Multidimensional scaling informed by F-ratio: Visualizing microbiome for inference (Supplementary Material)

1 Mapping function

In this section, we provide a rationale in deriving a mapping function $f_{\mathbf{z}}(F)$ that is used to minimize an FMDS objective function. The confirmatory term of the objective function is designed to minimize a difference in p-values that are obtained by pseudo F statistics under the original S-dimension and the two-dimension. Each p-value can be obtained based on an empirical distribution of pseudo F from permuted labels under respective dimension (Anderson, 2001).

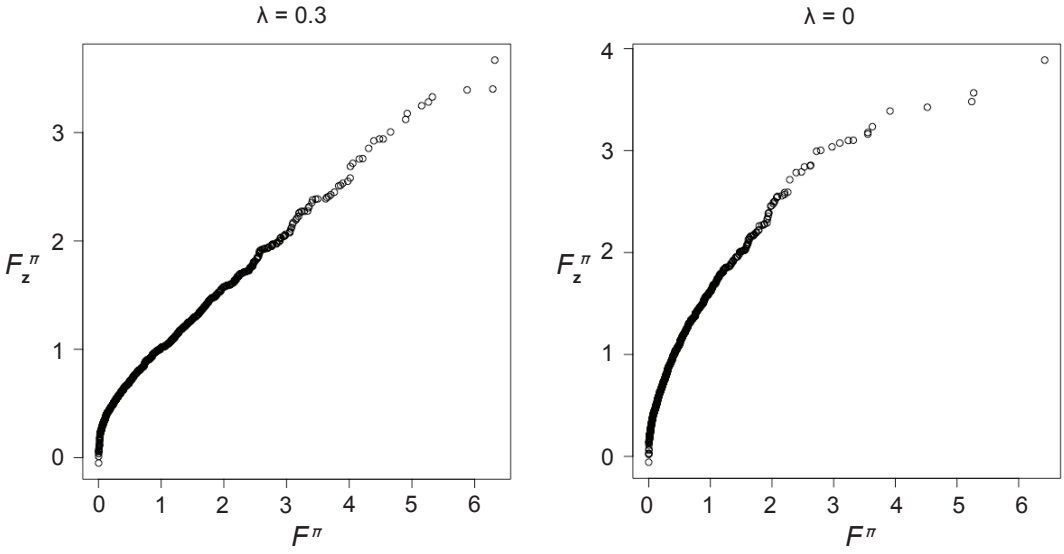
To estimate such pseudo F that satisfies the above, we permute the label set $[y_i]$ $(i = 1, \cdots N)$, denoted with a superscript π (namely y_i^π) and derive the following statistics:

$$F^\pi = \left(\frac{\sum_{i,j} d_{ij}^2}{2 \sum_{i,j} d_{ij}^2 \, \mathbb{1}\{y_i^\pi = y_j^\pi\}} - 1 \right) \cdot (N-2), \tag{S1}$$

$$F_{\mathbf{z}}^\pi = \left(\frac{\sum_{i,j} \|\mathbf{z}_i - \mathbf{z}_j\|_2^2}{2 \sum_{i,j} \|\mathbf{z}_i - \mathbf{z}_j\|_2^2 \, \mathbb{1}\{y_i^\pi = y_j^\pi\}} - 1 \right) \cdot (N-2). \tag{S2}$$

Note that Equation S2 represents a pseudo F that is calculated based on a two-dimensional configuration $\mathbf{z}$, denoted as $F_{\mathbf{z}}$.

Using a pair $(F^\pi, F_{\mathbf{z}}^\pi)$ for every permutation, a mapping function $f_{\mathbf{z}} : F^\pi \to F_{\mathbf{z}}^\pi$ can be derived by performing a local regression. An example is given below, where it is shown that $f_{\mathbf{z}}(\cdot)$ can change by the choice of hyperparameter λ.



Mapping pseudo F's between two dimensionalities. Each data point was obtained by permuting labels over 1,000 iteration, and by setting a hyperparameter for performing the Majorize-Minimization algorithm ($\lambda = 0.3$, left; $\lambda = 0$, right).

