Supporting Information for "Zero-inflated Poisson Factor Model with Application to Microbiome Read Counts" by

Tianchen Xu*

Mailman School of Public Health, Columbia University, NY 10032, USA

Ryan T. Demmer

School of Public Health, University of Minnesota, MN 55454, USA

Gen Li

Mailman School of Public Health, Columbia University, NY 10032, USA

1 Web Appendix A

In order to obtain the initial U and V, we apply the singular value decomposition (SVD) to the log-transformed matrix $\widetilde{A}$ and obtain the components $\{U', V', S'\}$ (i.e., $\ln(\widetilde{A}) = U'S'V'^{\top}$). Set $V^{old} = V'$ and $U^{old} = (s'_{11}u'_{(,1)}, s'_{22}u'_{(,2)}, \ldots, s'_{KK}u'_{(,K)})$, where s'_{kk} is the k-th diagonal element of S'. Assuming matrix U^{old} is known, we estimate the loadings $v_{(j,)}$ and shape parameter τ by fitting a ZIP regression (will be discussed in Section 2.3) with $a_{(,j)}$ as the response, vector $u^{old}_{(,1)}$, $u^{old}_{(,2)}$, $\cdots$, $u^{old}_{(,K)}$ as the covariates and a scaling vector N as an extra offset parameter. Since A has m columns, we need to fit m GLMs to obtain m rows in V. However, an important assumption of our model is that the link between p_{ij} and λ_{ij} (i.e., τ) remains the same across all m different GLMs. To accommodate this, we solve all m models simultaneously to get a globally best τ value. Here we combine the response, covariates and offset parameter in m regressions into larger scale matrices $A^{(u)}$, $U^{\star\, old}$ and $N^{(u)}$ and fit a ZIP regression with $A^{(u)}$ as response variable, columns in $U^{\star\, old}$ as covariates and $N^{(u)}$ as the offset parameter:

$$A^{(u)} = \begin{pmatrix} a_{(,1)} \\ a_{(,2)} \\ \vdots \\ a_{(,m)} \end{pmatrix}, \qquad U^{\star\, old} = \underbrace{\begin{pmatrix} U^{old} & & & \\ & U^{old} & & \\ & & \ddots & \\ & & & U^{old} \end{pmatrix}}_{m \text{ times}}, \qquad N^{(u)} = \left.\begin{pmatrix} N \\ N \\ \vdots \\ N \end{pmatrix}\right\} m \text{ times} .$$

The fitted coefficient V^s is a combined vector of $v_{(i,)}$ such that $V^s = (v_{(1,)}, v_{(2,)}, \cdots, v_{(m,)})$. We could obtain the fitted V^{new} by cutting V^s to $v_{(i,)}$'s and rearrange $v_{(i,)}$'s to matrix V^{new}.

Then update U^{new} in a similar fashion. The response and covariates are $A^{(v)}$ and every columns in $V^{\star new}$ and an offset parameter $N^{(v)}$ is needed as well:

$$A^{(v)} = \begin{pmatrix} a_{(1,)}^{\top} \\ a_{(2,)}^{\top} \\ \vdots \\ a_{(n,)}^{\top} \end{pmatrix}, \quad V^{\star new} = \underbrace{\begin{pmatrix} V^{new} & & & \\ & V^{new} & & \\ & & \ddots & \\ & & & V^{new} \end{pmatrix}}_{n \text{ times}}, \quad N^{(v)} = \Big(\underbrace{N_1, \cdots N_1}_{m \text{ times}}, \cdots, \underbrace{N_n, \cdots, N_n}_{m \text{ times}}\Big)^{\top}.$$

The fitted coefficient is $U^s = (u_{(1,)}, u_{(2,)}, \cdots, u_{(n,)})$ and thus the U^{new} is able to be reconstructed from U^s in a similar way like V^{new}.

After U, V are updated, one more step is involved to ensure the uniqueness and orthogonality of these updated components. We apply SVD to the $U^{new}V^{new\,T}$ and label the components by $\{U', V', S'\}$. Set $U^{old} = (s'_{11}u'_{(,1)}, s'_{22}u'_{(,2)}, \ldots, s'_{KK}u'_{(,K)})$ and $V^{old} = V'$.

We use the updated U, V and τ to obtain the estimates of Λ and P in the current round of iteration and then calculate the likelihood value $L(A)$:

$$L(A) = \prod_{i,j} L(a_{ij}; U, V, \tau, N) = \prod_{i,j} \left\{ p_{ij}\mathbb{I}(a_{ij} = 0) + (1 - p_{ij}) \frac{(N_i\lambda_{ij})^{a_{ij}} e^{-N_i\lambda_{ij}}}{a_{ij}!} \right\}$$

where $\ln(\lambda_{ij}) = \sum_{k=1}^{K} u_{ik}v_{jk}$ and $\mathrm{logit}(p_{ij}) = -\tau \ln(\lambda_{ij})$.

