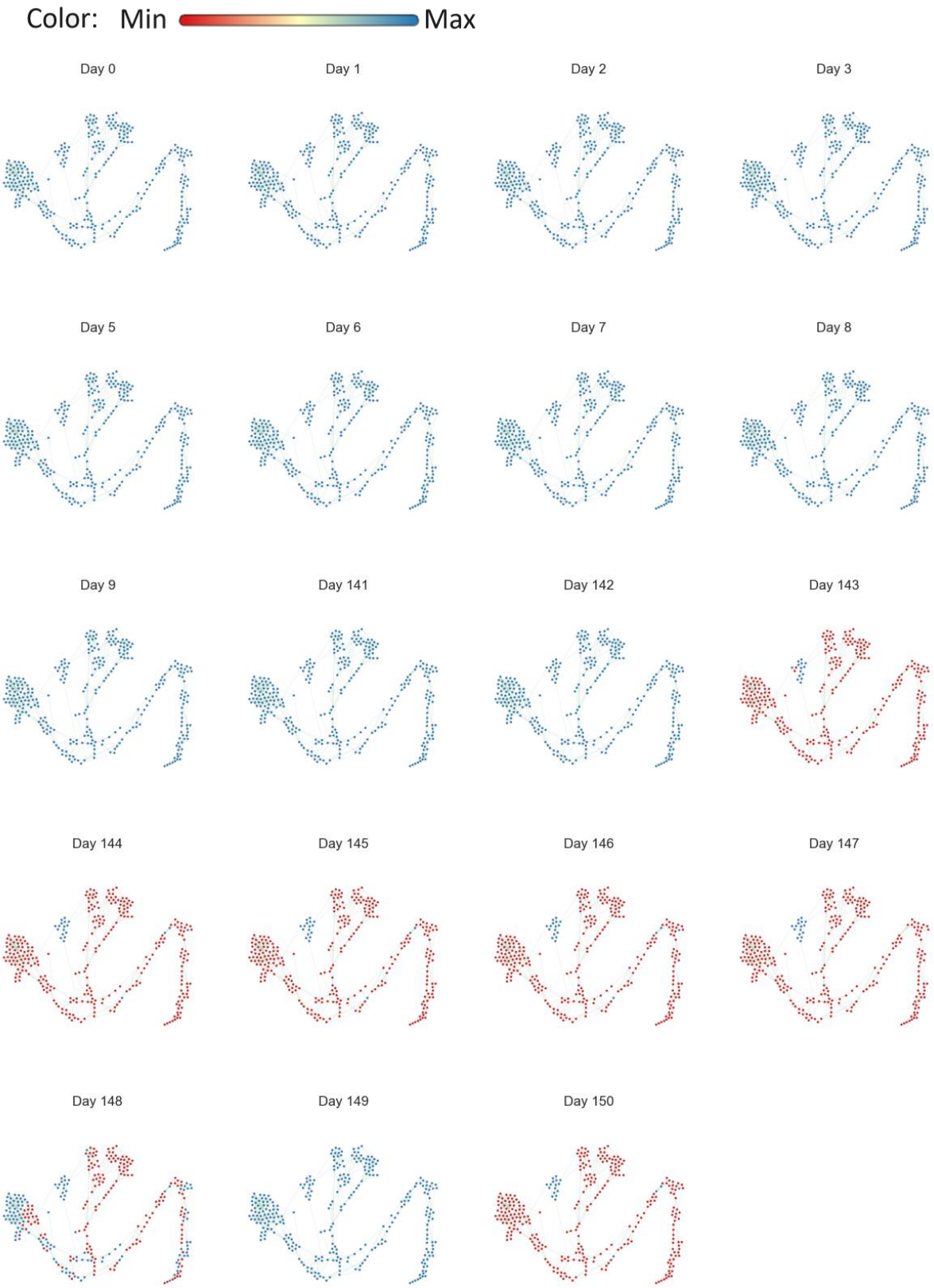
8.3.11 Christensenellaceae



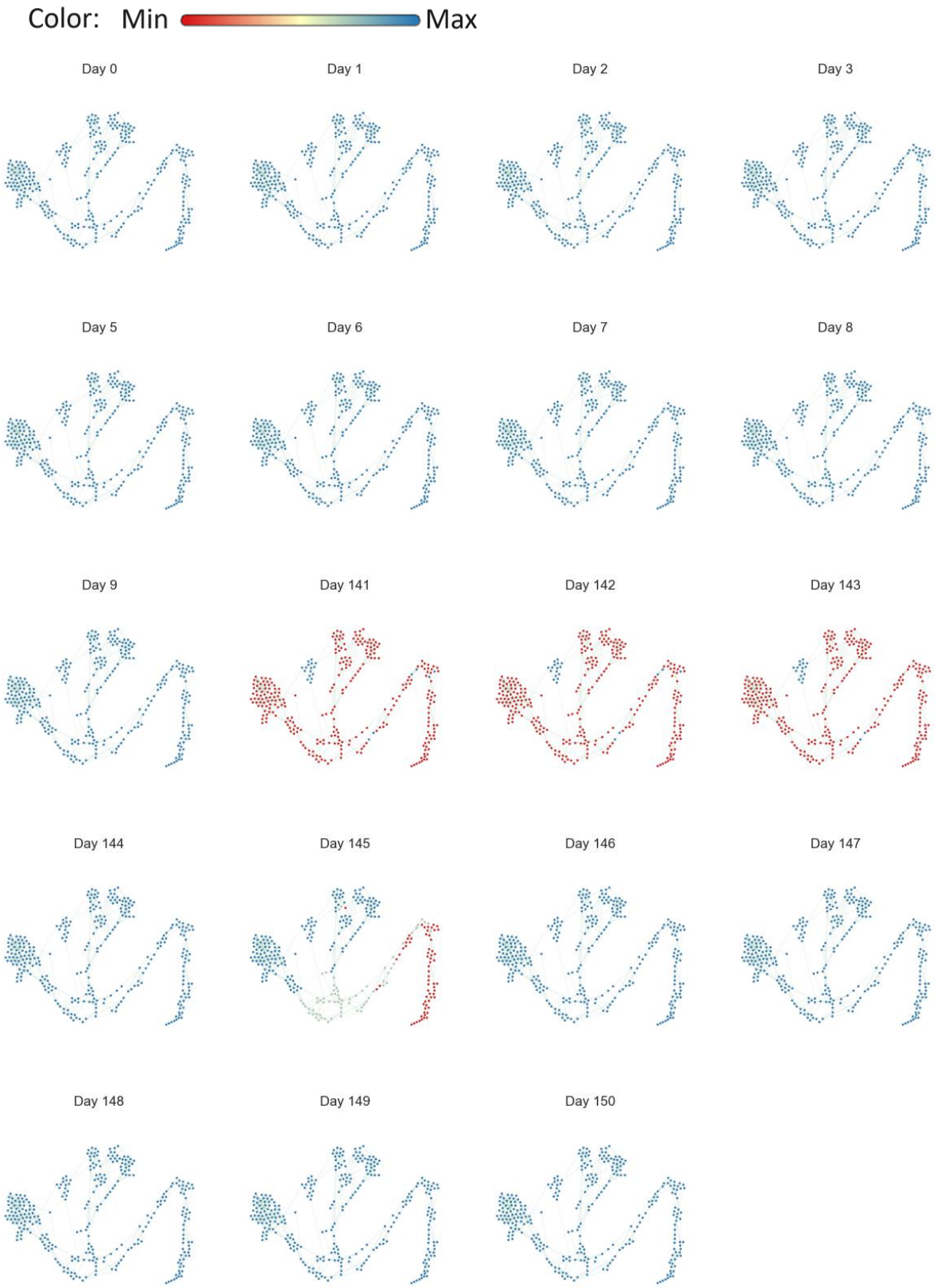
8.3.12 Clostridiaceae



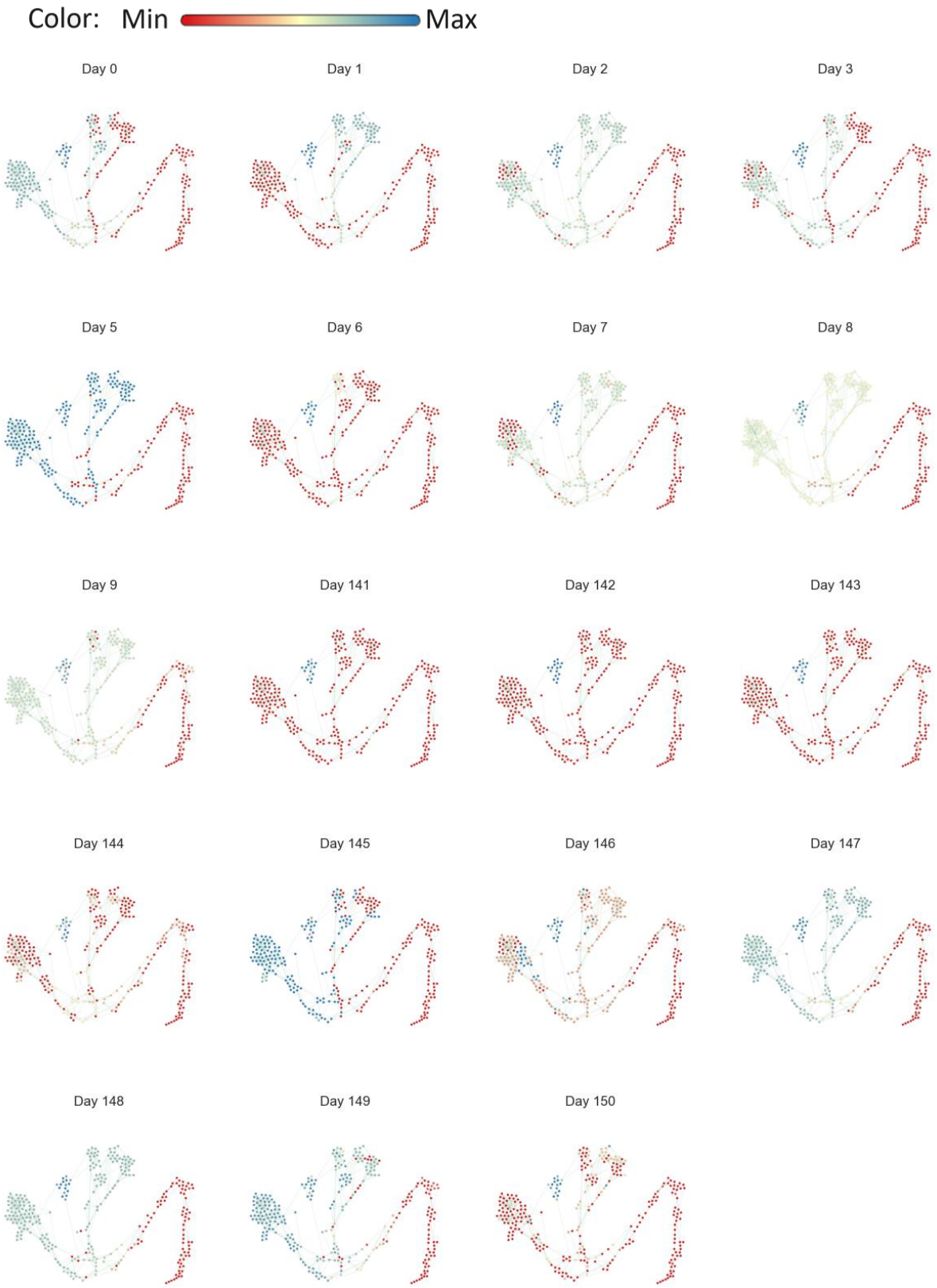
8.3.13 Defluviitaleaceae



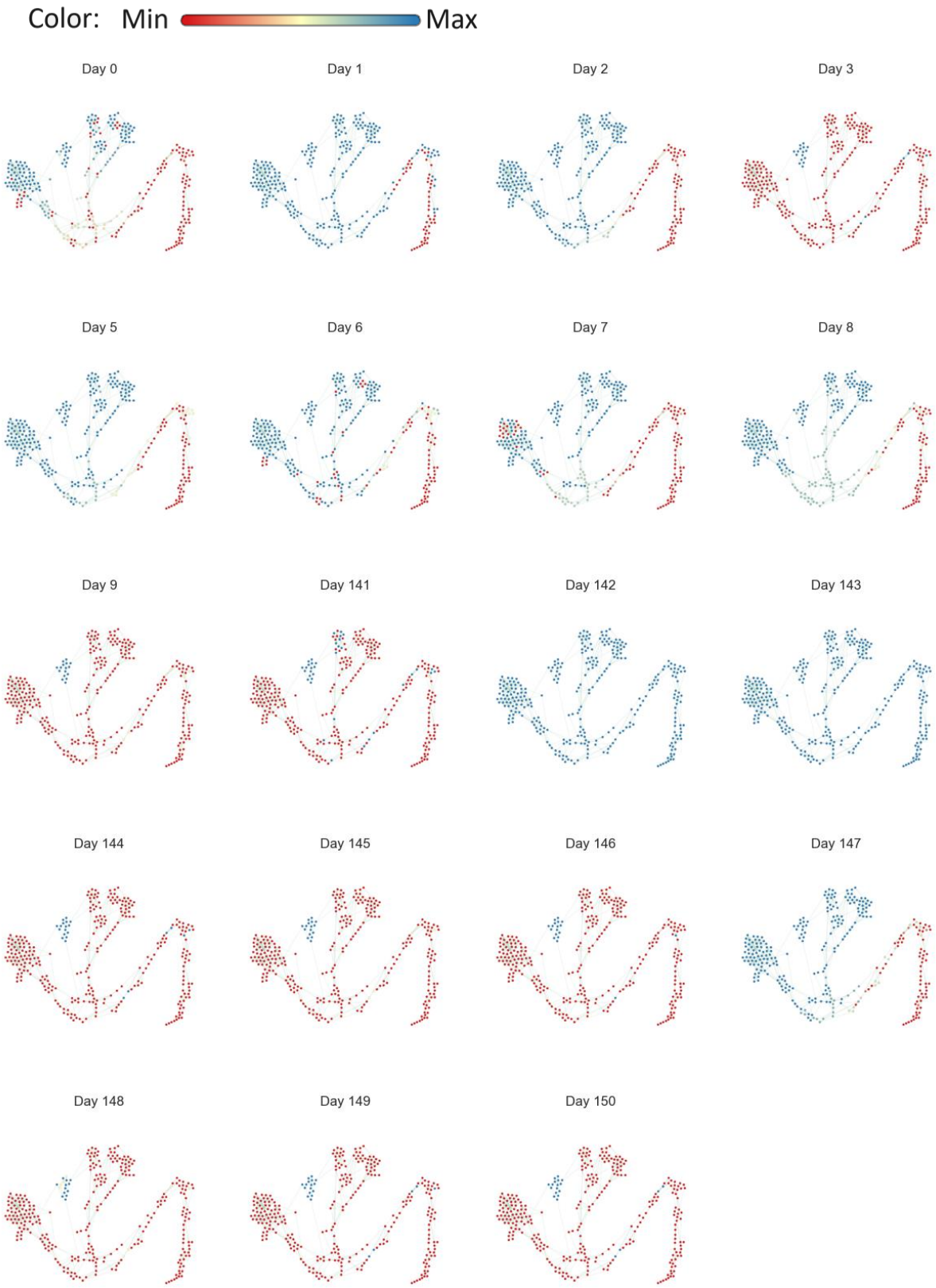
8.3.14 Deinococcaceae



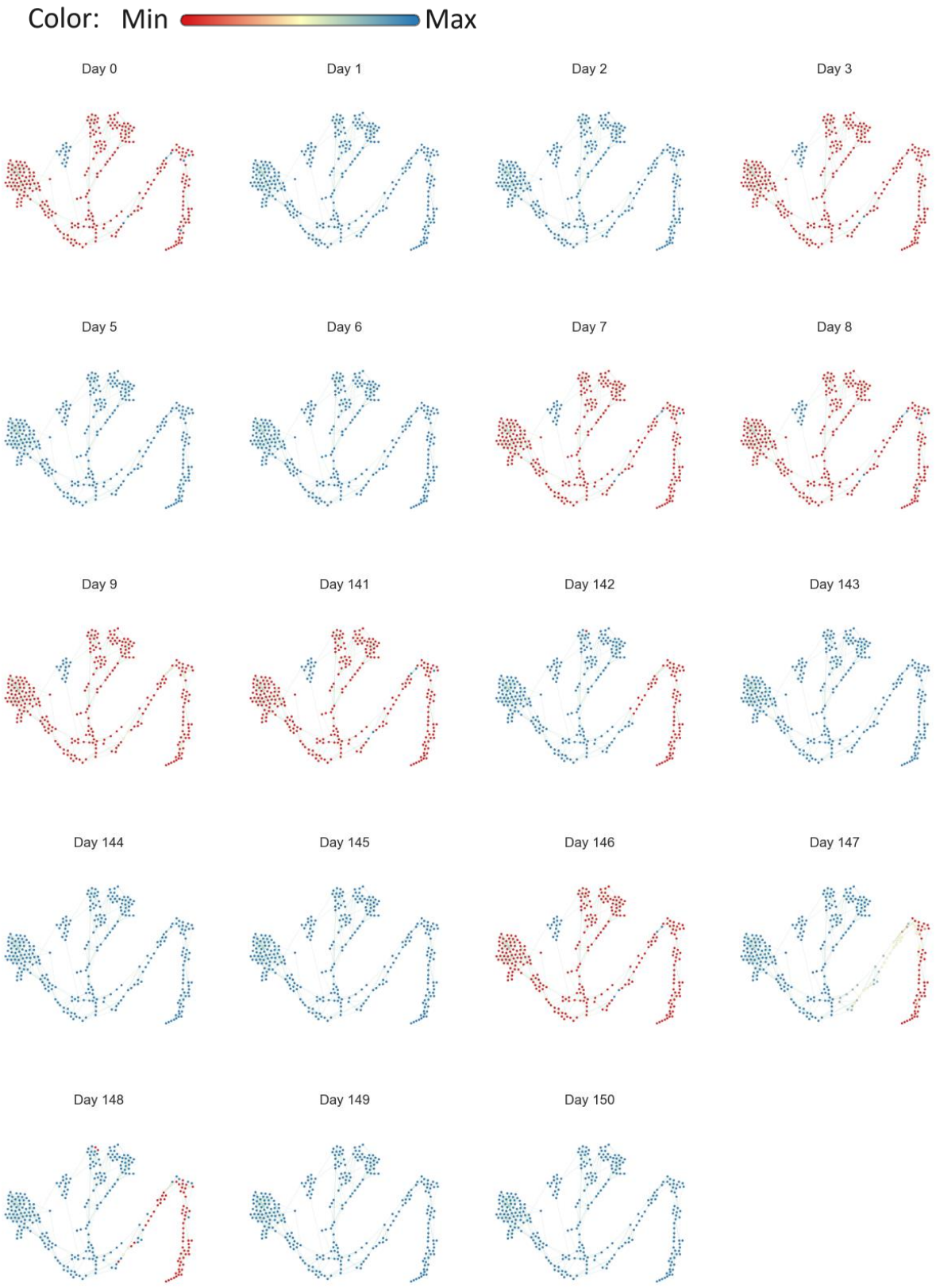
8.3.15 Eggerthellaceae



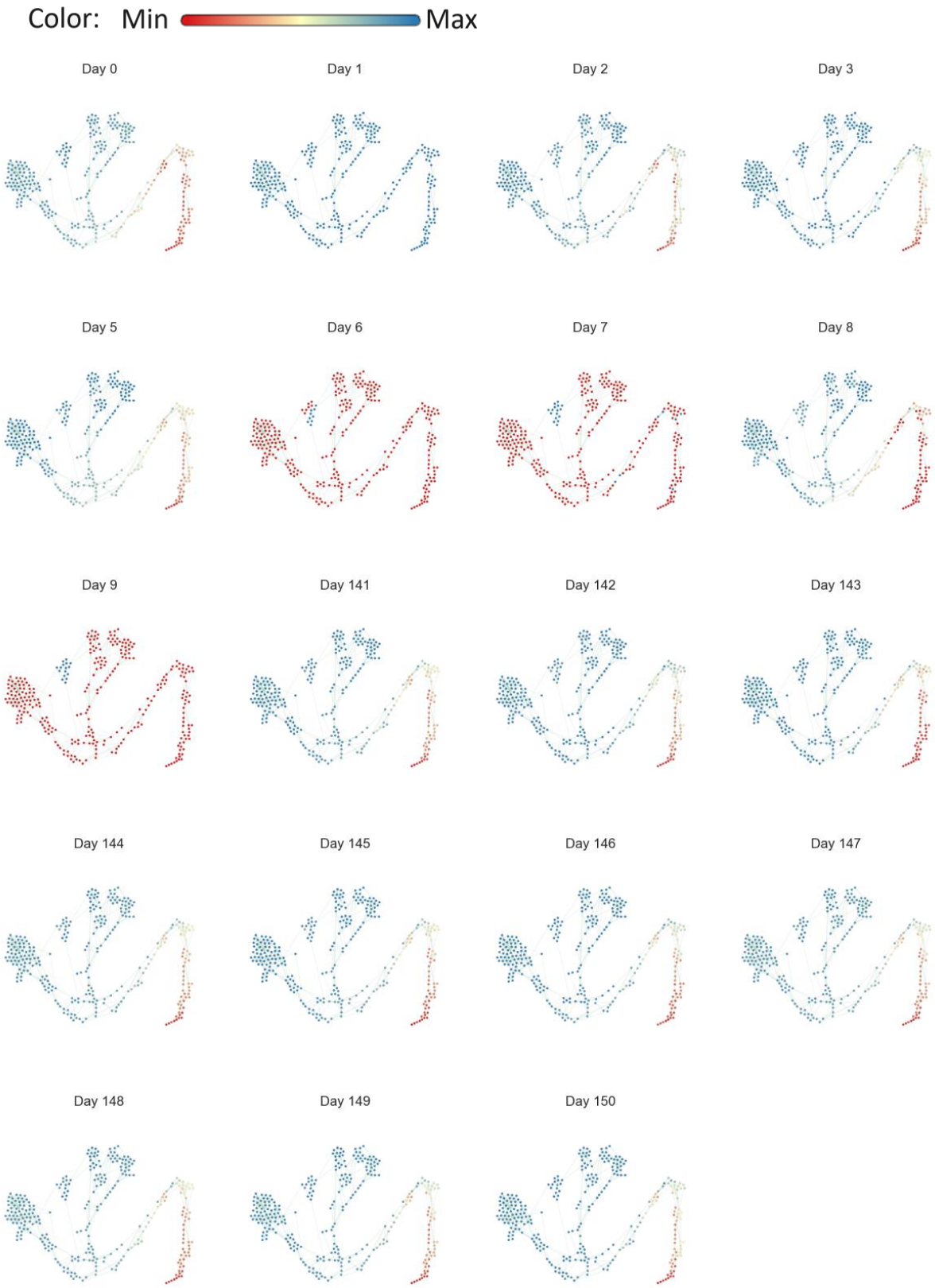
8.3.16 Enterobacteriaceae



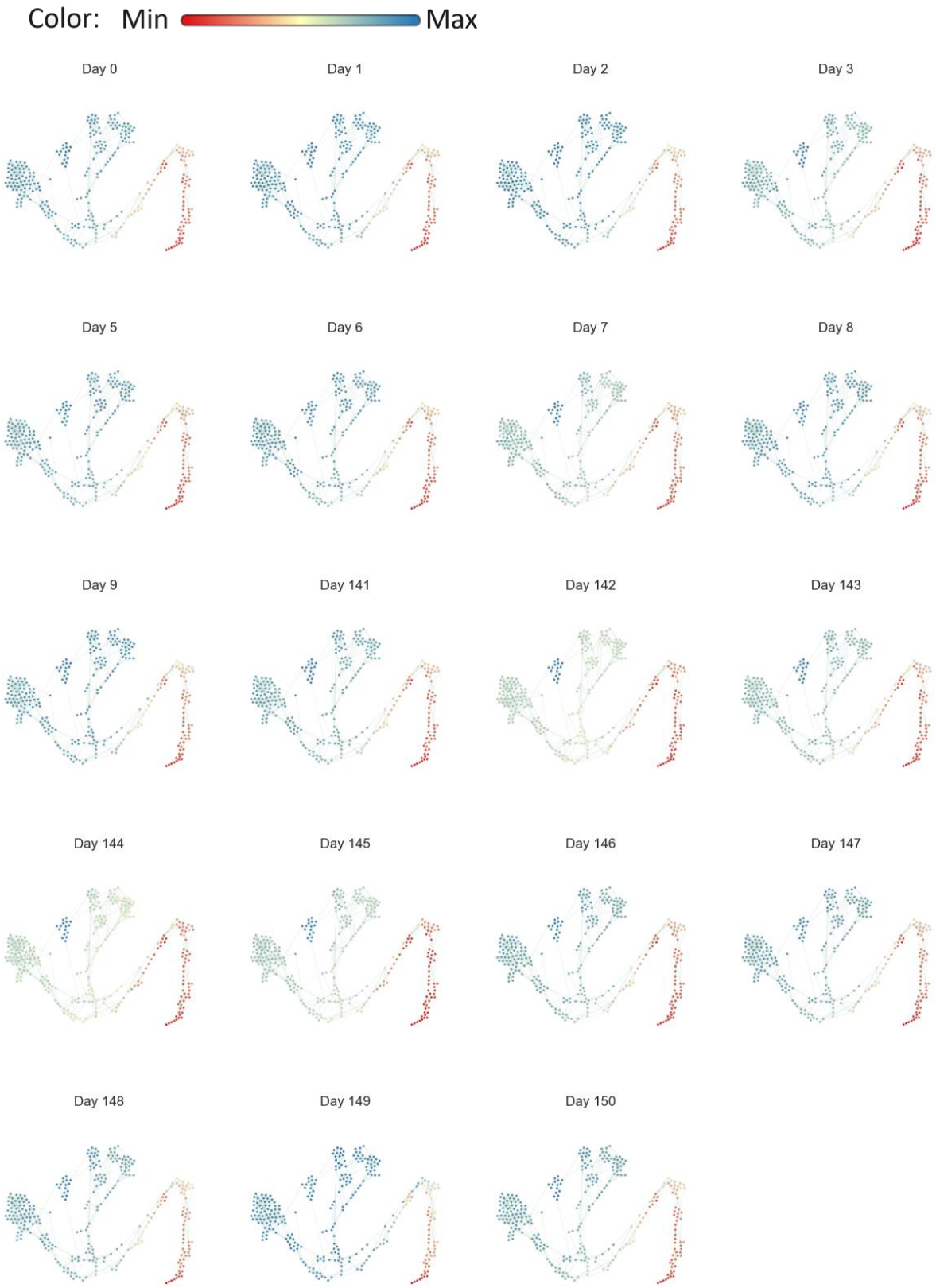
8.3.17 Erysipelatoclostridiaceae



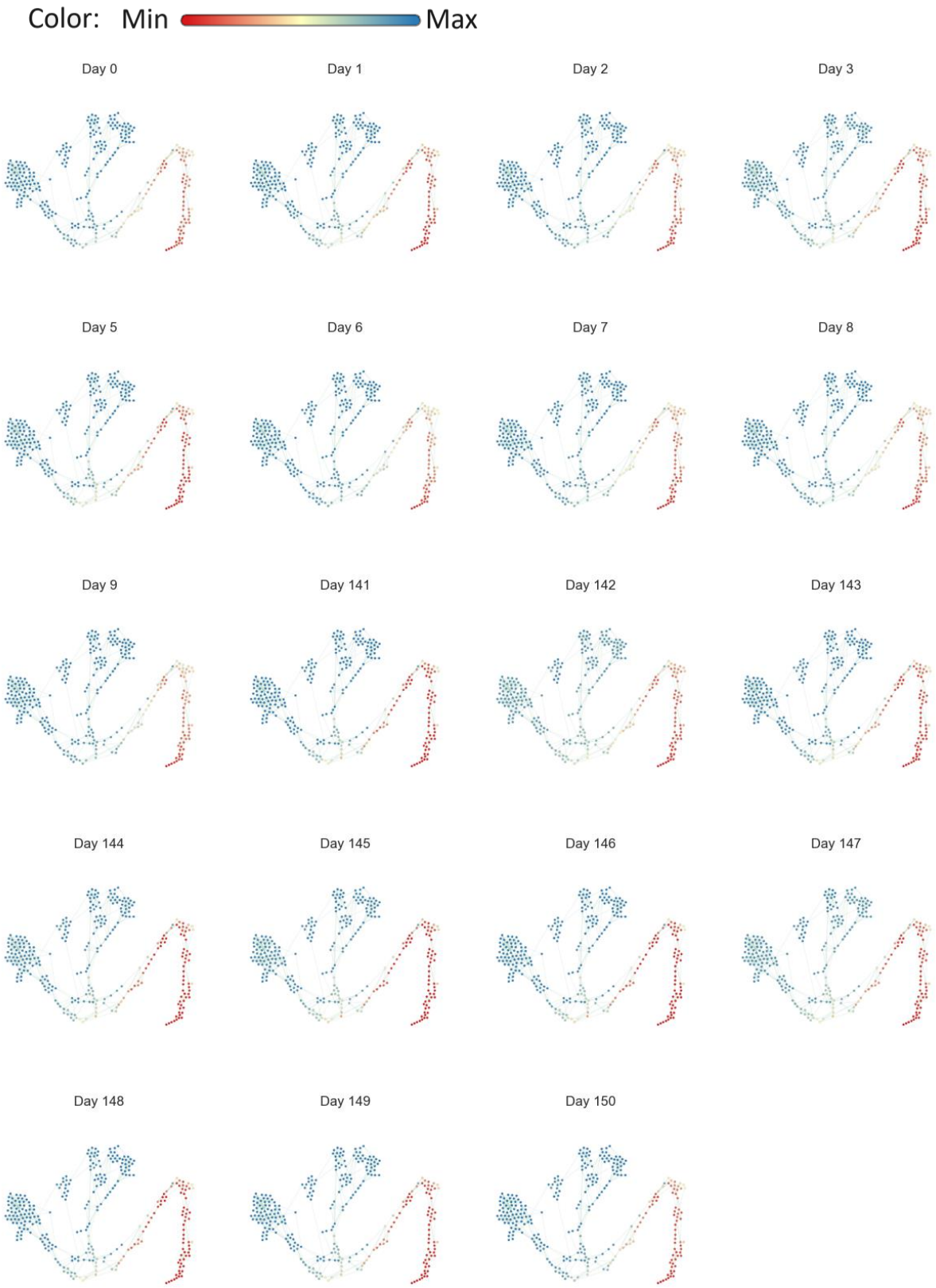
8.3.18 Erysipelotrichaceae



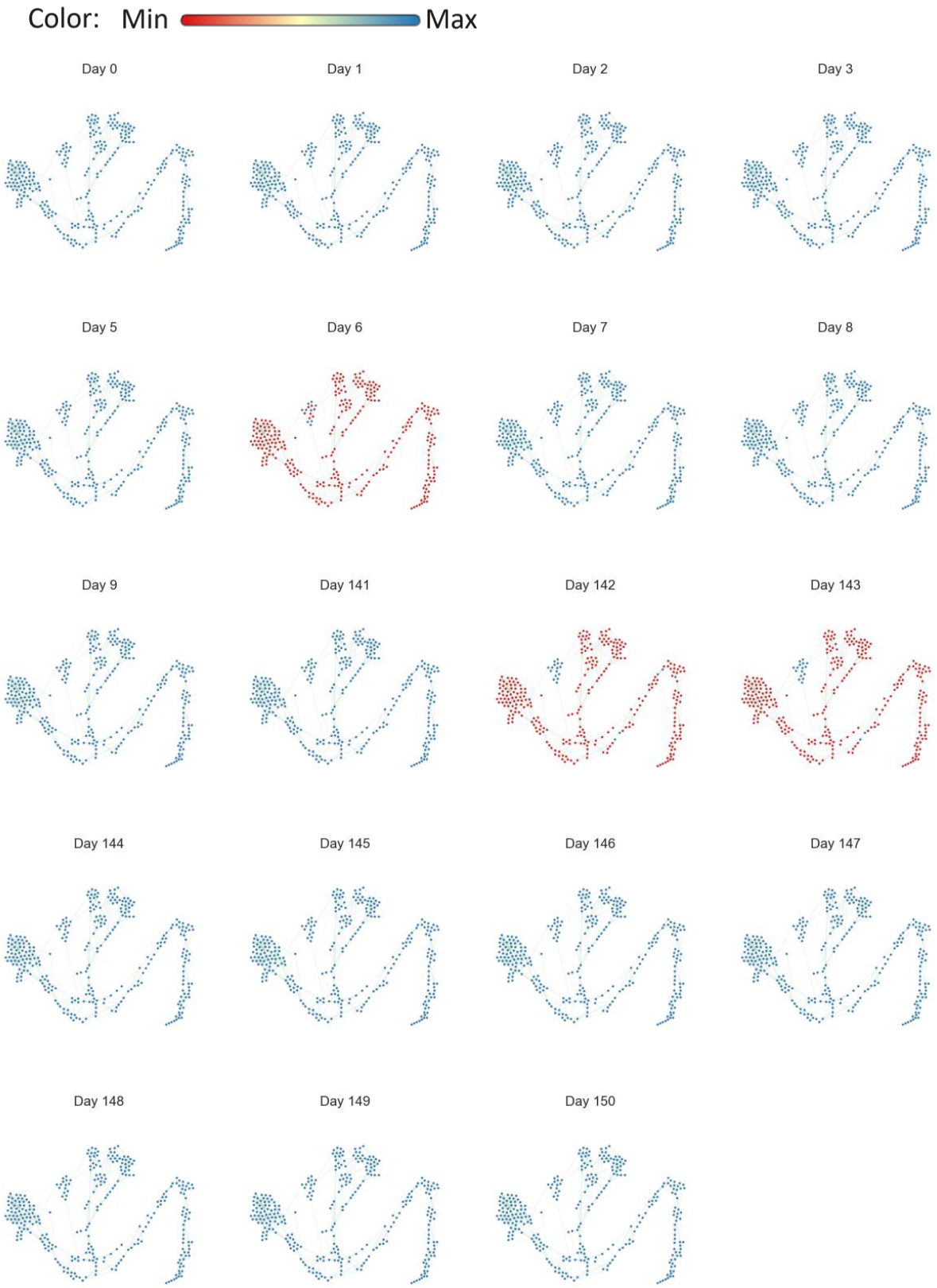
8.3.19 Lachnospiraceae



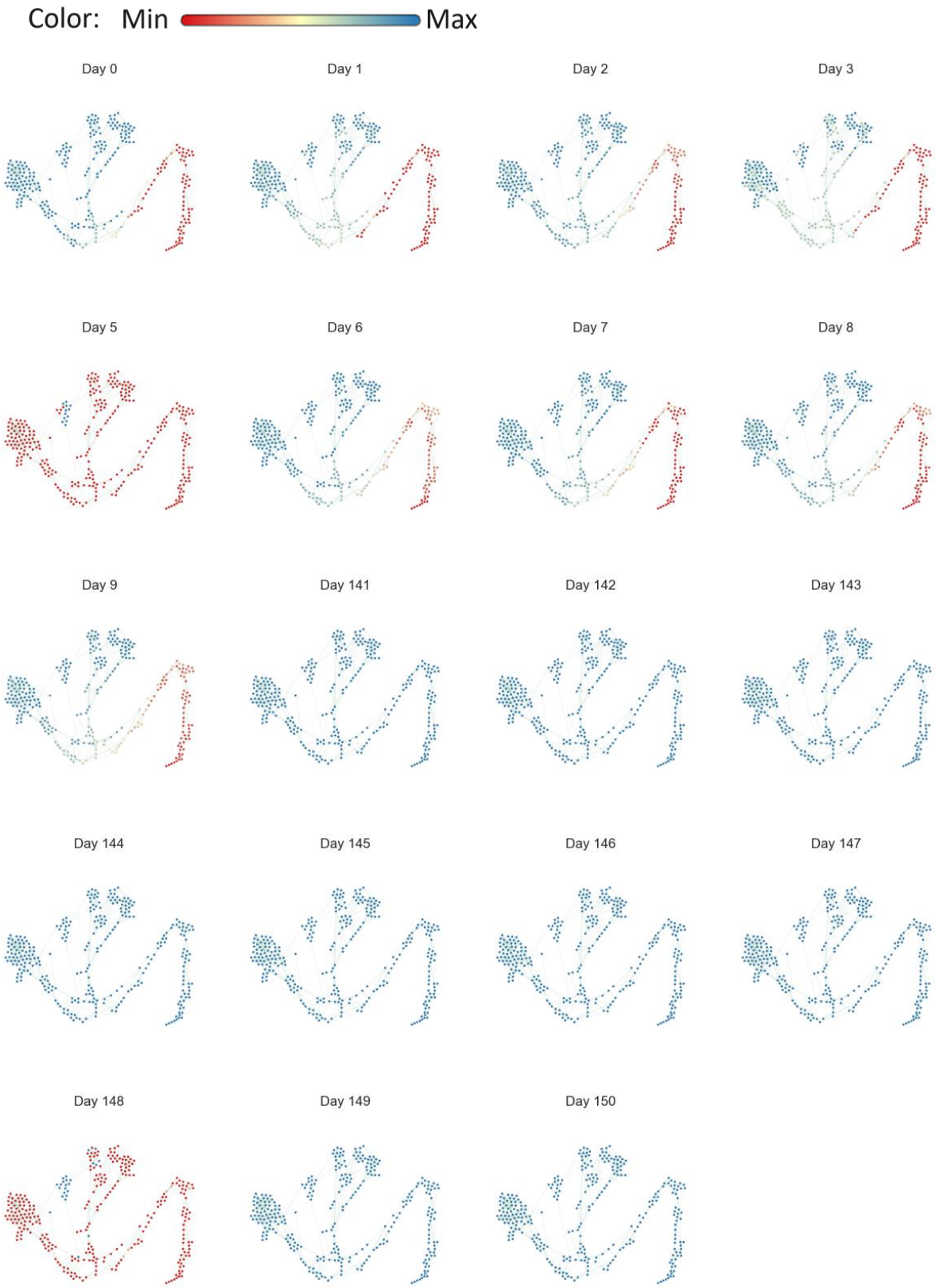
8.3.20 Lactobacillaceae



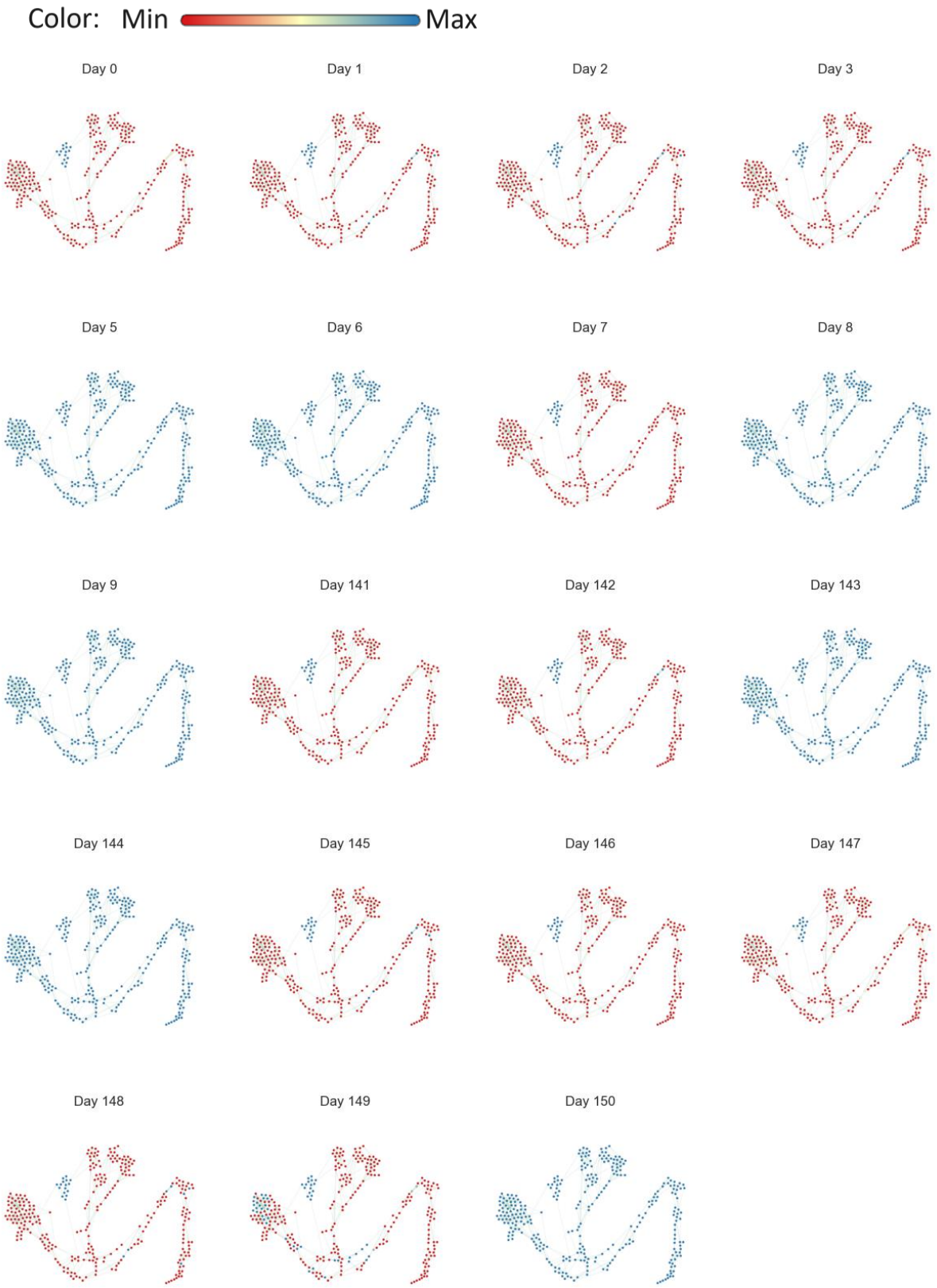
8.3.21 Listeriaceae



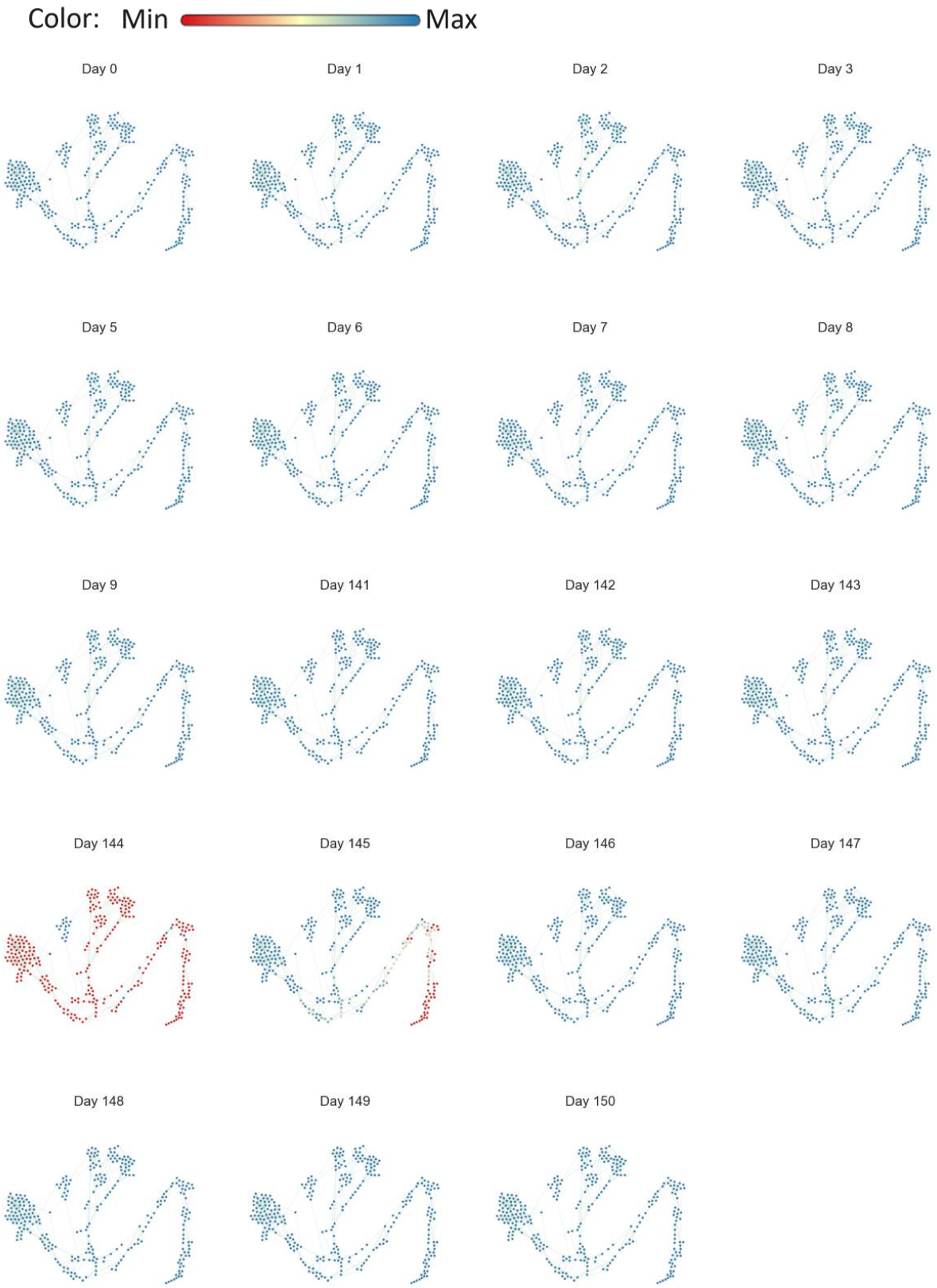
8.3.22 Mitochondria



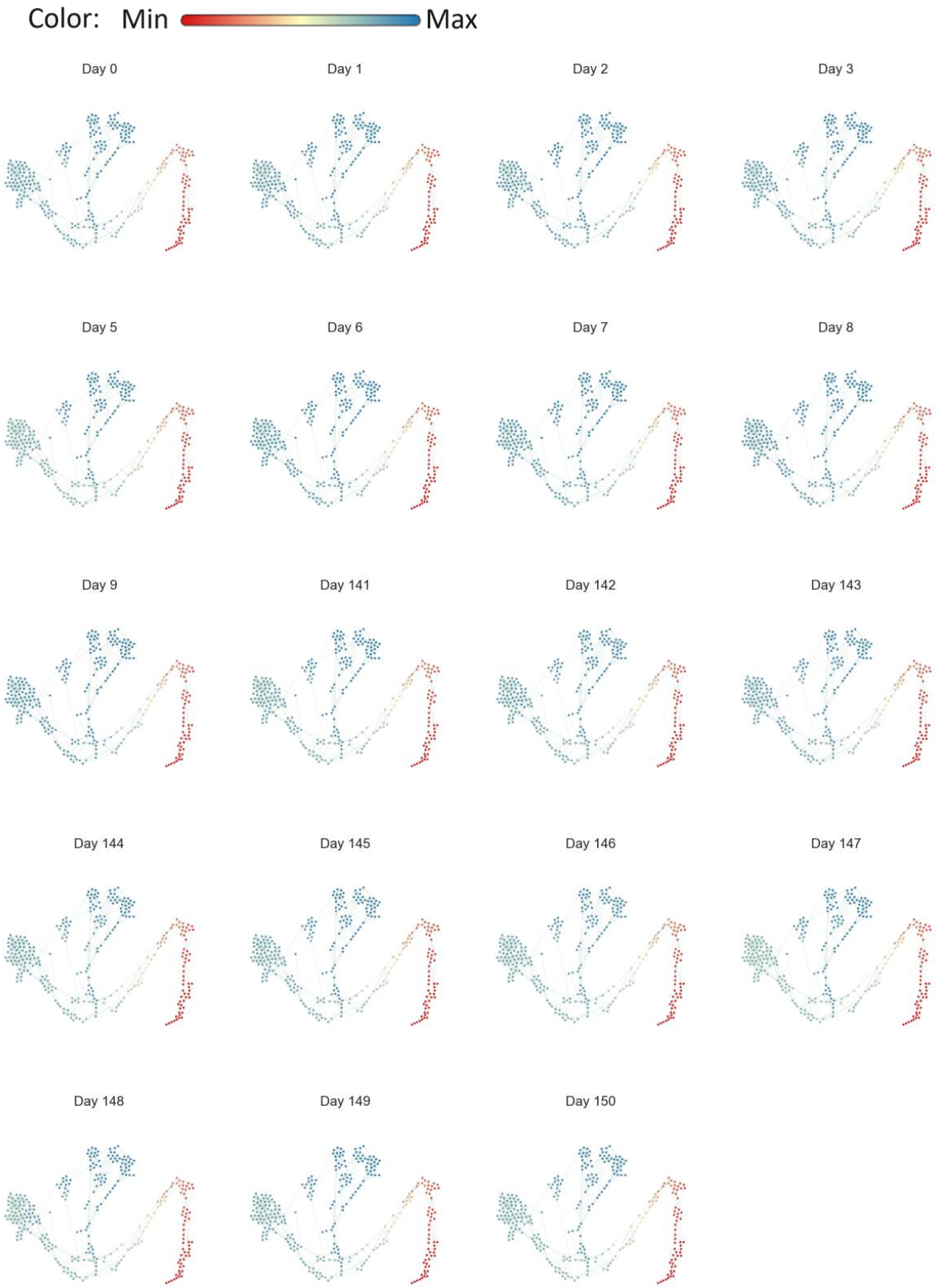
8.3.23 Monoglobaceae



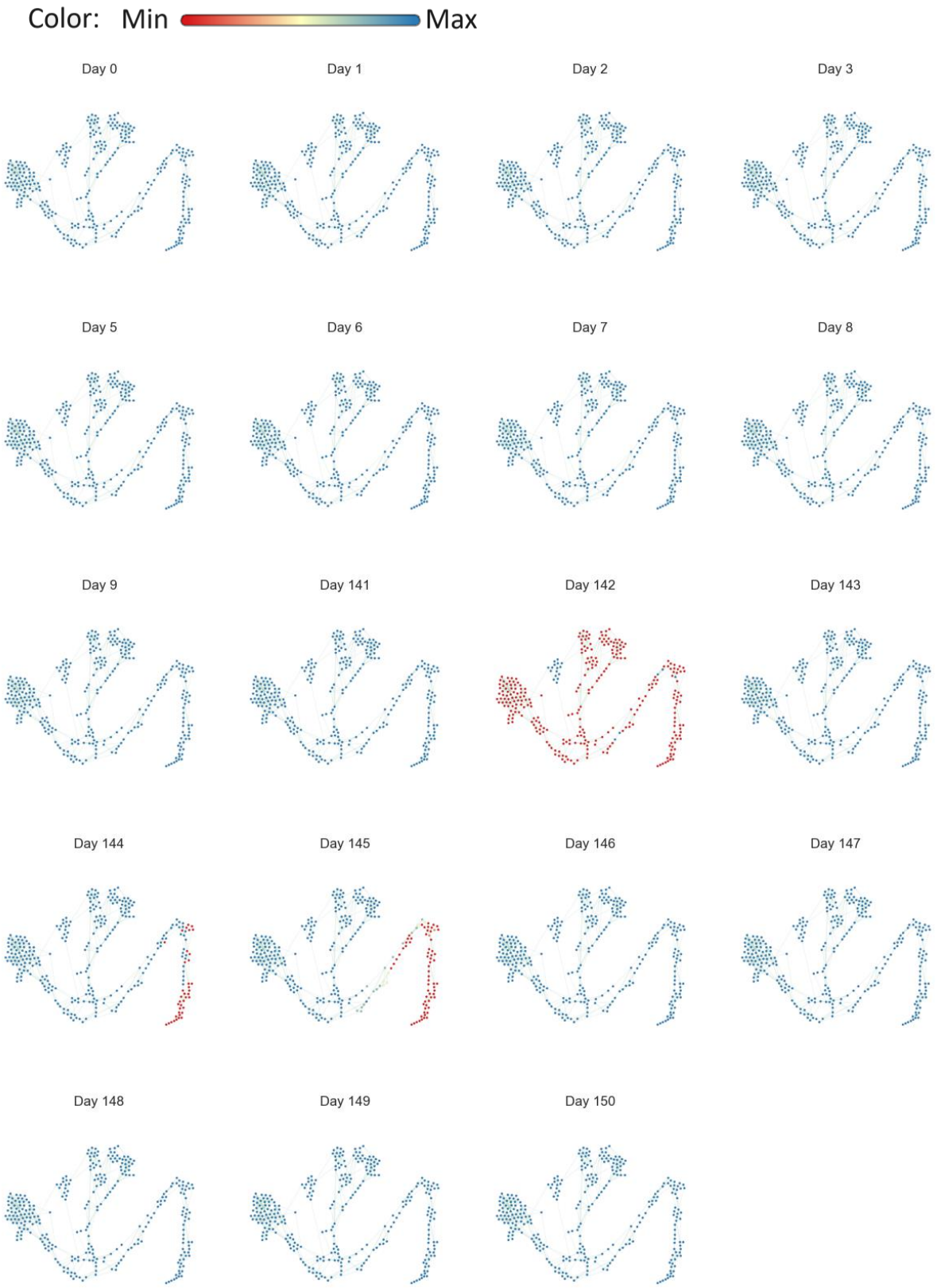
8.3.24 Moraxellaceae



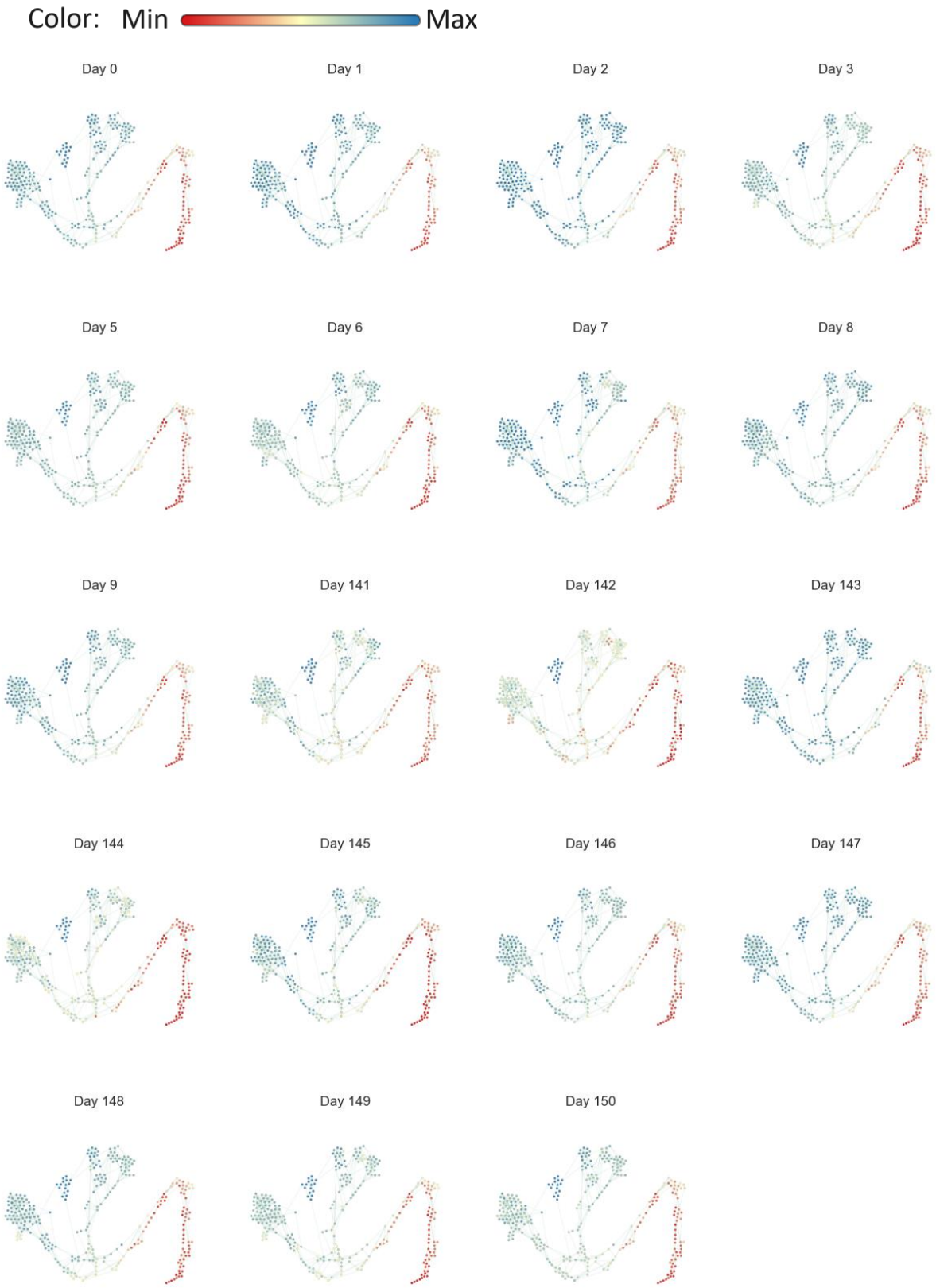
8.3.25 Muribaculaceae



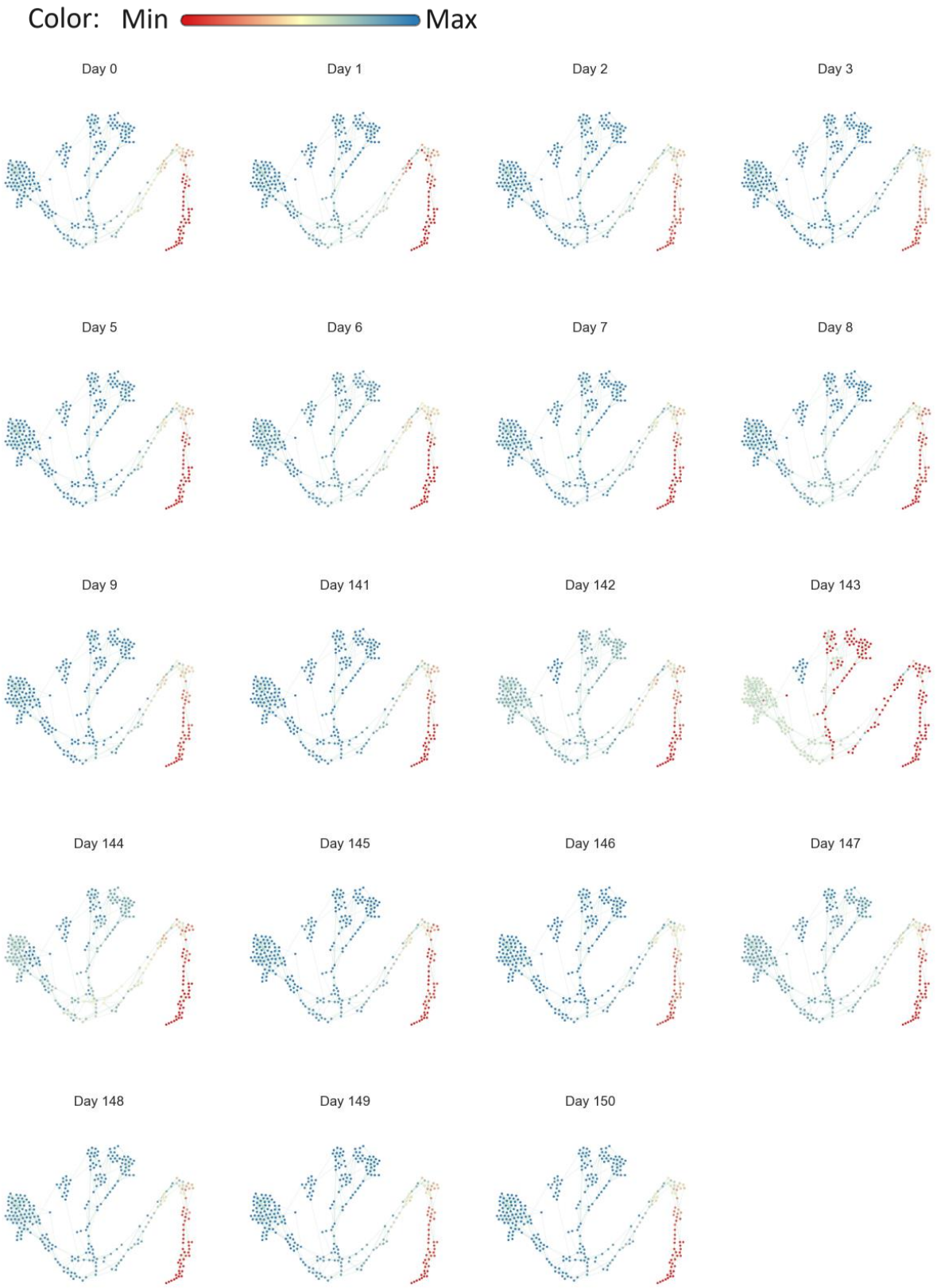
8.3.26 Neisseriaceae



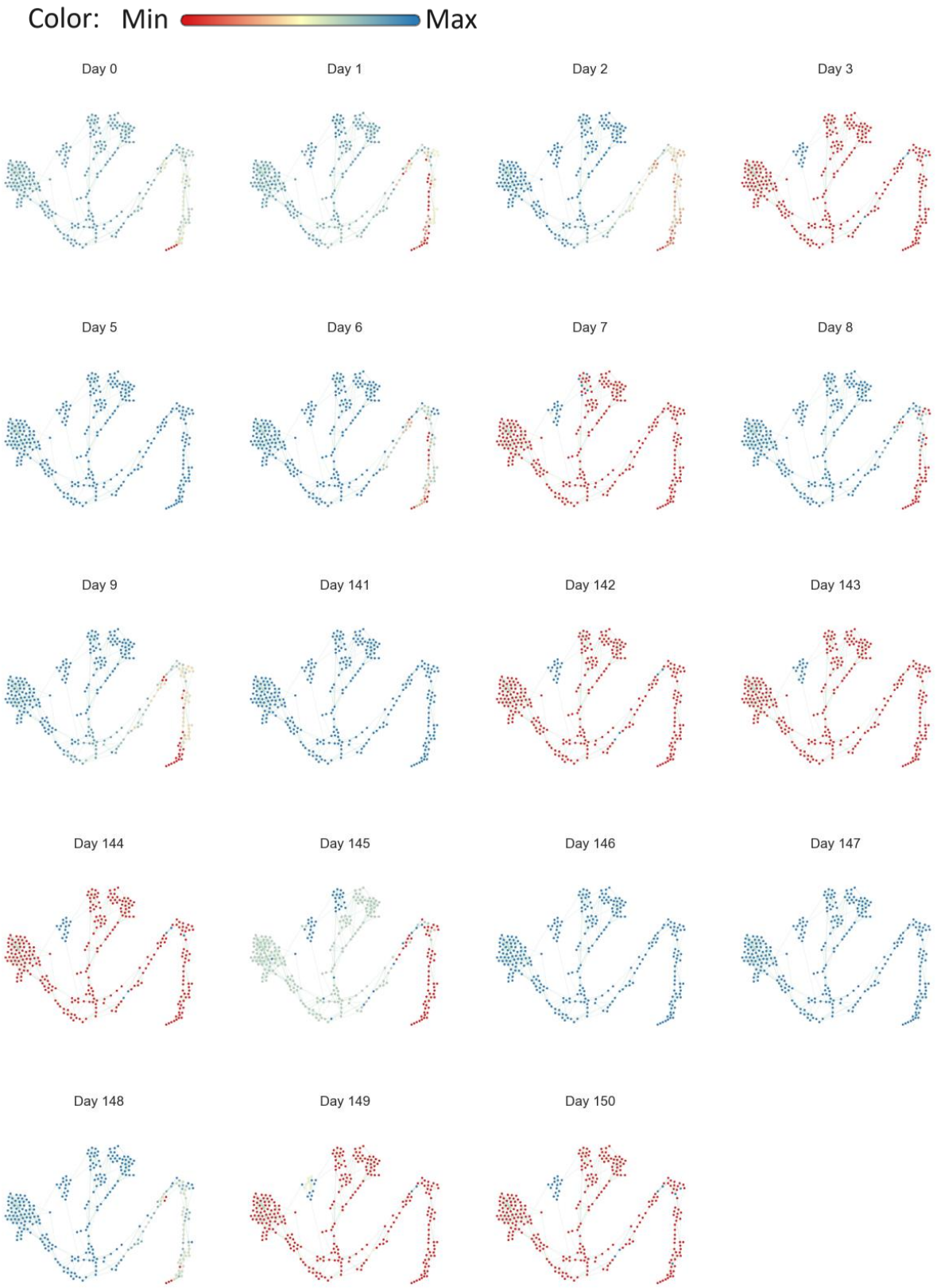
8.3.27 Oscillospiraceae



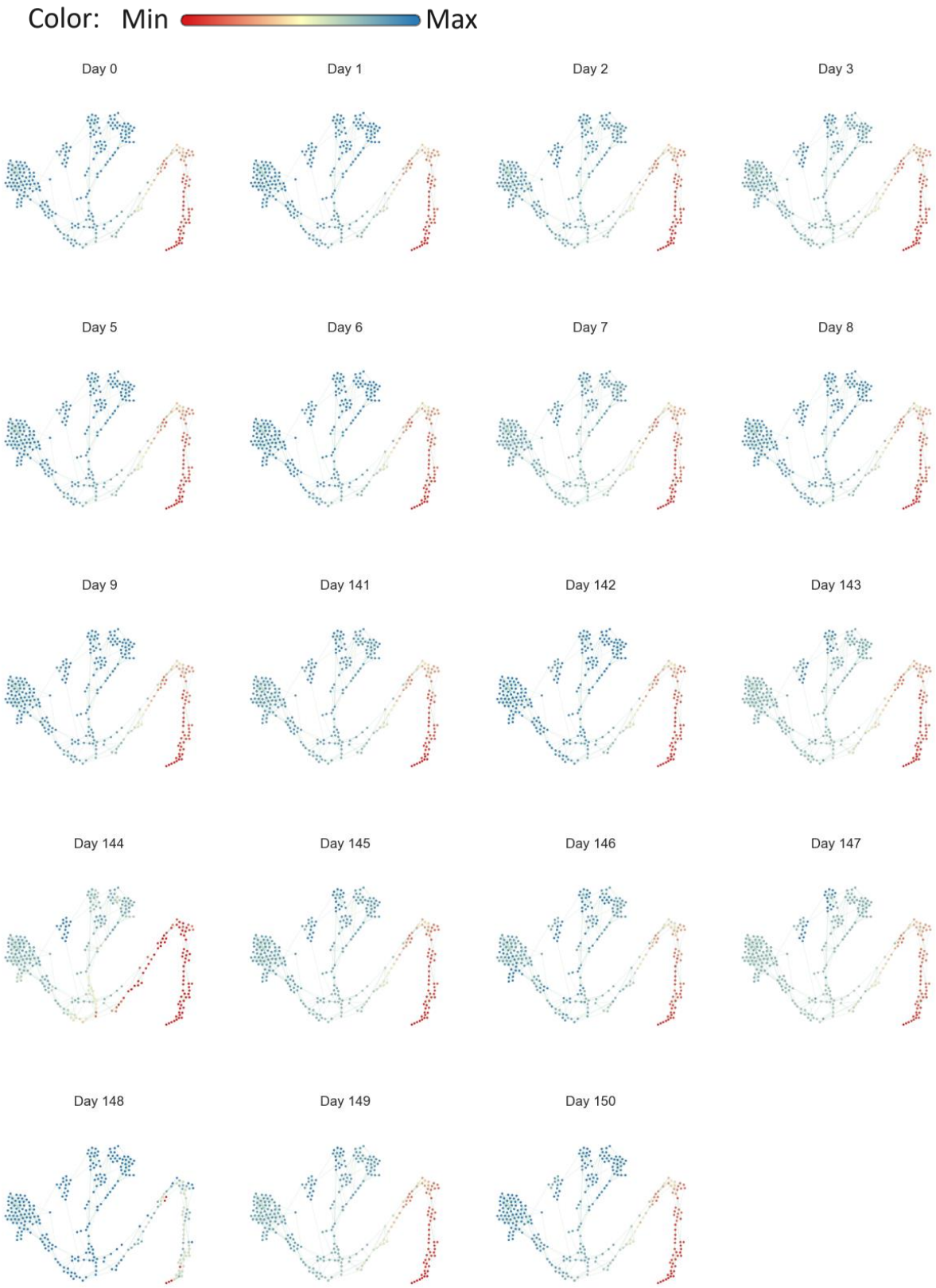
8.3.28 Peptococcaceae



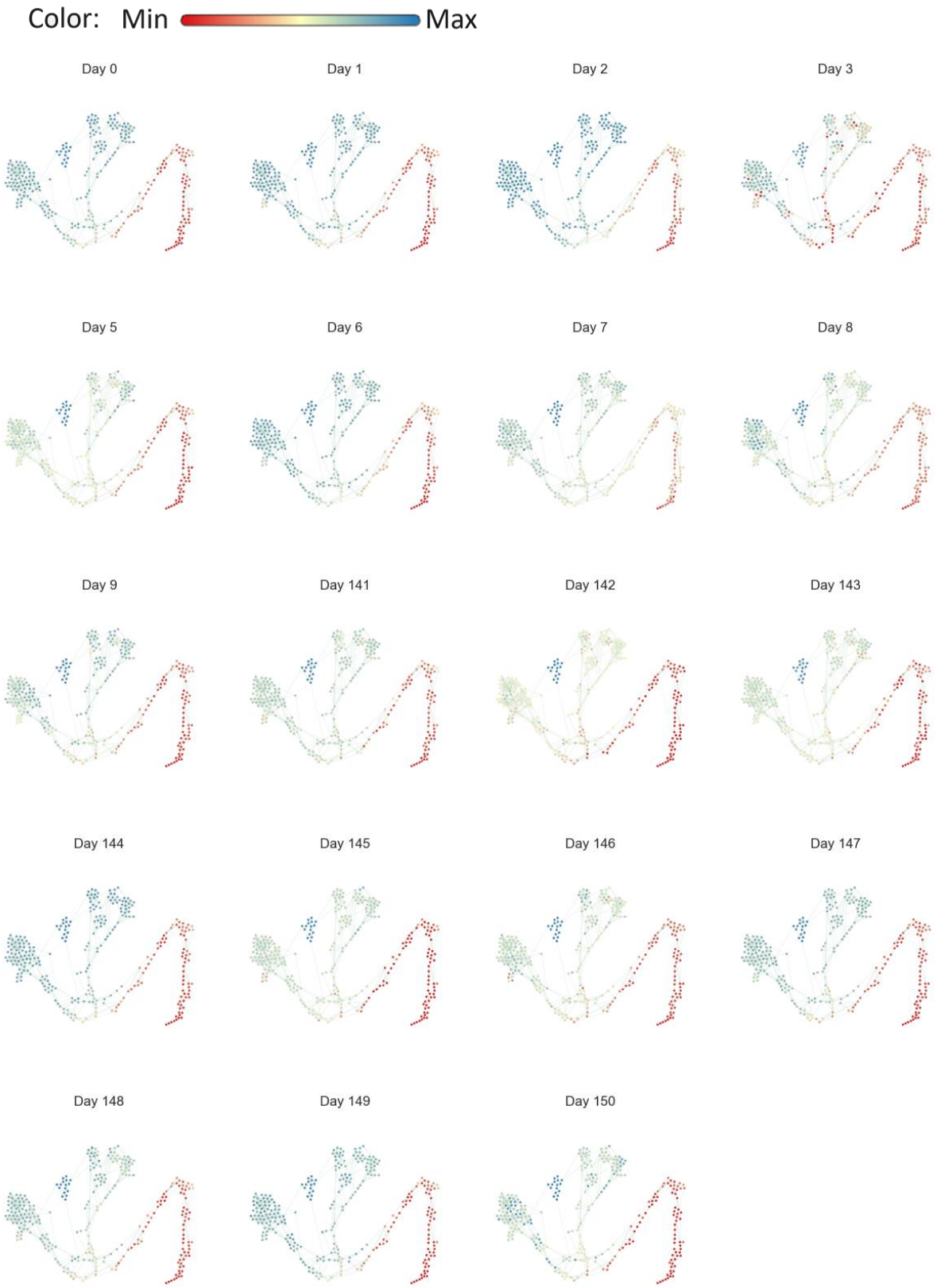
8.3.29 Pseudomonadaceae



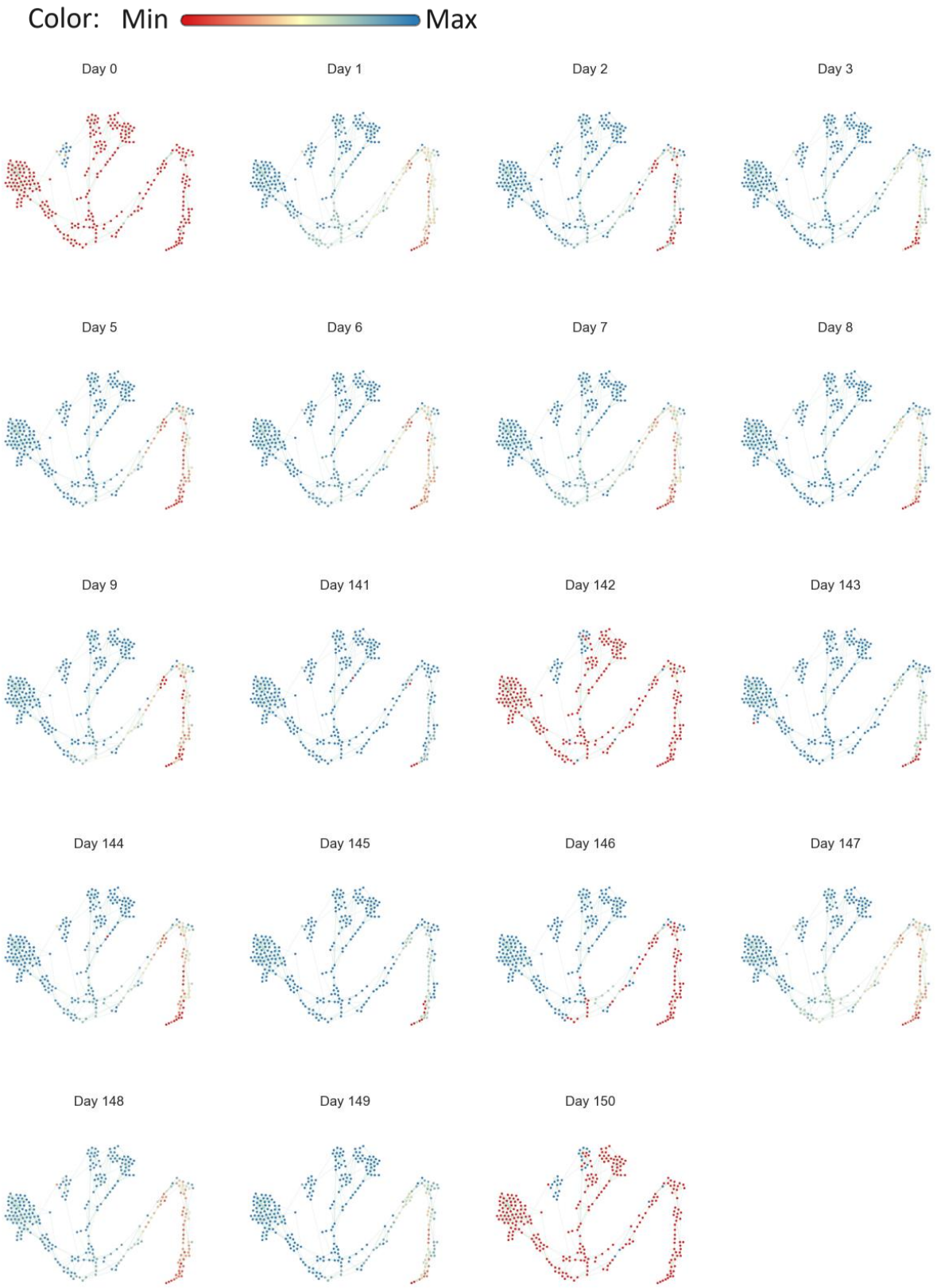