Finally, the confirmatory term for our FMDS objective is derived by seeking $\mathbf{z}$ such that

$$\arg\min_{\mathbf{z}} |F_{\mathbf{z}} - f_{\mathbf{z}}(F)| \tag{S3}$$

$$= \arg\min_{\mathbf{z}} \left| (N-2) \cdot \left(\frac{\sum_{i,j} \|\mathbf{z}_i - \mathbf{z}_j\|_2^2}{2\sum_{i,j} \epsilon_{ij} \|\mathbf{z}_i - \mathbf{z}_j\|_2^2} - 1 \right) - f_{\mathbf{z}}(F) \right| \tag{S4}$$

$$= \arg\min_{\mathbf{z}} \left| \frac{\sum_{i,j} \|\mathbf{z}_i - \mathbf{z}_j\|_2^2}{2\sum_{i,j} \epsilon_{ij} \|\mathbf{z}_i - \mathbf{z}_j\|_2^2} - 1 - \frac{f_{\mathbf{z}}(F)}{N-2} \right| \tag{S5}$$

$$= \arg\min_{\mathbf{z}} \left| \frac{\sum_{i,j} \|\mathbf{z}_i - \mathbf{z}_j\|_2^2 - 2\sum_{i,j} \epsilon_{ij} \|\mathbf{z}_i - \mathbf{z}_j\|_2^2 \cdot [1 + f_{\mathbf{z}}(F)/(N-2)]}{2\sum_{i,j} \epsilon_{ij} \|\mathbf{z}_i - \mathbf{z}_j\|_2^2} \right| \tag{S6}$$

$$\approx \arg\min_{\mathbf{z}} \left| \sum_{i,j} \|\mathbf{z}_i - \mathbf{z}_j\|_2^2 - 2\sum_{i,j} \epsilon_{ij} \|\mathbf{z}_i - \mathbf{z}_j\|_2^2 \cdot \left(1 + \frac{f_{\mathbf{z}}(F)}{N-2} \right) \right| \tag{S7}$$

$$= \arg\min_{\mathbf{z}} \left| \sum_{i,j} \left[1 - 2\epsilon_{ij} \left(1 + \frac{f_{\mathbf{z}}(F)}{N-2} \right) \right] \|\mathbf{z}_i - \mathbf{z}_j\|_2^2 \right|. \tag{S8}$$

2 Majorize-minimization algorithm

We provide an analytical expression to derive an iteration and update rule using Majorize-Minimization (MM) algorithm. Here, a configuration $\mathbf{z}^* = (\mathbf{z}_1^*, \cdots \mathbf{z}_N^*) \in \mathbb{R}^{N \times 2}$ is sought to minimize an objective term for FMDS, $O(\mathbf{z})$. We enable this by applying the MM algorithm for every index $k = 1, \cdots N$ minimizing $O(\mathbf{z})$ while other configuration points except for $\mathbf{z}_k$ are fixed. In other words,

$$\mathbf{z}_k^* = \arg\min_{\mathbf{z}_k} O(\mathbf{z}|\mathbf{z}_k) \tag{S9}$$

$$= \arg\min_{\mathbf{z}_k} \sum_{i,j} (d_{ij} - \|\mathbf{z}_i - \mathbf{z}_j\|_2)^2 + \lambda \left| \sum_{i,j} \left[1 - 2\epsilon_{ij} \left(1 + \frac{f_{\mathbf{z}}(F)}{N-2} \right) \right] \|\mathbf{z}_i - \mathbf{z}_j\|_2^2 \right| \tag{S10}$$

$$= \arg\min_{\mathbf{z}_k} \sum_{j=1}^{N} (d_{jk} - \|\mathbf{z}_j - \mathbf{z}_k\|_2)^2 + \lambda\delta(\mathbf{z}) \sum_{j=1}^{N} \left[1 - 2\epsilon_{jk} \left(1 + \frac{f_{\mathbf{z}}(F)}{N-2} \right) \right] \cdot \|\mathbf{z}_j - \mathbf{z}_k\|_2^2 \tag{S11}$$

$$= \arg\min_{\mathbf{z}_k} \sum_{j=1}^{N} \left[1 + \lambda\delta(\mathbf{z}) \left(1 - 2\epsilon_{jk} \left(1 + \frac{f_{\mathbf{z}}(F)}{N-2} \right) \right) \right] \|\mathbf{z}_k - \mathbf{z}_j\|_2^2 - 2d_{jk} \|\mathbf{z}_k - \mathbf{z}_j\|_2 \tag{S12}$$

where we have defined ϵ_{ij} and $\delta(\mathbf{z})$ as

$$\epsilon_{ij} = \mathbb{1}\{y_i = y_j\}, \quad \delta(\mathbf{z}) = \text{sign} \left\{ \sum_{i,j} \left[1 - 2\epsilon_{ij} \left(1 + \frac{f_{\mathbf{z}}(F)}{N-2} \right) \right] \|\mathbf{z}_i - \mathbf{z}_j\|_2^2 \right\} \tag{S13}$$

for simplicity. As described by Borg and Groenen (1997b), applying MM algorithm starts with majorizing with Equation S12, which is written as

$$\sum_{j=1}^{N} \left[1 + \lambda\delta(\mathbf{z}) \left(1 - 2\epsilon_{jk} \left(1 + \frac{f_{\mathbf{z}}(F)}{N-2} \right) \right) \right] \|\mathbf{z}_k - \mathbf{z}_j\|_2^2 - 2d_{jk} \frac{\sum_{s=1}^{2} (z_{ks} - z_{js})(\tilde{z}_{ks} - z_{js})}{\|\tilde{\mathbf{z}}_k - \mathbf{z}_j\|_2}, \tag{S14}$$

where $\tilde{\mathbf{z}}_k$ is a fixed term (not updated) while $\mathbf{z}_k$ still remains as a variable.