When the percentage of total likelihood difference between two iterations is less than a certain small value, the algorithm terminates; Otherwise, we continue to update U, V, τ until convergence. In ZIP regression step where U, V and τ are updated, we will use the EM algorithm to estimate the coefficients (see Section 2.3), and thus the likelihood increases due to the nature of EM algorithm used in regression estimation. The likelihood remains the same in SVD step. Overall, the algorithm is guaranteed to converge.

2 Web Appendix B

LM algorithm introduces a positive damping parameter μ. If we reduce μ, the LM algorithm behaves like Newton's method, which is a good way to get quadratic convergence in the final stages of the iteration; while if we enlarge μ, the descent direction in LM algorithm is closer to the gradient descent method, which is free from the information of second derivative of the objective function and thus the algorithm still works when the Hessian matrix of the objective function is ill-conditioned or nearly singular. Such an algorithm could be put into the trust-region framework and is implemented in many solvers with common programming languages [Yuan, 1999, Moré and Sorensen, 1983].

B.1 LM Algorithm

We use Levenberg-Marquardt method to solve β and τ. The objective function is $Q(\beta, \tau) = -\ln(L)$ and we want to minimize the objective function. The log likelihood function is:

$$\begin{aligned}
\ln(L) &= \sum_{i=1}^{n} z_i \ln(p_i) + \sum_{i=1}^{n} (1 - z_i)\big\{y_i \ln(m_i\lambda_i) - m_i\lambda_i - \ln(y_i!) + \ln(1 - p_i)\big\} \\
&= -\tau Z X\beta - \sum_{i=1}^{n} \ln(1 + e^{-\tau X_i^\top \beta}) \\
&\qquad + (1 - Z)\{\mathrm{diag}(y_i)(X\beta + \ln(m)) - \mathrm{diag}(m_i)e^{X\beta} - \ln(Y!)\}
\end{aligned}$$

where $\mathrm{diag}(\alpha_\mathrm{i})$ represents a matrix whose diagonal elements are α_i; ln is a pointwise operator on the vector.

The first and second derivative of $\ln(L)$ are:

$$\begin{aligned}
J_\beta &= \frac{\partial \ln(L)}{\partial \beta^\top} = (-\tau Z + \tau W + (1 - Z)U)\, X \\
J_\tau &= \frac{\partial \ln(L)}{\partial \tau} = (W - Z)X\beta \\
H_{\beta\beta} &= \frac{\partial^2 \ln(L)}{\partial \beta \partial \beta^\top} = X^\top R X \\
H_{\tau\tau} &= \frac{\partial^2 \ln(L)}{\partial \tau^2} = -\sum_{i=1}^{n} \frac{(X_i^\top \beta)^2 e^{\tau X_i^\top \beta}}{(e^{\tau X_i^\top \beta} + 1)^2} \\
H_{\tau\beta} &= \frac{\partial^2 \ln(L)}{\partial \tau \partial \beta^\top} = (V - Z)X
\end{aligned}$$

where $W = \big((e^{\tau X_1 \beta} + 1)^{-1}, \ldots, (e^{\tau X_n \beta} + 1)^{-1}\big)$,

$$U = \text{diag}\Big(y_i - m_i e^{X_i^\top \beta}\Big),$$
$$R = \text{diag}\Big(-\tfrac{\tau^2 e^{\tau X_i^\top \beta}}{(e^{\tau X_i^\top \beta}+1)^2} - (1-z_i)(m_i e^{X_i^\top \beta})\Big),$$
$$V = \Big(\tfrac{e^{\tau X_1 \beta} - \tau X_1 \beta e^{\tau X_1 \beta}+1}{(e^{\tau X_1 \beta}+1)^2}, \ldots, \tfrac{e^{\tau X_n \beta} - \tau X_n \beta e^{\tau X_n \beta}+1}{(e^{\tau X_n \beta}+1)^2}\Big).$$

In addition, diag(α_i) represents a matrix whose diagonal elements are α_i; ln is a pointwise operator on the vector.

Correspondingly, the Jacobian (J) and Hessian (H) matrix of objective function $Q(\beta, \tau)$are:

$$J(\beta,\tau) = -\begin{pmatrix} J_\beta^\top \\ J_\tau \end{pmatrix} \qquad H(\beta,\tau) = -\begin{pmatrix} H_{\beta\beta} & H_{\tau\beta}^\top \\ H_{\beta\tau} & H_{\tau\tau} \end{pmatrix}$$

Initialize:

Step 1: Set initial β_0 and τ_0, which is discussed in Section 2.3. Then we calculate J_0 with these initial values.