8.3.30 Rikenellaceae



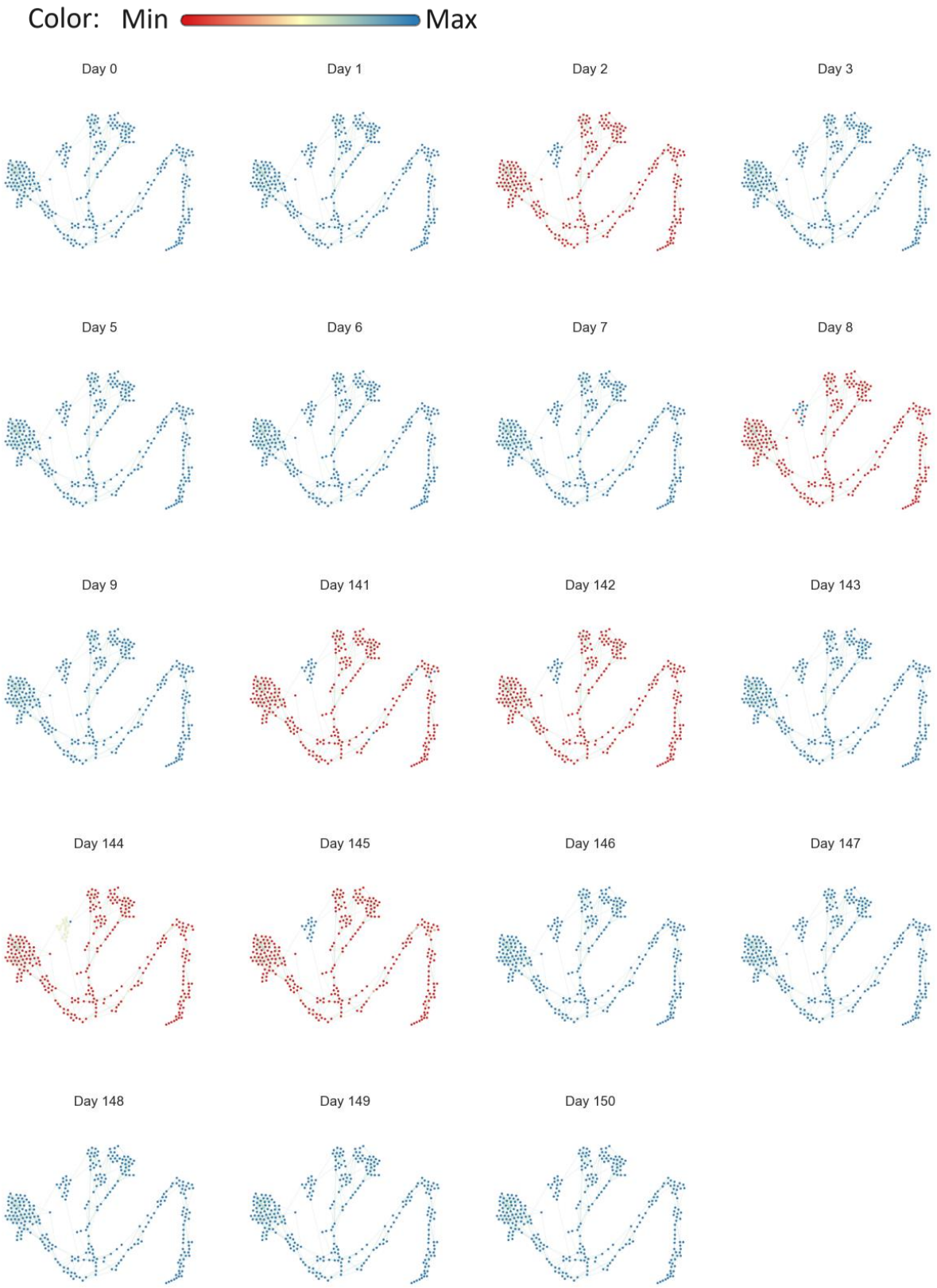
8.3.31 Ruminococcaceae



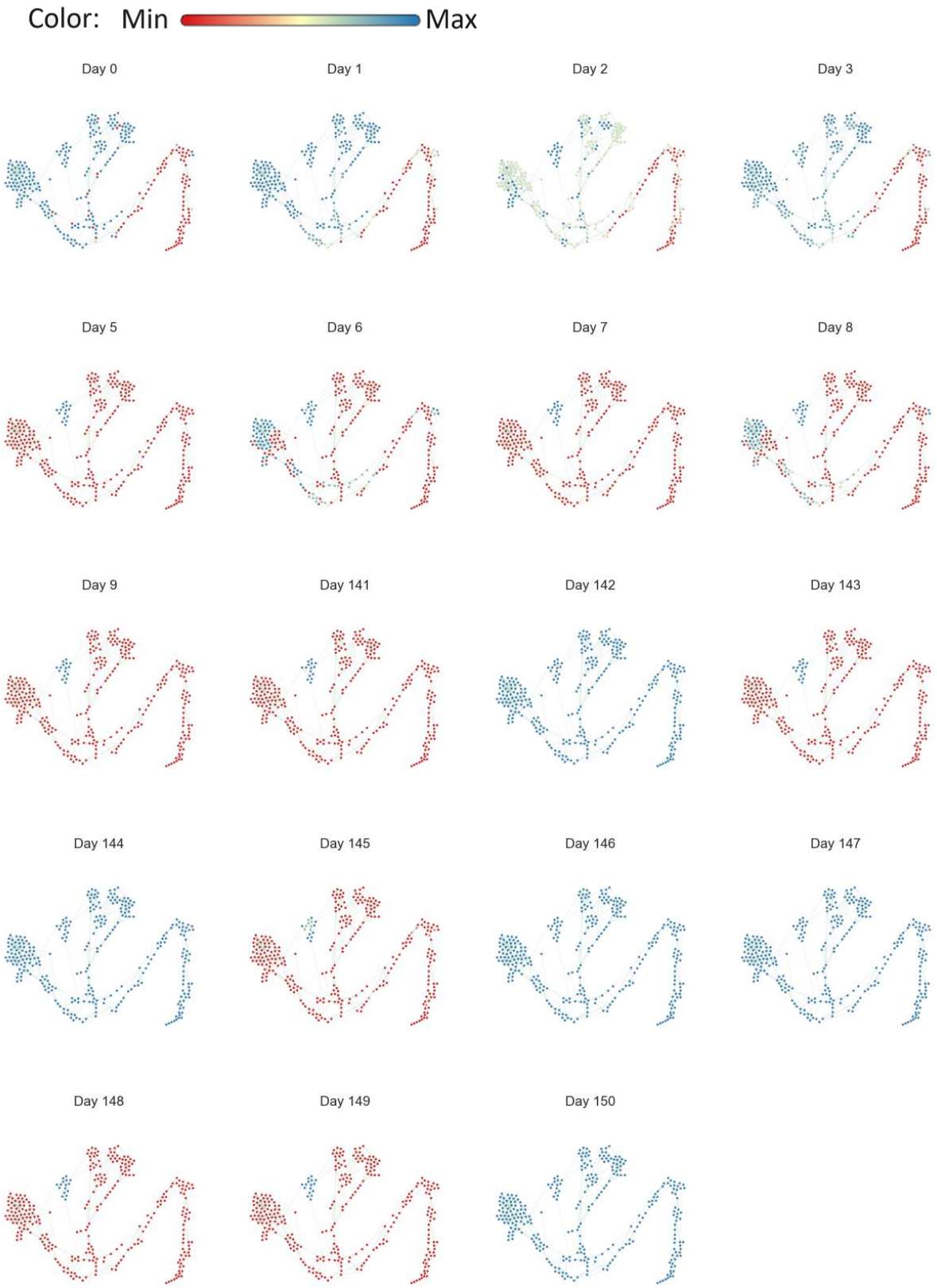
8.3.32 Saccharimonadaceae



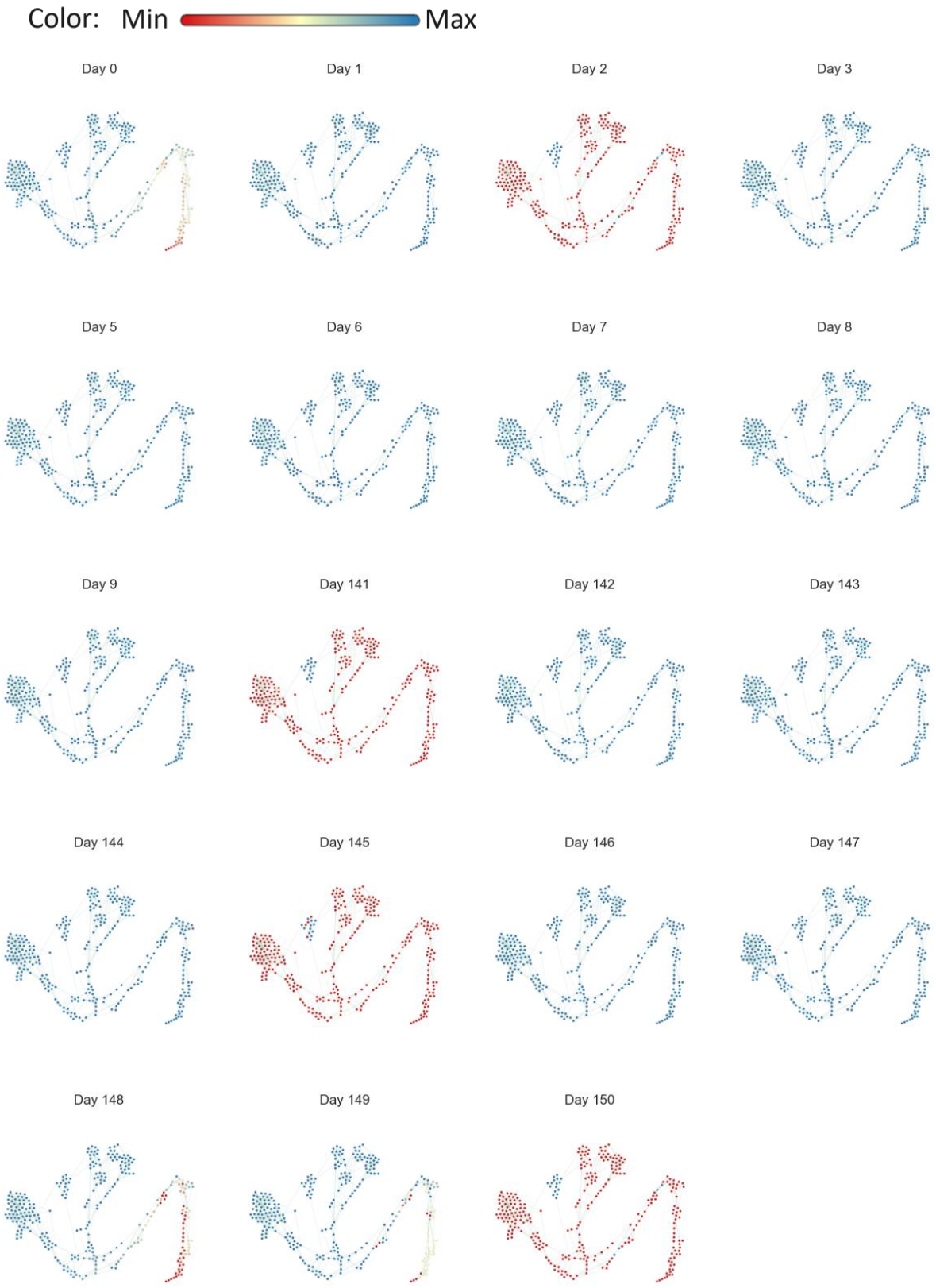
8.3.33 Staphylococcaceae



8.3.34 Streptococcaceae



8.3.35 [Eubacterium] coprostanoligenes group



8.4 Python Code

This code was wrote using Jupiter notebook, these are the content of cell copied one after each other from different notebooks. Not all the codes used for generating the plots is included.

```
import pandas as pd
from scipy.spatial.distance import pdist, squareform
from scipy.sparse import csr_matrix
from scipy.sparse.csgraph import minimum_spanning_tree
import numpy as np
import random
import csv
from returns.pipeline import flow
import os
from pprint import pprint
import pandas as pd
import plotly.express as px
from matplotlib import pyplot as plt

def find_directories(directory,file_name):
    directories_with_test_csv = []
    for root, dirs, files in os.walk(directory):
        if file_name  in files:
            directories_with_test_csv.append(root)
    return directories_with_test_csv

MiSeq_directories= find_directories("C:\\Data
Science\\UHasselt\\Thesis\\data_Jannes\\Results\\MiSeq_SOP","ASV_table.csv
")
print(f"Number of available output for MiSeq
dataset:{len(MiSeq_directories)}")