Next, we assume that a change of mapping function $f_{\mathbf{z}}(F)$ is negligible and that $\delta(\mathbf{z})$ remains constant during the iteration (e.g., a small change in $\mathbf{z}_k$ by a step). These allow us to approximate Equation S14 with a quadratic expression in terms of $\mathbf{z}$ which can be readily minimized. To find its minimum at $\mathbf{z}_k = \mathbf{z}_k^\dagger$, a derivative is taken

with respect to z_{ks} and is set to zero. In other words, we obtain

$$0 = \sum_{j=1}^{N} \left[1 + \lambda\delta(\mathbf{z}) \left(1 - 2\epsilon_{jk} \left(1 + \frac{f_{\mathbf{z}}(F)}{N-2}\right)\right)\right] (z^{\dagger}_{ks} - z_{js}) - d_{jk} \frac{\tilde{z}_{ks} - z_{js}}{\|\tilde{\mathbf{z}}_k - \mathbf{z}_j\|_2}. \tag{S15}$$

Noting that for a balanced design where $\sum_{j=1}^{N} \epsilon_{jk} = N/2$, for $k = 1, \cdots N$, we rewrite the above with

$$\begin{aligned}
&\sum_{j=1}^{N} \left[1 + \lambda\delta(\mathbf{z}) \left(1 - 2\epsilon_{jk} \left(1 + \frac{f_{\mathbf{z}}(F)}{N-2}\right)\right)\right] z^{\dagger}_{ks} \\
&\quad = \left[N + \lambda\delta(\mathbf{z})N - \lambda\delta(\mathbf{z})N \left(1 + \frac{f_{\mathbf{z}}(F)}{N-2}\right)\right] z^{\dagger}_{ks} \\
&\quad = \left(N - \frac{N\lambda\delta(\mathbf{z}) f_{\mathbf{z}}(F)}{N-2}\right) z^{\dagger}_{ks} \\
&\quad = \sum_{j=1}^{N} \left[1 + \lambda\delta(\mathbf{z}) \left(1 - 2\epsilon_{jk} \left(1 + \frac{f_{\mathbf{z}}(F)}{N-2}\right)\right)\right] z_{js} + d_{jk} \frac{\tilde{z}_{ks} - z_{js}}{\|\tilde{\mathbf{z}}_k - \mathbf{z}_j\|_2}.
\end{aligned} \tag{S16}$$

$$\therefore z^{\dagger}_{ks} = \frac{(N-2)}{N(N-2) - N\lambda\delta(\mathbf{z}) f_{\mathbf{z}}(F)} \cdot \left\{ \sum_{j=1}^{N} \left[1 + \lambda\delta(\mathbf{z}) \left(1 - 2\epsilon_{jk} \left(1 + \frac{f_{\mathbf{z}}(F)}{N-2}\right)\right)\right] z_{js} + d_{jk} \frac{\tilde{z}_{ks} - z_{js}}{\|\tilde{\mathbf{z}}_k - \mathbf{z}_j\|_2} \right\}. \tag{S17}$$

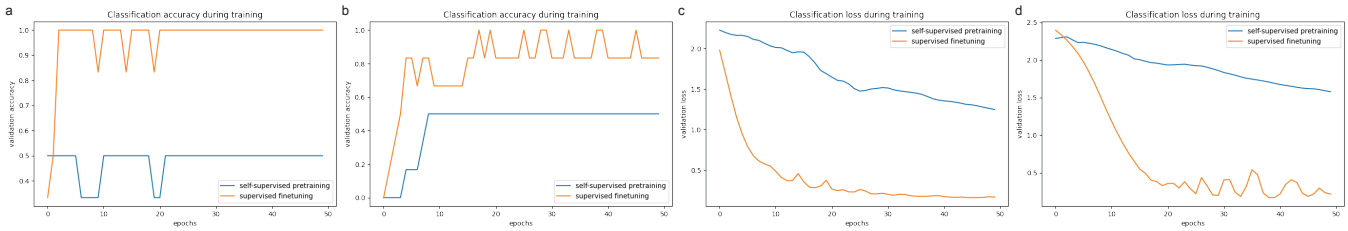
Rewriting Equation S17 in a vector form gives us an update rule of $\mathbf{z}$ in Algorithm 2 as following:

$$\mathbf{z}_k \leftarrow \frac{(N-2)}{N(N-2) - N\lambda\delta(\mathbf{z}) f_{\mathbf{z}}(F)} \cdot \left\{ \sum_{j=1}^{N} \left[1 + \lambda\delta(\mathbf{z}) \left(1 - 2\epsilon_{jk} \left(1 + \frac{f_{\mathbf{z}}(F)}{N-2}\right)\right)\right] \mathbf{z}_j + d_{jk} \frac{\mathbf{z}_k - \mathbf{z}_j}{\|\mathbf{z}_k - \mathbf{z}_j\|_2} \right\}. \tag{S18}$$

3 Neural network architecture

In this section, we describe a procedure for analyzing microbiome data using neural network architecture. To construct a classifier of bacterial community with a convolutional neural network, we first convert the microbial compositional data into a two-dimensional image matrix by implementing PopPhy-CNN (Reiman et al., 2020). In detail, each matrix reflects a phylogenetic tree structure by bacterial 16S rRNA amplicon (amplicon sequence variant or ASV) and its relative abundance which is normalized by cumulative sum scaling (CSS) (Paulson et al., 2013). Seventy-two microbial samples across all sites were converted to 2D arrays with a size of 10 x 42. The data was randomly split into training and validation sets (12 and 60 each) using the stratified K-Fold.

Next, we employ a self-supervised learning method, SimCLR (Chen et al., 2020) as a benchmark. To preserve the phylogenetic information by its row or column index, crop or flip filters were excluded from the data augmentation procedure. Instead, random brightness and contrast / jitter filter were applied with the following parameters: (0.6, 0.2) for pretraining, (0.3, 0.1) for finetuning. The encoder for our SimCLR consists of one Gaussian noise filter, two 2D convolution layers (with kernel size of 5 by 3), and one fully connected layer with 32 output nodes. The projection head consists of two dense layers (32 output nodes each), and the linear probe consists of one dense layer (10 output nodes). Pretraining was performed for 50 epochs, resulting its validation accuracy of 50%. Finetuning followed for another 50 epochs, resulting its validation accuracy of 83.3% (see Figure). Finally, a trained encoder was used to represent a feature (vector of 32) for each microbiome sample. For each pair of features, L2-squared distance was calculated to obtain Stress-1 and Shepard plot.



Performance of SimCLR classifier by 50 training epochs. Validation accuracy from (a) Site 1 and (b) Site 2 community data. Validation loss by categorical cross-entropy from (c) Site 1 and (d) Site 2 data.

4 Alternative evaluation metrics

In this section, we provide an alternative evaluation of various dimension reduction methods by utilizing the metric proposed by Rhodes et al. (2021). The metric has been introduced to evaluate the performance of RF-PHATE and the procedure is as follows. First, find the classification accuracy using the original instances as predictors in a k-NN classifier model. Next, find the regression MSE from a k-NN regression model, where the original distances serve as predictors and the embeddings as the response. Lastly, compute the correlation coefficient between these two importance scores. The assertion is that a higher correlation indicates better preservation of variable structure. Table S1 shows the numerical assessment of methods for simulated and microbial community datasets.

Table S1: Performance evaluation of dimensionality reduction methods using simulated and community site data with alternative evaluation criterion proposed by Rhodes et al. (2021).

	SIMULATION	**SITE 1**	**SITE 2**
FMDS	0.370	0.041	0.231
MDS	0.334	-0.016	0.150
SMDS	0.502	0.592	0.485
UMAP-S	0.728	0.784	0.722
UMAP-U	0.412	0.135	0.183
t-SNE	0.319	0.122	0.173
Isomap	0.315	0.009	0.140

We observe the alternative metric does not directly quantify the effectiveness of embeddings within the context of inference. In specific, examining the correlation between two importance scores (data labels and data-embedding) constitutes a partial evaluation and does not fully explain the embedding-labels relationship. To overcome the limitation, a new metric has been designed for our study as described in the main text. The metric calculates both the correlation between F-ratios and the ratio of p-values and examines data-labels and embedding-labels relationships.

5 Computational complexity

In this section, we provide an upper bound of computational complexity for running algorithms to perform FMDS. It is evaluated on the basis of a single iteration, as the theoretical cost of the entire majorization approach is yet to be characterized (see a recent work by Streeter (2023)). We discuss the time complexity of each step in MAPPING and MMFMDS functions as outlined in the main text (Algorithms 1 and 2).