Step 2: Set initial damping parameter μ:

$$\mu = \rho \cdot J_m$$

where ρ is user specified and the default value is 1×10^{-5}; J_m is the maximum element in matrix $J_0^\top J_0$.

Iterating:

Step 3: Multiply μ by 2 until $(H(\beta,\tau) + \mu I)$ is positive definite.

Step 4: Obtain the descent direction h by solving equation:

$$(H(\beta,\tau) + \mu I)h = -J(\beta,\tau).$$

Step 5: Define and compute the gain ratio δ:

$$\delta = -\frac{Q(\beta,\tau) - Q((\beta,\tau)+h)}{h^\top J(\beta,\tau) + \frac{1}{2}h^\top (H(\beta,\tau)+\mu I)h}.$$

Step 6: If $\delta > 0.001$, we obtain better $\ln L(\beta,\tau)$ value, so we update β, τ, μ by:

$$\begin{pmatrix} \beta_{k+1} \\ \tau_{k+1} \end{pmatrix} = \begin{pmatrix} \beta_k \\ \tau_k \end{pmatrix} + h, \qquad \mu = \mu \cdot \max\left\{\frac{1}{3}, 1 - (2\delta - 1)^3\right\}$$

If $\delta \leq 0.001$, do not update β, τ. Update $\mu = 2\mu$.

Step 7: Repeat from step 3 to step 6 until the algorithm converges. The convergence criterion is discussed in Section 2.3.

B.2 Initial Values

Similar to many fitting algorithms, the LM algorithm only finds a local extremum, which requires us to provide a reasonable initial τ_0 and β_0. Fitted β from Poisson regression could be used as the initial value in the first iteration of the EM algorithm. For initial value of τ_0, we have to estimate $\overline{p}$ first. Assuming that all y_i from the Poisson distribution are not zero, estimate $\overline{p}$ under such an assumption:

$$\overline{p} = \frac{\#\ y_i\text{'s that are zeros}}{n}$$

Obviously, this expression will overestimate the $\overline{p}$, because zeros from the Poisson distribution are wrongly regraded and counted as a part of true zeros, especially when the estimated $\overline{p}$ is relatively low. However, such a estimator is usually accurate enough to help the fitting algorithm to converge. Then we are able to estimate the initial τ_0 at beginning of each round by the following expression (the derivation is at end of Web Appendix B):

$$\tau_0 = -\frac{n\,\mathrm{logit}(\overline{p})}{\sum_{i=1}^{n} X_i^\top \beta} \tag{1}$$

In practice, if the model is stable and the algorithm converges successfully with all initial parameters (τ and β) in the previous iteration, we tend to use the fitted the parameters in the previous iteration, for which will accelerate the convergency speed. Only if using the previous parameters as the initial parameter causes divergence, we will then turn to the estimation of β_0 and τ_0 mentioned above.

Derivation of equation (1)*:*

Since we have the relationship between p_i and λ_i:

$$\text{logit}(p_i) = -\tau \ln(\lambda_i)$$

Plug $\ln(\lambda_i) = X_i^\top \beta$ into the expression above:

$$\text{logit}(p_i) = -\tau X_i^\top \beta$$

We use $\text{logit}(\overline{p})$ to present the mean value on the left hand side, and take the average of $X_i\beta$:

$$\text{logit}(\overline{p}) = -\frac{\tau}{n}\sum_{i=1}^{n} X_i^\top \beta$$

Correspondingly, the estimation of τ_0 is:

$$\tau_0 = -\frac{n\,\text{logit}(\overline{p})}{\sum_{i=1}^{n} X_i^\top \beta}$$

3 Web Appendix C

We generate rank-3 synthetic NGS data of 200 samples ($n = 200$) and 100 taxa ($m = 100$) according to the assumption in our paper. The Poisson logarithmic rate matrix $\Lambda = UV^\top$, where $U \in \mathbb{R}^{m\times 3}$ is a left singular vector matrix, and $V \in \mathbb{R}^{n\times 3}$ is a right singular vector matrix. We consider three different clustering patterns in the samples as depicted in U. To generate U, we create a 200-by-3 matrix U such that:

$$\begin{aligned} &U(36:80,1) = 2.0, \qquad && U(81:140,1) = 1.7 \\ &U(1:35,2) = 1.8, && U(36:80,2) = 0.9 \\ &U(36:200,3) = 1.7 \end{aligned}$$