MiSeq_directories.remove("C:\\Data
Science\\UHasselt\\Thesis\\data_Jannes\\Results\\MiSeq_SOP\\MiSeq_SOP_run3
\\sampleID_165")
print(f"Number of available output for MiSeq
dataset:{len(MiSeq_directories)}")

pipeline_indexes=[]
for directory in MiSeq_directories:
    index=-1
    while directory[index:].isnumeric():
        index=index-1
    if "run1" in directory:
        pipeline_indexes.append(int(directory[index+1:]))
    elif "run2" in directory:
        pipeline_indexes.append(int(directory[index+1:])+100)
    else:
        pipeline_indexes.append(int(directory[index+1:])+400)

def find_directories(directory,file_name):
    directories_with_test_csv = []
    for root, dirs, files in os.walk(directory):
        if file_name  in files:
            directories_with_test_csv.append(root)
    return directories_with_test_csv

MDAW_directories = find_directories_with_test_csv(f"F:\Statistics Data
Science\Thesis\data_Jannes\Results\MDAW",
                                                  "ASV_table.csv")
print(f"Number of available output for MADW
dataset:{len(MDAW_directories)}")
MiSeq_directories= find_directories_with_test_csv(f"F:\Statistics Data
Science\Thesis\data_Jannes\Results\MiSeq_SOP",
                                                  "ASV_table.csv")
print(f"Number of available output for MiSeq
dataset:{len(MiSeq_directories)}")
