

MDSINE 2.0 Supplement

March 12, 2020

Contents

1	Notation	3
1.1	Index notation	3
2	Model details	3
2.1	Dirichlet process for interactions and perturbations	3
2.2	Baseline logistic growth	4
2.3	Process Variance	4
2.4	Auxiliary Trajectory	5
2.5	Perturbations	5
2.6	Error model	5
3	Model inference	6
3.1	Logistic growth parameters and perturbation magnitudes	6
3.2	Logistic growth prior variances	6
3.3	Process variance parameters	7
3.4	Latent and auxiliary trajectory	7
3.5	Interaction and perturbation indicator	9
3.6	Cluster assignment	9
3.7	Cluster concentration	10
3.8	Interaction and perturbation indicator probability	10
4	Setting hyperparameters and initializations	10
4.1	Cluster assignment and concentration	10
4.2	Interactions and perturbations	10
4.3	Process variance parameters	11
4.4	Logistic growth parameters and prior variances	11
4.5	Latent and auxiliary trajectories	11
4.6	Staged initialization	12
4.6.1	First version	12
4.6.2	Second (current) version	12
5	Benchmarking	12
5.1	Data simulation	13
5.2	Simulating noise	13
5.2.1	Reads	13
5.3	qPCR	13
5.4	Evaluation metrics	14
	Appendix A Derivation of marginalization	14

Appendix B	Learning negative binomial dispersion parameters	15
B.1	Model Details	15
B.2	Initialization	15
B.3	Model Inference	16
B.3.1	Latent and auxiliary trajectory	16
B.3.2	Negative binomial dispersion parameters	17

1 Notation

We parameterize the Inverse Gamma distribution with shape α and scale β as

$$\text{InvGamma}(x; \alpha, \beta) = \frac{\beta^\alpha}{\Gamma(\alpha)} x^{\alpha-1} \exp\left(-\frac{\beta}{x}\right) \quad (1)$$

We parameterize the Gamma distribution with shape k and scale θ as

$$\text{Gamma}(x; k, \theta) = \frac{1}{\Gamma(k)\theta^k} x^{k-1} \exp\left(-\frac{x}{\theta}\right) \quad (2)$$

We parameterize the Negative Binomial distribution with mean ϕ and dispersion ϵ as:

$$\text{NegBin}(y; \phi, \epsilon) = \frac{\Gamma(r+y)}{y!\Gamma(r)} \left(\frac{\phi}{r+\phi}\right)^y \left(\frac{r}{r+\phi}\right)^r \quad (3)$$

$$r = \frac{1}{\epsilon} \quad (4)$$

Normal distributions are written in terms of the variance $\text{Normal}(\mu, \sigma^2)$.

1.1 Index notation

To indicate the effect from a to b , we use the control theory notation $b_{b,a}$, not graphical notation $z_{a,b}$.

2 Model details

2.1 Dirichlet process for interactions and perturbations

We incorporate a Dirichlet Process (DP)-based clustering technique ([2], [3]) to learn redundant interaction structures (interaction modules) and perturbation effects among OTUs. Let $\mathbf{c}_i \in \mathbb{Z}^+$ be the cluster assignment of OTU i . In the context of our dynamical systems module, only interaction coefficients between modules need to be learned ($\mathbf{b}_{\mathbf{c}_i, \mathbf{c}_j}$), rather than interactions between each pair of OTUs ($b_{i,j}$). If OTU i and OTU j are in different clusters, i.e. $\mathbf{c}_i \neq \mathbf{c}_j$, then $\mathbf{b}_{\mathbf{c}_i, \mathbf{c}_j} \in \mathbb{R}$ is the coefficient representing the (interaction) effect from cluster \mathbf{c}_j to cluster \mathbf{c}_i . If OTU i and OTU j are in the same cluster, i.e. $\mathbf{c}_i = \mathbf{c}_j$, then we assume there is no interaction for interpretability. If OTU l is in the same cluster as OTU j , different than OTU i , then $\mathbf{b}_{\mathbf{c}_i, \mathbf{c}_l} = \mathbf{b}_{\mathbf{c}_i, \mathbf{c}_j}$ by definition.