with all the other entries being 0, and then jitter all the entries by adding random numbers generated from $N(0, 0.06^2)$. Similarly, To generate V, we create a 100-by-3 matrix V such that:

$$\begin{aligned} &V(61:100,1) = 1.7 && \\ &V(36:60,2) = 1.7, && V(61:100,2) = 1.0 \\ &V(1:25,3) = 1.7, && V(26:100,3) = 0.9 \end{aligned}$$

with all the other entries being 0, and then jitter all the entries by adding random numbers generated from $N(0, 0.05^2)$. The three columns of U and V are plotted in the columns of Web Figure 1(a) and the true $\ln(\lambda)$ matrix is plotted in Web Figure 1(b). Each row in U corresponds to one sample and each row in V indicates one taxon profile. In Web Figure 1(c) & (d), we applied complete linkage hierarchical clustering to U, V [Eisen et al., 1998]. It is clear that both taxa and samples could be clustered into 4 groups.

4 Web Appendix D

As an example, Web Figure 2 shows a typical case that how the fitted distribution changes in setting (1) as inflated zeros percentages go from 0% to 40%. All methods work well when inflated zeros do not exist. When the true zero percentage goes higher (20%), the estimated distribution from log-SVD shifts to the left to capture the excessive zeros. When true zero percentage continues growing to 40%, our method is the only method that could keep the right clustering of both samples and taxa. log-SVD fails to recover the underlying Poisson rate distribution under such a high true zero percentage. PSVDOS performs better to some extent, but also fails to capture the right taxa clustering just like log-SVD. GOMMS successfully clusters the taxa, but the sample clustering is not reflected because it assigns each taxon a single probability of true zero and this might limit the heterogeneity between samples.

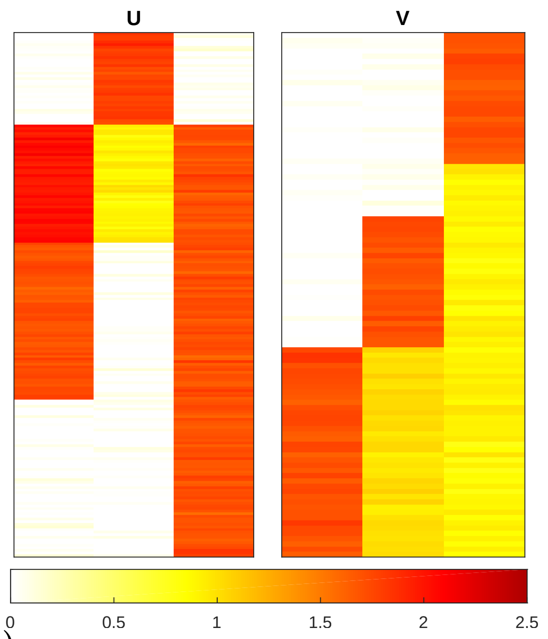
References

Michael B Eisen, Paul T Spellman, Patrick O Brown, and David Botstein. Cluster analysis and display of genome-wide expression patterns. *Proceedings of the National Academy of Sciences*, 95(25):14863–14868, 1998.

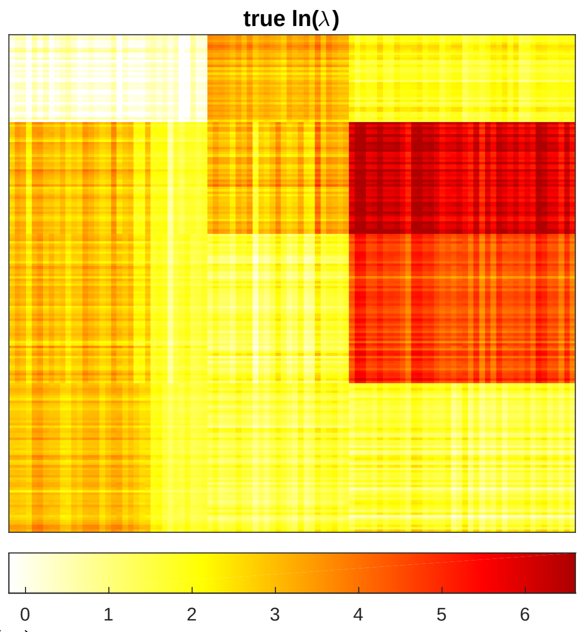
Jorge J Moré and Danny C Sorensen. Computing a trust region step. *SIAM Journal on Scientific and Statistical Computing*, 4(3):553–572, 1983.

Yaxiang Yuan. Nonlinear optimization: trust region algorithms. In *Proceedings of the Third Chinese SIAM Conference*, pages 84–102, 1999.