```

```
def make_dataframes(directories,file_name):
    dataframes=[]
    for directory in directories:
        try:
            dataframes.append(pd.read_csv(directory+"\\"+file_name))
        except pd.errors.EmptyDataError:
            print(f"The file is empty in directory {directory}")
    print(f"Number of available dataframes for
{file_name}:{len(dataframes)}")
    return dataframes

MiSeq_ASV_dataframes=make_dataframes(MiSeq_directories,"ASV_table.csv")
MiSeq_count_dataframes=make_dataframes(MiSeq_directories,"seq_count_table.
csv")

def get_DNA_list(dataframes,category):
    if category==1:
        DNA_list=[]
        for dataframe in dataframes:
            dataframe.columns=['DNA']+dataframe.columns[1:].tolist()
            DNA_list.extend(dataframe['DNA'].tolist())
    else:
        DNA_list=[]
        for dataframe in dataframes:
            DNA_list.extend(dataframe.columns[1:])
    return DNA_list

count = collections.Counter(MiSeq_DNA2)
counter=0
freq=[]
total=0
for k,v in count.items():
    if v>2:
        freq.append(v)
        counter=counter+

def make_dataframes(directories):
    count_dataframes=[]
    ASV_dataframes=[]
    for directory in directories:
        try:

ASV_dataframes.append(pd.read_csv(directory+"\\ASV_table.csv"))

count_dataframes.append(pd.read_csv(directory+"\\seq_count_table.csv"))