Additionally, perturbation effects are modeled on the cluster level: γ_{j, \mathbf{c}_i} , where γ_{j, \mathbf{c}_i} is the j^{th} perturbation for the cluster assigned to OTU i . If OTU i and OTU l are in the same cluster, $\gamma_{j, \mathbf{c}_i} = \gamma_{j, \mathbf{c}_l}$. Analogously, if OTU i and OTU l are not in the same cluster, $\gamma_{j, \mathbf{c}_i} \neq \gamma_{j, \mathbf{c}_l}$.

We generate the mixture weights C_1, C_2, \dots from a "stick-breaking" construction with concentration parameter θ :

$$V_1, V_2, \dots \sim \text{Beta}(1, \theta) \quad (5)$$

$$C_k = V_k \prod_{j=1}^{k-1} (1 - V_j) \quad (6)$$

Atom locations η_1, η_2, \dots are drawn from a base distribution G_0 .

$$\eta_1, \eta_2, \dots \sim G_0 \quad (7)$$

$$\Theta = \sum_{k=1}^{\infty} C_k \delta_{\eta_k} \quad (8)$$

where Θ is a draw from a Dirichlet process with concentration parameter θ and base distribution G_0 , or $\Theta \sim \text{DP}(\theta, G_0)$. δ_{η_k} is an indicator function that is 0 everywhere, except for $\delta_{\eta_k}(C_k) = 1$.

To specify the dirichlet process mixture model, we specify cluster assignment variable \mathbf{c}_i for each OTU i , and a density $\pi_{\mathbf{c}}$ which takes an atom location as a parameter.

$$\mathbf{c}_i \sim \text{Multinomial}(C_k) \quad (9)$$

$$i|\mathbf{c}_i \sim \pi_{\mathbf{c}}(\cdot|\eta_{C_k}) \quad (10)$$

2.2 Baseline logistic growth

For the underlying dynamical systems model, we use the stochastic version of the generalized Lotka-Volterra (gLV) equations:

$$\mu_{si}(k+1) = \mathbf{x}_{si}(k) + \mathbf{x}_{si}(k) \left(\mathbf{a}_{i,1} \left(1 + \sum_j \gamma_{j,\mathbf{c}_i} \mathbf{z}_{\mathbf{c}_i}^{(\gamma_j)} h_j(k) \right) \right) + \mathbf{a}_{i,2} \mathbf{x}_{si}(k) \quad (11)$$

$$+ \sum_{\mathbf{c}_i \neq \mathbf{c}_j} \mathbf{b}_{\mathbf{c}_i, \mathbf{c}_j} \mathbf{z}_{\mathbf{c}_i, \mathbf{c}_j}^{(\mathbf{b})} \mathbf{x}_{sj}(k) \Delta_k$$

$$\mathbf{x}_{si}(k+1) \sim \text{Normal}(\mu_{si}(k+1), \Delta_k \sigma_{w,si}^2) \quad (12)$$

where $i \in \{1, \dots, n_o\}$, $s \in \{1, \dots, n_S\}$. $\mathbf{x}_{si}(k) \in \mathbb{R}_{\geq 0}$ is the abundance of OTU i , subject s , at time t_k , $k \in \{1, \dots, n_T\}$, $\Delta_k = t_{k+1} - t_k$. Variables $\mathbf{a}_{1,i}$ and $\mathbf{a}_{2,i}$ represent the growth and self-interaction parameters for each OTU, respectively, and are parameterized by a truncated normal:

$$\mathbf{a}_{i,1} \sim \text{Normal}_{(+)}(0, \sigma_{\mathbf{a}_1}^2) \quad (13)$$

$$\mathbf{a}_{i,2} \sim \text{Normal}_{(-)}(0, \sigma_{\mathbf{a}_2}^2) \quad (14)$$