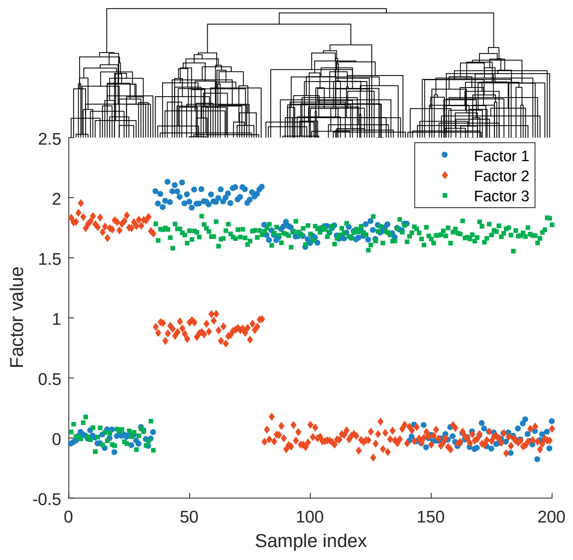
(a)



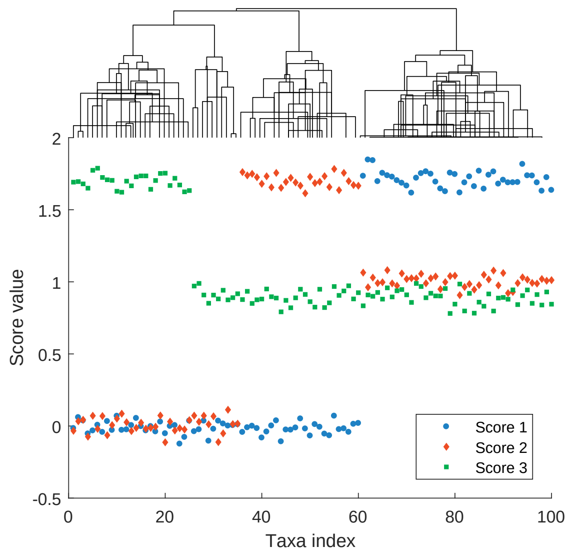
(b)



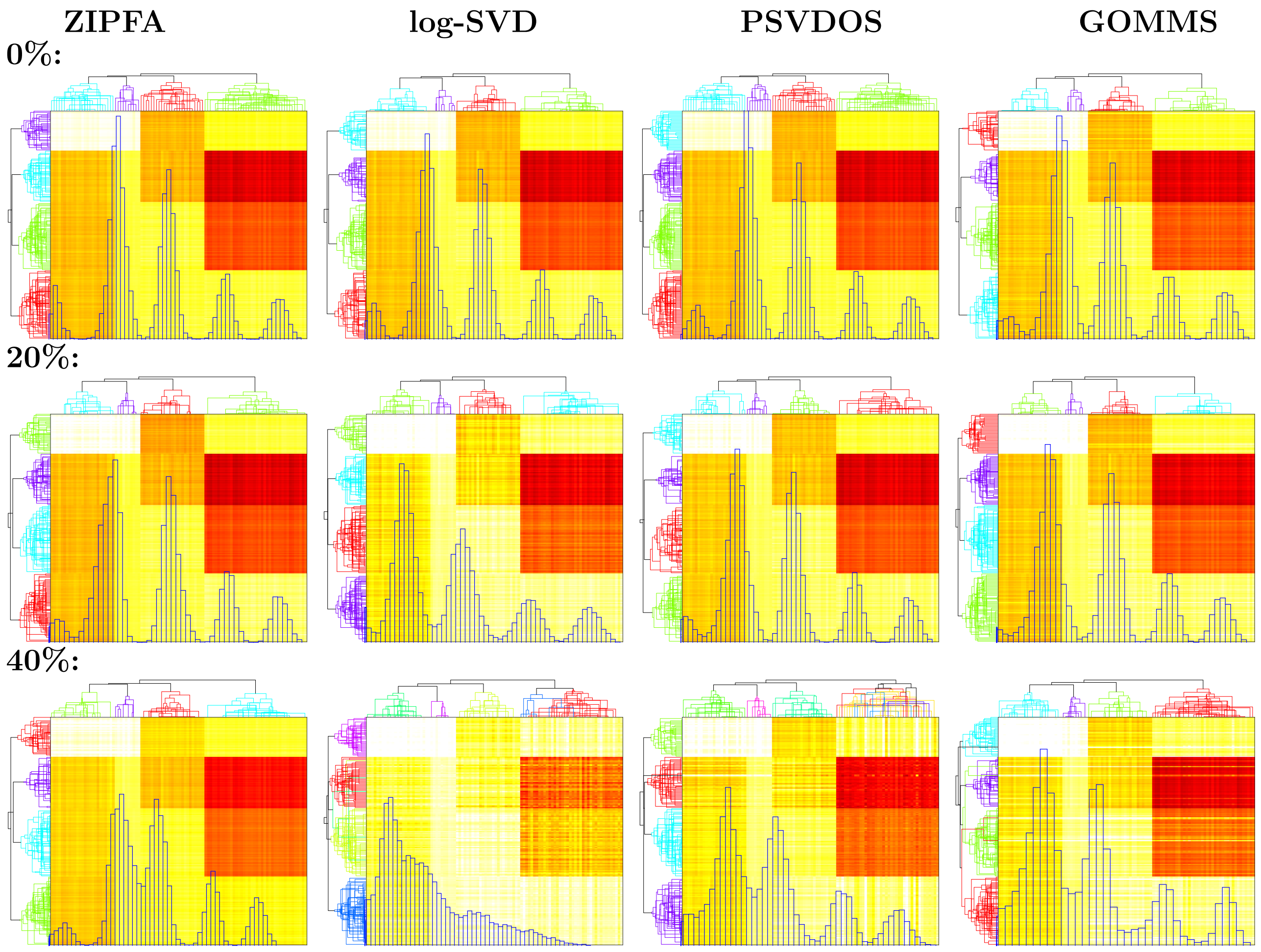
(c)