            count_dataframes[-1].drop(count_dataframes[-
1].tail(1).index,inplace=True)
        except pd.errors.EmptyDataError:
            print(f"The file is empty in directory {directory}")
    return ASV_dataframes,count_dataframes

records_list=[]
for i in range(len(count_dataframes)):

ratio_list.append(round(ASV_dataframes[i].shape[0]/count_dataframes[i].sha
pe[1]*100,2))

fig,ax=plt.subplots()
ax.boxplot(ratio_list)
ax.set_title("Rattio of sequences with hierarch")
fig.show()

ratio_list=[]
for i in range(len(count_dataframes)):

ratio_list.append(round(ASV_dataframes[i].shape[0]/count_dataframes[i].sha
pe[1]*100,2))
```

```
fig,ax=plt.subplots()
ax.boxplot(ratio_list)
ax.set_title("Rattio of sequences with hierarch")
fig.show()

for dataframe in ASV_dataframes:
    for column in dataframe.columns[1:]:
        dataframe[column][dataframe[column].notnull()]=column
        dataframe[column][dataframe[column].isnull()]='NA'

unique_values={}
for dataframe in ASV_dataframes:
    for column in dataframe.columns[1:]:
        unique_values[column]=unique_values.get(column,[])

unique_values[column].append(len(set(dataframe[column][dataframe[column].n
otnull()])))

for column in unique_values.keys():
    fig,ax=plt.subplots()
    ax.boxplot(unique_values[column])
    ax.set_title(column)