We place Inverse Gamma priors on the variances:

$$\sigma_{\mathbf{a}_1}^2 \sim \text{InvGamma}(\alpha_{a_1}, \beta_{a_1}) \quad (15)$$

$$\sigma_{\mathbf{a}_2}^2 \sim \text{InvGamma}(\alpha_{a_2}, \beta_{a_2}) \quad (16)$$

Note that a_1 and a_2 correspond to each OTU individually and are not part of our clustering scheme. The variable $\mathbf{b}_{\mathbf{c}_i, \mathbf{c}_j}$ represents the interaction coefficient from cluster \mathbf{c}_i to cluster \mathbf{c}_j . We assume a normal prior.

$$\mathbf{b}_{\mathbf{c}_i, \mathbf{c}_j} \sim \text{Normal}(0, \sigma_{\mathbf{b}}^2) \quad (17)$$

$$\sigma_{\mathbf{b}}^2 \sim \text{InvGamma}(\alpha_b, \beta_b) \quad (18)$$

The variable $\mathbf{z}_{\mathbf{c}_i, \mathbf{c}_j}^{(\mathbf{b})}$ represents the binary indicator variable that selects the interaction from \mathbf{c}_j to \mathbf{c}_i . We assume $\mathbf{z}_{\mathbf{c}_i, \mathbf{c}_j}^{(\mathbf{b})}$ is Bernoulli distributed with probability $\pi_{\mathbf{b}}$

$$\mathbf{z}_{\mathbf{c}_i, \mathbf{c}_j}^{(\mathbf{b})} \sim \text{Bernoulli}(\pi_{\mathbf{b}}) \quad (19)$$

$$\pi_{\mathbf{b}} \sim \text{Beta}(b_{z,1}, b_{z,2}) \quad (20)$$

2.3 Process Variance

The process variance $\sigma_{w,si}^2$ is heteroscedastic, scaling with the magnitude of the trajectory. The parameters controlling the scaling can be learned across all OTUs

$$\sigma_{w,si}^2 = \mathbf{v}_1 \mathbf{q}_{si}^2(k) + \mathbf{v}_2 \hat{c}_w^2 \quad (21)$$

or be specific to each OTU:

$$\sigma_{w,si}^2 = \mathbf{v}_{1i} \mathbf{q}_{si}^2(k) + \mathbf{v}_{2i} \hat{c}_w^2 \quad (22)$$

where \hat{c}_w is the amount of bacterial reintroduction each day. \mathbf{v}_1 and \mathbf{v}_2 are parameterized with truncated normal priors

$$\mathbf{v}_i \sim \text{Normal}_{(+)}(\mu_{\mathbf{v}_i}, \sigma_{\mathbf{v}_i}^2) \quad (23)$$

where $i = 1, 2$.

2.4 Auxiliary Trajectory

In order to model the non-negativity of microbiome data while still maintaining inference efficiency, we apply a relaxation method which has been introduced previously for microbiome dynamics [5]. We closely couple the latent dynamical trajectory $\mathbf{x}_{si}(k)$ to an auxiliary trajectory $\mathbf{q}_{si}(k)$ that is strictly non-negative via a normal distribution

$$(\mathbf{q}_{si}(k) - \mathbf{x}_{si}(k)) \sim \text{Normal}(0, \sigma_q^2) \quad (24)$$

$$\mathbf{q}_{si}(k) \sim \text{Uniform}(0, \infty) \quad (25)$$

$$\sigma_q^2 = v_1^{\text{aux}} \mathbf{q}_{si}^2(k) + v_2^{\text{aux}} \quad (26)$$

where v_1^{aux} and v_2^{aux} are global parameters and are small.