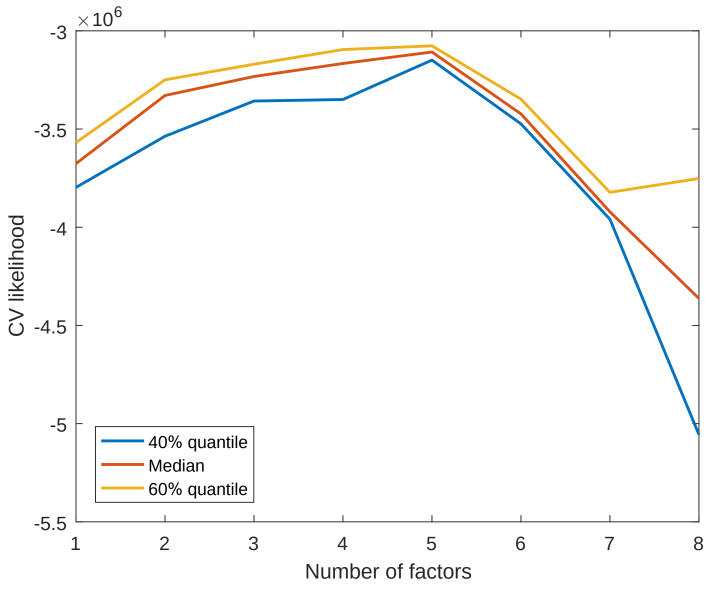
(d)



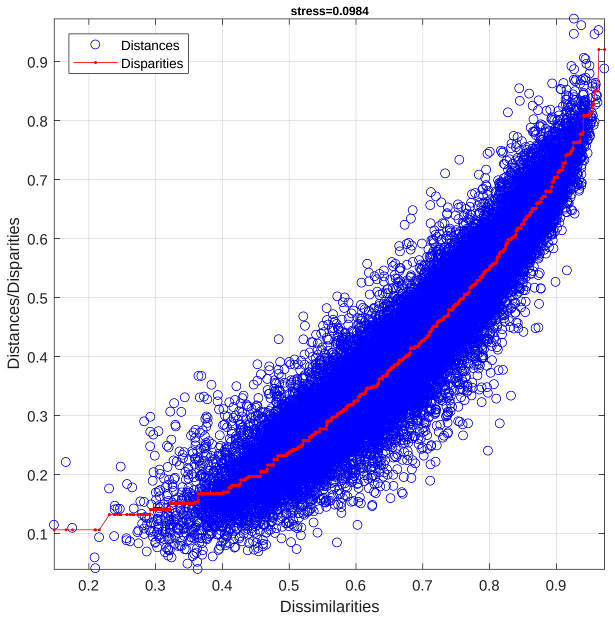
Web Figure. 1: Plots of simulation parameters. (a) True left singular vectors U and true right singular vector V, indicating taxon clusters. (b) Heatmap of true $\ln(\lambda)$ matrix. (c)(d) The factor values for each sample/taxa. They could be clustered into 4 groups.