For MAPPING function, an F-ratio is computed from a set of permuted labels y^{π} and each of input matrices d, $\mathbf{z}$. Each computation takes $\mathcal{O}(N^2)$ operations where N is the sample size. The step repeats for a number of iteration $p = 999$, resulting in $\mathcal{O}(2pN^2)$ operations. Additional steps are taken to sort the lists of permuted F-ratios with $\mathcal{O}(2N \log N)$. In total, the complexity is $\mathcal{O}(2pN^2 + 2N \log N)$.

For MMFMDS function, an F-ratio is computed once from d, y then mapped to $f_{\mathbf{z}}(F)$, which takes $\mathcal{O}(N^2 + \log N)$ operations. Next, a sign of FMDS confirmatory term $\delta(\mathbf{z})$ is obtained and the step takes $\mathcal{O}(N^2)$ operations.

Finally, a configuration $\mathbf{z}$ is updated for every point, taking $\mathcal{O}(N^2)$ operations. Therefore, the complexity for one iteration of the majorization algorithm is $\mathcal{O}(3N^2 + \log N)$.

In summary, the computational cost of performing FMDS (unit iteration) is $\mathcal{O}(2pN^2+3N^2+2N\log N+\log N) \approx \mathcal{O}(pN^2)$. It is compared with other dimension reduction methods as below.

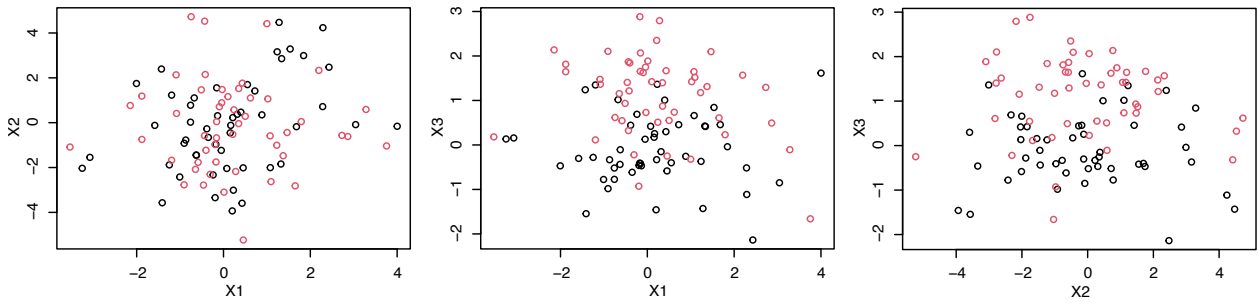
Table S2: Comparison of time complexity between different dimensionality reduction methods.

	Complexity	Algorithm
FMDS[1]	$\mathcal{O}(pN^2)$	Majorization (Borg and Groenen, 1997a)
MDS	$\mathcal{O}(N^3)$	Torgerson (1952)
SMDS[1]	$\mathcal{O}(N^2)$	Witten and Tibshirani (2011)
UMAP	$\mathcal{O}(N\log N)$	McInnes et al. (2018)
t-SNE[1]	$\mathcal{O}(N^2)$	van der Maaten and Hinton (2008)
Isomap	$\mathcal{O}(N^2\log N)$	Tenenbaum et al. (2000)

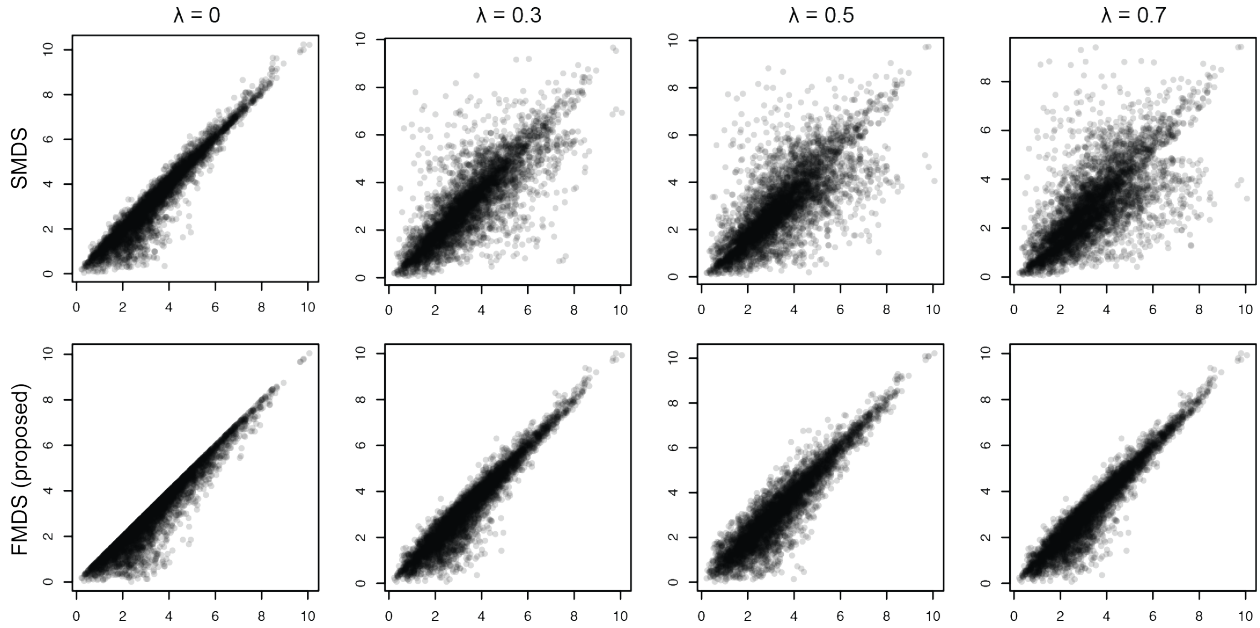
All experiments are run on a single Macbook Pro 2016 laptop with a 3.1 GHz Dual-core Intel core i5 CPU.