    fig.show()

ASV_dataframes_count=len(ASV_dataframes)
summary={}
counter=0
for dataframe in ASV_dataframes:
    fig = px.sunburst(dataframe, path=list(dataframe.columns)[1:])
    for index,_id in enumerate(fig.data[0].ids):
        summary[_id]=summary.get(_id,[0]*ASV_dataframes_count)
        summary[_id][counter]=fig.data[0].values[index]
    counter=counter+1

family_groups=[]
for index,dataframe in enumerate(ASV_dataframes):
    family_groups.append({})
    for category in dataframe["Family"].unique():
        family_groups[index][category]=
list(dataframe[dataframe["Family"]==category].iloc[:,0])

family_count=[]
for index,dataset in enumerate(family_groups):
    family_count.append({})
    family_count[index]["ID"]=list(count_dataframes[0].iloc[:,0])
    for category in list(dataset.keys()):
        if type(category) is str:

family_count[index][category]=list(count_dataframes[index][dataset[categor
y]].sum(axis=1))
            if not any(family_count[index][category]):
                family_count[index].pop(category)

all_family=set()
for dataset in family_count:
    for category in dataset.keys():
        all_family.add(category)

for index,dataset in enumerate(family_count):
    for family in all_family:
        if not family in dataset.keys():
            dataset[family]=0