2.5 Perturbations

We assume that the baseline growth rate can be modulated by a perturbation. The variable γ_{j,\mathbf{c}_i} represents the magnitude of perturbation j and has a normal prior.

$$\gamma_{j,\mathbf{c}_i} \sim \text{Normal}(0, \sigma_{\gamma_j}^2) \quad (27)$$

The period of time when perturbation j is active is given by a step function $h_j(k)$

$$h_j(k) = \begin{cases} 1 & t_j^{\text{on}} < t_k \leq t_j^{\text{off}} \\ 0 & \text{otherwise} \end{cases} \quad (28)$$

Additionally, there is a binary indicator variable $\mathbf{z}_{\mathbf{c}_i}^{(\gamma_j)}$ that selects whether cluster \mathbf{c}_i is affected by perturbation γ_j and is parameterized by a Bernoulli.

$$\mathbf{z}_{\mathbf{c}_i}^{(\gamma_j)} \sim \text{Bernoulli}(\pi_{\gamma_j}) \quad (29)$$

$$\pi_{\gamma_j} \sim \text{Beta}(b_{\gamma_j,1}, b_{\gamma_j,2}) \quad (30)$$

2.6 Error model

The observed data are the sequencing counts $\mathbf{y}_{si}(k)$ and qPCR measurements $\mathbf{Q}_s(k)$, which both depend of the auxiliary trajectory $\mathbf{q}_{si}(k)$. We set the error model dependent on the auxiliary trajectory instead of the latent trajectory because it is strictly positive. We can model $\mathbf{y}_{si}(k)$ by a negative binomial distribution [4]

$$\mathbf{y}_{si}(k) \sim \text{NegBin}(\phi_{si}(\mathbf{q}_s(k), r_s(k)), \epsilon_{si}(\mathbf{q}_s(k), a_0, a_1)) \quad (31)$$

$$\phi_{si}(\mathbf{q}_s(k), r_s(k)) = r_s(k) \frac{\mathbf{q}_{si}(k)}{\sum_i \mathbf{q}_{si}(k)} \quad (32)$$

$$\epsilon_{si}(\mathbf{q}_s(k), a_0, a_1) = \frac{a_0}{\mathbf{q}_{si}(k) / \sum_i \mathbf{q}_{si}(k)} + a_1 \quad (33)$$

where $r_s(k)$ is the total number of reads at time t_k for subject s , and a_0 and a_1 are the negative binomial dispersion scaling parameters which are pre-trained on raw reads. We can model $\mathbf{Q}_s(k)$ with a normal distribution with mean as the total auxiliary abundance at time t_k :

$$\mathbf{Q}_s(k) \sim \text{Normal}(\sum_i \mathbf{q}_{si}(k), \sigma_{Q_s(k)}^2) \quad (34)$$

where $\sigma_{Q_s(k)}^2$ is the variance over the triplicate measurements of the qPCR at time t_k and subject s .

3 Model inference

Before we learn the parameters of the model, we learn the negative binomial dispersion parameters (a_0 and a_1) offline and then keep them fixed during inference. To see how inference on these parameters is performed, see appendix B. Inference for all the other parameters is performed using Markov Chain Monte Carlo (MCMC). We use Gibbs steps for variables with conjugate priors and Metropolis-Hastings (MH) steps otherwise. The order of the main sampling loop is as follows:

1. Sample cluster interaction parameters \mathbf{b}
2. Sample growth and self-interaction parameters ($\mathbf{a}_1, \mathbf{a}_2$) by randomly choosing which one to update first
3. Sample logistic growth and interaction prior variance parameters $\sigma_{\mathbf{a}_1}^2, \sigma_{\mathbf{a}_2}^2$, and $\sigma_{\mathbf{b}}^2$
4. Sample perturbation indicator $\mathbf{z}_{\mathbf{c}_i}^{(\gamma_j)}$
5. Sample perturbation probability π_{γ_j}
6. Sample perturbation magnitude γ_{j, \mathbf{c}_i}
7. Sample process variance parameters \mathbf{v}_1 and \mathbf{v}_2
8. Sample latent trajectory $\mathbf{x}_{si}(k)$ and auxiliary trajectory $\mathbf{q}_{si}(k)$
9. Sample cluster assignment \mathbf{c}_i for each OTU i
10. Sample cluster concentration θ
11. Sample the indicator for the interactions $\mathbf{z}^{(\mathbf{b})}$
12. Sample the probability of there being an interaction $\pi_{\mathbf{b}}$