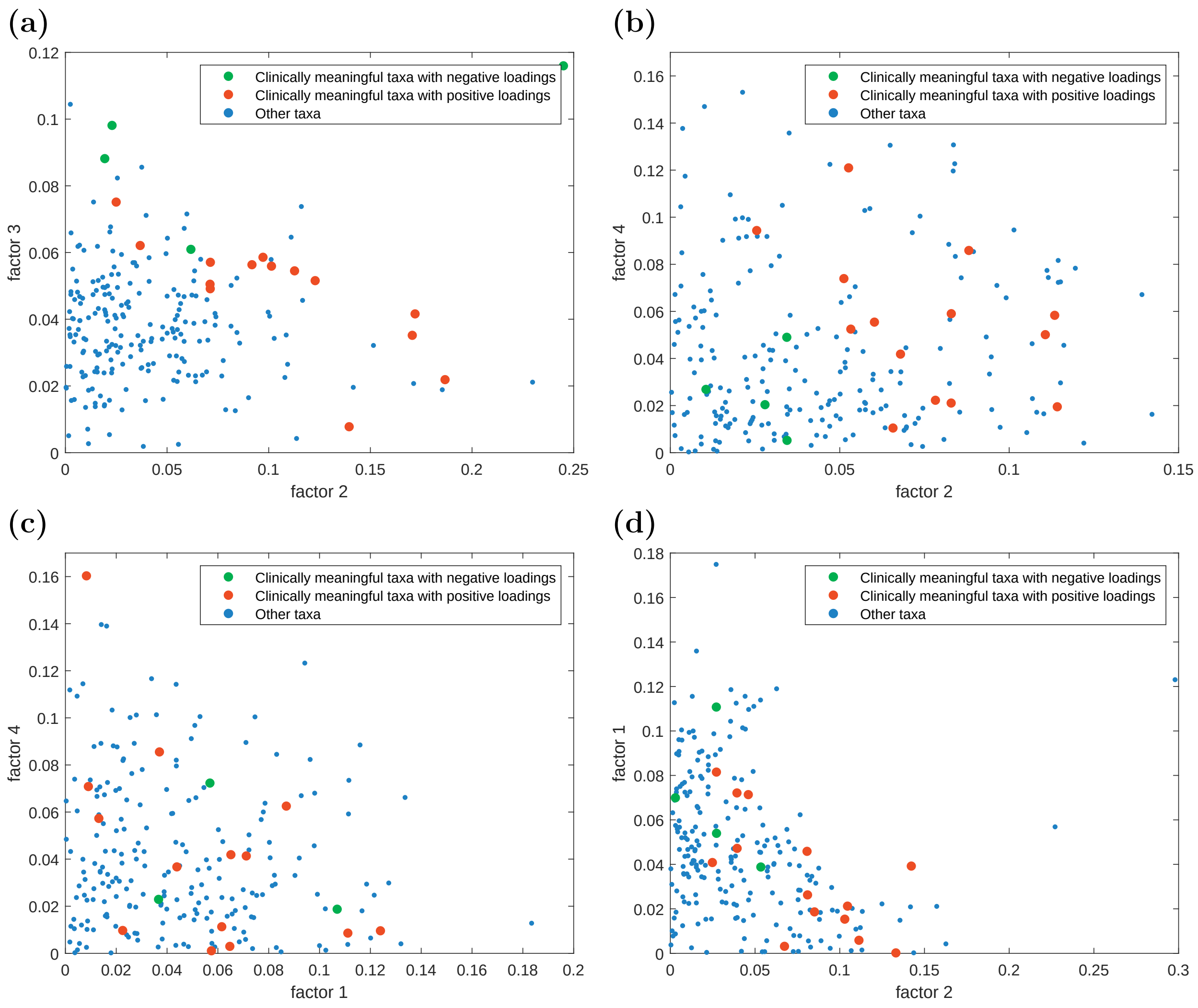
Web Figure. 2: Heatmap of $\widehat{U}\widehat{V}^\top$ from different methods in setting 1. Blue histogram shows the distribution of fitted $\widehat{U}\widehat{V}^\top$; Phylogenetic tree on the top and left shows clustering of taxa and samples.



Web Figure. 3: Cross validation to choose the number of factors in real data analysis. At a specified rank, we obtain several CV likelihoods by assigning test sets randomly. Three solid lines indicate 40%, 50% and 60% quantile of the CV likelihoods.