[1]Corresponds to a single iteration and does not represent a total complexity.

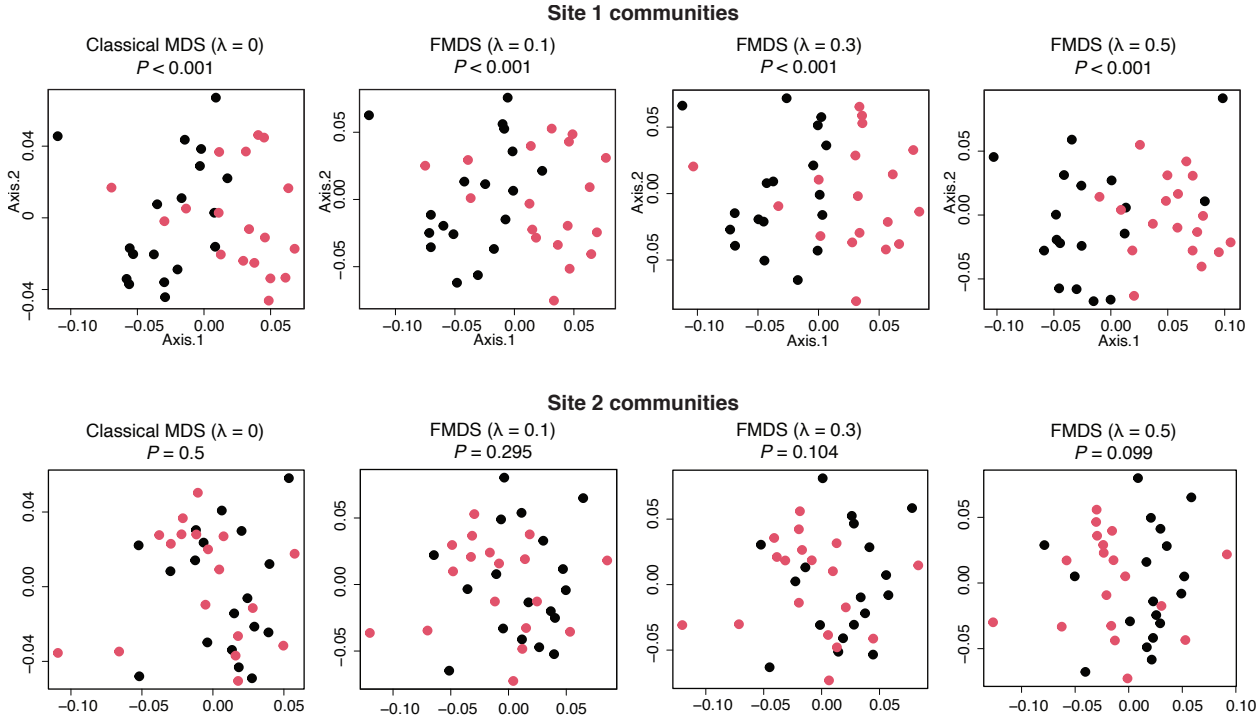
6 Supplementary figures



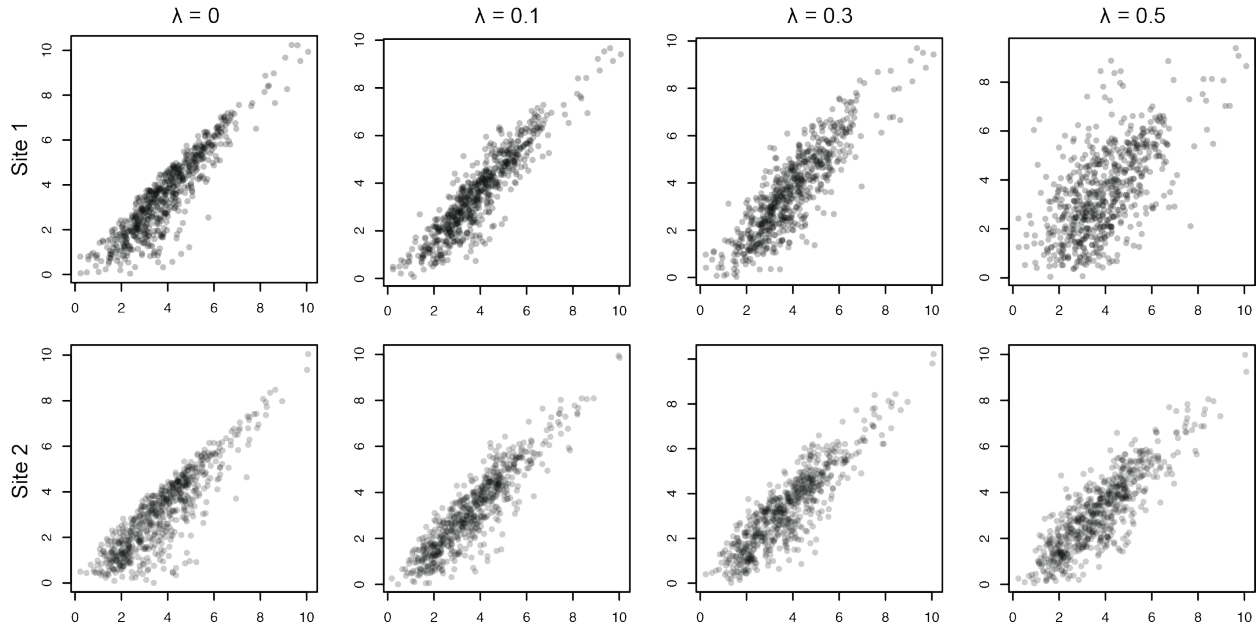
Supplementary Figure 1: Two-dimensional plot of 100 simulated data points.



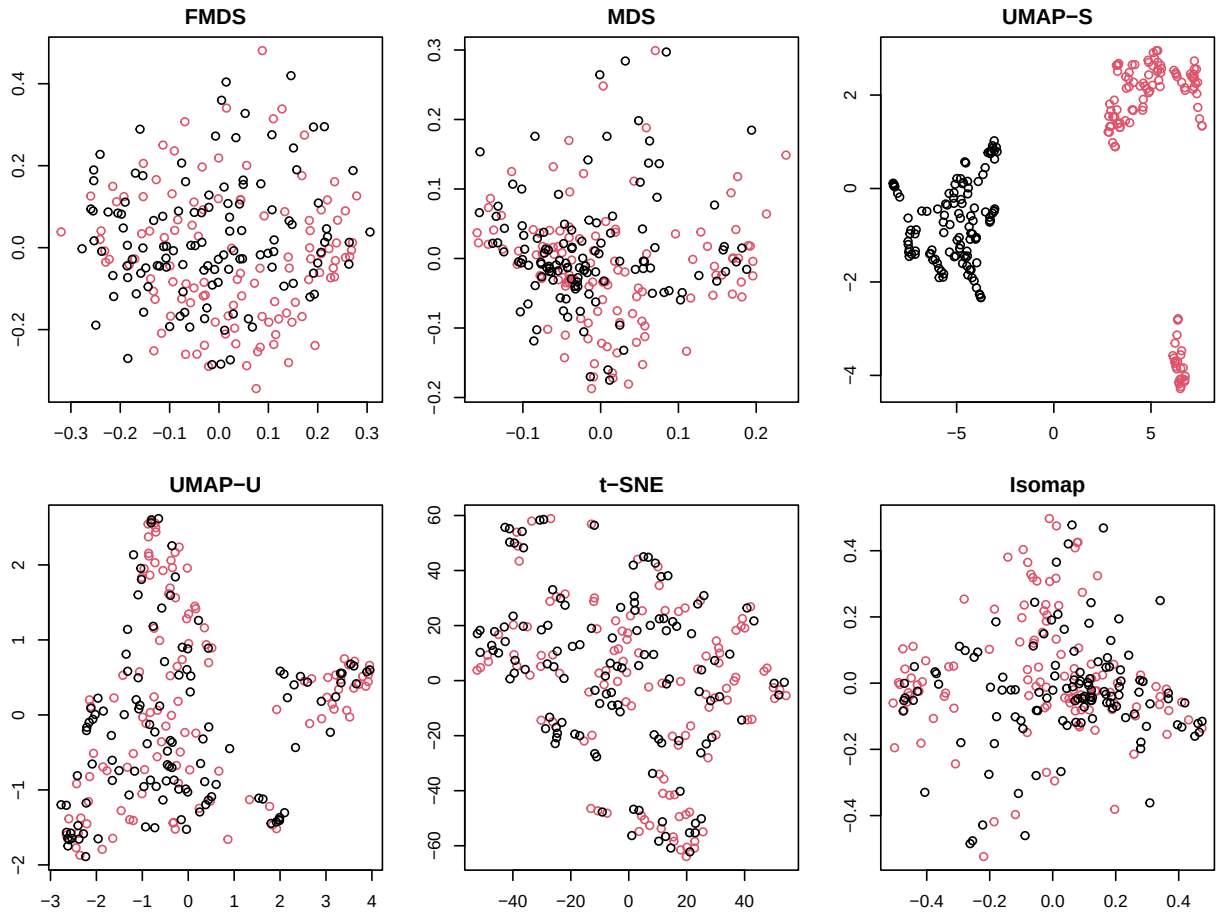
Supplementary Figure 2: Shepard plot of simulation data using SuperMDS (Witten and Tibshirani, 2011) (first row) and F-informed MDS (second row). A comparison is made based on a ratio between confirmatory and MDS term. Note that the FMDS consistently shows a linear relationship irrespective of the λ values, while the SMDS configurations show large dispersion even with the small λ values.