3.1 Logistic growth parameters and perturbation magnitudes

The variables $\mathbf{a}_{i,1}$, $\mathbf{a}_{i,2}$, $\mathbf{b}_{\mathbf{c}_i, \mathbf{c}_j}$, and γ_{j, \mathbf{c}_i} each have conjugate normal priors, so they can be sampled with straightforward Gibbs sampling.

3.2 Logistic growth prior variances

The variables $\sigma_{\mathbf{a}_1}^2$, $\sigma_{\mathbf{a}_2}^2$, and $\sigma_{\mathbf{b}}^2$ have conjugate Inverse Gamma priors, so they are sampled using straightforward Gibbs steps.

3.3 Process variance parameters

\mathbf{v}_1 and \mathbf{v}_2 are each updated with an MH step using a truncated normal proposal ($[0, \infty]$) centered on its current value. For each parameter, the proposal is

$$\mathbf{v}_d^{(*)} \sim \text{Normal}_{(+)}\left(\mathbf{v}_d^{(j-1)}, \sigma_{(\text{prop})}^2(\mathbf{v}_d)\right) \quad (35)$$

for $d \in \{1, 2\}$. The proposal variance $\sigma_{(\text{prop})}^2(\mathbf{v}_d)$ is adjusted to 0.44, the optimal value for a single variable MH step [1]. Every 50 MCMC interactions during the first half of burnin, $\sigma_{(\text{prop})}^2(\mathbf{v}_d)$ is tuned based on the acceptance rate of the previous 50 iterations. Once the tuning stops it is fixed for the rest of inference. Denoting the process variance computed with the proposed parameter by $\sigma_w^{2(*)}$ and the current parameter by $\sigma_w^{2(j-1)}$, the acceptance probability for global parameters \mathbf{v}_i is:

$$r_{\text{accept}} = \frac{\prod_s \left(\prod_i \text{Normal}\left(\frac{\mathbf{x}_{si}(k) - \mu_{si}(k)}{\sqrt{\Delta_k}}; 0, \sigma_w^{2(*)}\right) \right) \text{Normal}_{(+)}\left(\mathbf{v}_d^{(*)}; \mu_{\mathbf{v}_d}, \sigma_{\mathbf{v}_d}^2\right) \text{Normal}_{(+)}\left(\mathbf{v}_d^{(j-1)}; \mathbf{v}_d^{(*)}, \sigma_{\mathbf{v}_d}^2\right)}{\prod_s \left(\prod_i \text{Normal}\left(\frac{\mathbf{x}_{si}(k) - \mu_{si}(k)}{\sqrt{\Delta_k}}; 0, \sigma_w^{2(j-1)}\right) \right) \text{Normal}_{(+)}\left(\mathbf{v}_d^{(j-1)}; \mu_{\mathbf{v}_d}, \sigma_{\mathbf{v}_d}^2\right) \text{Normal}_{(+)}\left(\mathbf{v}_d^{(*)}; \mathbf{v}_d^{(j-1)}, \sigma_{\mathbf{v}_d}^2\right)} \quad (36)$$