Web Figure. 4: Shepard stress plot for nMDS model in Subsection 4.2



Web Figure. 5: Absolute taxa loadings on the two most significant factors. Each point is a loading coordinate of a taxon on these two factors. Blue or red dots are clinically meaningful taxa in other literature. (a) Loadings on factor 2, 3 of our proposed ZIPFA. (b) Loadings on factor 2, 4 of log-SVD. (c) Loadings on factor 1, 4 of PSVDOS. (d) Loadings on factor 1, 2 of GOMMS

Web Table. 1: Coefficients and p-value of 5 factors in different models

Response	ZIPFA		log-SVD		PSVDOS		GOMMS		PCoA		nMDS	
	coef	p-value	coef	p-value	coef	p-value	coef	p-value	coef	p-value	coef	p-value
	0.0215	0.038*	0.0039	0.434	0.0043	0.352	0.0009	0.903	-0.4522	0.008**	-0.5448	0.004**
	-0.0039	0.029*	0.0090	0.001**	0.0018	0.673	-0.0034	0.150	0.1073	0.587	-0.3279	0.154
MeanAttLoss ^{a}	0.0138	0.003**	-0.0053	0.190	-0.0020	0.294	0.0020	0.492	-0.4579	0.033*	-0.2522	0.295
	0.0071	0.089$^{\cdot}$	-0.0023	0.569	0.0016	0.524	-0.0024	0.318	0.0702	0.793	-0.1403	0.628
	-0.0012	0.709	0.0081	0.138	0.0028	0.464	0.0000	0.993	-0.2946	0.334	0.4371	0.176
	0.0088	0.111	0.0014	0.586	0.0052	0.032*	0.0097	0.016*	-0.2420	0.008**	-0.3114	0.002**
	-0.0033	0.001***	0.0062	0.000***	-0.0006	0.796	-0.0048	0.000***	0.0878	0.409	-0.1943	0.116
Mean PD ^{b}	0.0056	0.023*	-0.0031	0.140	0.0001	0.938	0.0017	0.264	-0.2412	0.037*	-0.0685	0.596
	0.0018	0.420	-0.0043	0.046*	-0.0004	0.789	-0.0010	0.425	0.2354	0.103	-0.3275	0.036*
	-0.0013	0.453	0.0034	0.242	0.0024	0.241	0.0018	0.390	-0.0132	0.936	0.1198	0.489
	0.0069	0.108	0.0002	0.937	0.0014	0.444	0.0030	0.341	-0.3098	0.000***	-0.3442	0.000***
	-0.0032	0.000***	0.0053	0.000***	0.0018	0.285	-0.0030	0.002**	0.0462	0.575	-0.0327	0.733
Mean BP ^{c}	0.0049	0.010**	0.0011	0.491	0.0000	0.964	0.0007	0.591	-0.0168	0.851	0.0621	0.535
	0.0023	0.191	-0.0025	0.138	0.0025	0.017*	0.0000	0.975	0.1172	0.292	-0.1597	0.186
	-0.0016	0.224	0.0039	0.080$^{\cdot}$	0.0024	0.130	0.0014	0.405	0.0612	0.629	-0.0394	0.769

^{a}Mean attachment loss.
^{b}Mean probing depth.
^{c}Mean bleeding on probing.

Web Table. 2: The loading values and ranks of potentially clinically meaningful taxa in ZIPFA and log-SVD

ZIPFA			log-SVD		
taxa	rank	loading	taxa	rank	loading
Actinomyces HOT 169	1	-0.245	Eubacterium[11][G-3] spp.	10	0.114
Streptococcus_mutans	3	-0.187	Fretibacterium spp. HOT 360	11	0.113
Eubacterium[11][G-5] spp.	5	0.172	Fretibacterium spp.	14	0.111
Treponema spp.	7	0.171	Eubacterium[11][G-5] spp.	29	0.088
Bacteroidetes[G-5] HOT 511	10	0.140	Eubacterium[11][G-6] spp.	37	0.083
Porphyromonas_gingivalis	11	0.123	TM7[G-1] spp.	38	0.083
TM7[G-1] spp.	15	0.113	TM7 spp.	44	0.079
Porphyromonas spp.	21	0.101	Filifactor spp.	58	0.068
Filifactor spp.	25	0.097	Treponema spp.	60	0.066
Eubacterium[11][G-6] spp.	26	0.092	Bacteroidetes[G-5] HOT 511	67	0.060
TM7 spp.	41	0.071	Porphyromonas_gingivalis	82	0.053
Fretibacterium spp.	42	0.071	Porphyromonas spp.	84	0.053
Fretibacterium spp. HOT 360	43	0.071	Prevotella_intermedia	89	0.051
Actinomyces HOT 175	58	-0.062	Actinomyces_naeslundii	122	-0.034
Prevotella_intermedia	108	0.037	Actinomyces HOT 170	123	-0.034
Eubacterium[11][G-3] spp.	134	0.025	Actinomyces HOT 169	140	-0.028
Actinomyces_naeslundii	142	-0.023	Streptococcus_mutans	147	-0.026
Actinomyces HOT 170	161	0.019	Actinomyces HOT 175	195	-0.011