Supplementary Figure 3: Visualization of bacterial community using proposed FMDS. Microbial community samples are collected from Site 1 (top row) and Site 2 (bottom row).



Supplementary Figure 4: Shepard plot in microbial community data collected from site 1 (first row) and site 2 (second row) for a range of hyperparameters.



Supplementary Figure 5: Visualization of cirrhosis patients and healthy human gut microbiome. The following hyperparameters were used: $\lambda = 0.3$ (FMDS), neighbors $= 10$ (UMAP-S), neighbors $= 30$ (UMAP-U), perplexity$= 7$ (t-SNE), $K = 10$ (Isomap).

References

M. J. Anderson. A new method for non-parametric multivariate analysis of variance. *Austral Ecology*, 26(1): 32–46, 2001.

I. Borg and P. Groenen. *A Majorization Algorithm for Solving MDS*, pages 169–197. Springer New York, New York, NY, 1997a. ISBN 978-0-387-28981-6.

I. Borg and P. Groenen. *Confirmatory MDS*, pages 181–197. Springer New York, New York, NY, 1997b. ISBN 978-1-4757-2711-1.

T. Chen, S. Kornblith, M. Norouzi, and G. Hinton. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119, pages 1597–1607. Proceedings of Machine Learning Research, 2020.

L. McInnes, J. Healy, and J. Melville. UMAP: Uniform manifold approximation and projection for dimension reduction, February 2018.

J. N. Paulson, O. C. Stine, H. C. Bravo, and M. Pop. Differential abundance analysis for microbial marker-gene surveys. *Nature Methods*, 10, 2013. doi: 10.1038/nmeth.2658.

D. Reiman, A. A. Metwally, J. Sun, and Y. Dai. PopPhy-CNN: A phylogenetic tree embedded architecture for convolutional neural networks to predict host phenotype from metagenomic data. *IEEE Journal of Biomedical and Health Informatics*, 24(10):2993–3001, 2020.

J. S. Rhodes, A. Cutler, G. Wolf, and K. R. Moon. Random forest-based diffusion information geometry for supervised visualization and data exploration. In *2021 IEEE Statistical Signal Processing Workshop (SSP)*, pages 331–335, 2021.

M. Streeter. Universal majorization-minimization algorithms, July 01, 2023 2023. URL `https://ui.adsabs.harvard.edu/abs/2023arXiv230800190S`. 29 pages, 12 figures.

J. B. Tenenbaum, V. de Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–23, 2000.

W. S. Torgerson. Multidimensional scaling: I. theory and method. *Psychometrika*, 17(4):401–419, 1952. ISSN 1860-0980. doi: 10.1007/BF02288916.

L. van der Maaten and G. Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9(86): 2579–2605, 2008.

D. M. Witten and R. Tibshirani. Supervised multidimensional scaling for visualization, classification, and bipartite ranking. *Computational Statistics & Data Analysis*, 55(1):789–801, 2011.