    pd.DataFrame(dataset).to_csv(MiSeq_directories[index] +
f'\Family2_ASV.csv', index=False)

for index,dataset in enumerate(family_count):
    df=pd.DataFrame(dataset)
    df.loc[:, (df!= 0).any(axis=0)].to_csv(MiSeq_directories[index] +
f'\Family2_ASV.csv', index=False)
```

```
def alter_distances(distances, max_amount = 0.01):
    changer = lambda t: t + random.uniform(0,max_amount)
    return np.array([changer(d)for d in distances])

def calculate_mst(distances):
    X = csr_matrix(squareform(distances))
    mst = minimum_spanning_tree(X)
    return np.nonzero(mst)

def add_links_to_file(filename, links, pipeline_indexes,indicator=0):
    with open(filename, 'a', newline='') as csvfile:
        writer = csv.writer(csvfile,delimiter=',')
        for i in range(0,len(links[0])):

writer.writerow([pipeline_indexes[links[0][i]],pipeline_indexes[links[1][i
]],indicator])

for i in range(19):
    df=pd.read_csv("C:\\Data
Science\\UHasselt\\Thesis\\data_Jannes\\Results\\MiSeq_SOP\\Samples\\Famil
y_sample"+str(i+1)+".csv")
    distances = pdist(df,'braycurtis')
    for j in range(1):
        flow(distances,lambda d: calculate_mst(d),lambda d:
add_links_to_file("C:\\Data
Science\\UHasselt\\Thesis\\data_Jannes\\Results\\MiSeq_SOP\\Family_network
_sample"+str(i+1)+".csv",d,pipeline_indexes))

for i in range(19):
    df=pd.read_csv("C:\\Data
Science\\UHasselt\\Thesis\\data_Jannes\\Results\\MiSeq_SOP\\Samples\\Genus
_sample"+str(i+1)+".csv")
    distances = pdist(df,'braycurtis')
    distances
    for j in range(1):
        flow(distances,lambda d: calculate_mst(d),lambda d:
add_links_to_file("C:\\Data
Science\\UHasselt\\Thesis\\data_Jannes\\Results\\MiSeq_SOP\\Genus_network_
sample"+str(i+1)+".csv",d,pipeline_indexes))

with open("C:\\Data
Science\\UHasselt\\Thesis\\data_Jannes\\Results\\MiSeq_SOP\\Family_network
_all.csv", 'a', newline='') as csvfile:
        writer = csv.writer(csvfile,delimiter=',')
        writer.writerow(["Source","Target","indicator"])
for i in range(19):
    df=pd.read_csv("C:\\Data
Science\\UHasselt\\Thesis\\data_Jannes\\Results\\MiSeq_SOP\\Samples\\Famil
y_sample"+str(i+1)+".csv")
    distances = pdist(df,'braycurtis')
    for j in range(1):
        flow(distances,lambda d: calculate_mst(d),lambda d:
add_links_to_file("C:\\Data
Science\\UHasselt\\Thesis\\data_Jannes\\Results\\MiSeq_SOP\\Family_network
_all.csv",d,pipeline_indexes,i+1))


pipelines=pd.read_csv("C:\\Data
Science\\UHasselt\\Thesis\\data_Jannes\\meta\\MiSeq_SOP\\all_sampled_param
s_MiSeq_SOP.csv")
pipelines.index = range(1, len(pipelines) + 1)

pipelines.loc[pipeline_indexes].to_csv("C:\\Data
Science\\UHasselt\\Thesis\\data_Jannes\\Results\\MiSeq_SOP\\Nodes.csv")
```