If \mathbf{v}_d is specific for each OTU, the acceptance probability changes to:

$$r_{\text{accept}} = \frac{\prod_s \left(\text{Normal}\left(\frac{\mathbf{x}_{si}(k) - \mu_{si}(k)}{\sqrt{\Delta_k}}; 0, \sigma_w^{2(*)}\right) \right) \text{Normal}_{(+)}\left(\mathbf{v}_{di}^{(*)}; \mu_{\mathbf{v}_{di}}, \sigma_{\mathbf{v}_{di}}^2\right) \text{Normal}_{(+)}\left(\mathbf{v}_{di}^{(j-1)}; \mathbf{v}_{di}^{(*)}, \sigma_{\mathbf{v}_{di}}^2\right)}{\prod_s \left(\text{Normal}\left(\frac{\mathbf{x}_{si}(k) - \mu_{si}(k)}{\sqrt{\Delta_k}}; 0, \sigma_w^{2(j-1)}\right) \right) \text{Normal}_{(+)}\left(\mathbf{v}_{di}^{(j-1)}; \mu_{\mathbf{v}_{di}}, \sigma_{\mathbf{v}_{di}}^2\right) \text{Normal}_{(+)}\left(\mathbf{v}_{di}^{(*)}; \mathbf{v}_{di}^{(j-1)}, \sigma_{\mathbf{v}_{di}}^2\right)} \quad (37)$$

where this is calculated for each OTU i . In each Gibbs step, we randomly choose the order to update each OTU for each \mathbf{v}_d .

In each Gibbs step, we randomly choose whether we update \mathbf{v}_1 or \mathbf{v}_2 to enhance mixing.

3.4 Latent and auxiliary trajectory

Since the latent trajectory $\mathbf{x}_{si}(k)$ and auxiliary trajectory $\mathbf{q}_{si}(k)$ are tightly coupled, their joint posterior is likely highly correlated. We can achieve efficient sampling of these parameters by using a multivariate MH step with an adaptive proposal.

At each time point k , OTU i , subject s , and MH step j , the following multivariate jumping proposal is used:

$$\begin{pmatrix} \mathbf{x}_{si}(k)^{(*)} \\ \mathbf{q}_{si}(k)^{(*)} \end{pmatrix} \sim \text{Normal}\left(\begin{pmatrix} \mathbf{x}_{si}(k)^{(j-1)} \\ \mathbf{q}_{si}(k)^{(j-1)} \end{pmatrix}, \Sigma_{\text{prop}}(s, k, i)\right) \quad (38)$$

The proposal covariance $\Sigma_{\text{prop}}(s, k, i)$ is set equal to the empirical covariance of the accepted auxiliary and latent trajectory over all the previous MH steps at time point k and OTU i , multiplied by a scalar $c_{si}(k)$:

$$v_x(s, k, i) = [\mathbf{x}_{si}(k)^{(0)}, \mathbf{x}_{si}(k)^{(1)}, \dots, \mathbf{x}_{si}(k)^{(j-1)}] \quad (39)$$

$$v_q(s, k, i) = [\mathbf{q}_{si}(k)^{(0)}, \mathbf{q}_{si}(k)^{(1)}, \dots, \mathbf{q}_{si}(k)^{(j-1)}] \quad (40)$$

$$\Sigma_{\text{prop}}(s, k, i) = c_{si}(k) \text{Cov}(v_x(s, k, i), v_q(s, k, i)) \quad (41)$$

During the first half of the burnin period, the parameter $c_{si}(k)$ is tuned to adjust the acceptance rate towards 0.234, the optimal acceptance rate for a multivariate MH step [1]. After the tuning is finished, $c_{si}(k)$ is fixed for the rest of inference. Note that there is a separate proposal covariance for each pair of $\mathbf{q}_{si}(k)$ and $\mathbf{x}_{si}(k)$ for each s , k , and i .

The data likelihood function l is specified as follows:

$$l(\mathbf{y}_{si}(k), \mathbf{Q}(k); \mathbf{q}_{si}(k)) = \text{NegBin}(\mathbf{y}_{si}(k); \phi(\mathbf{q}_s(k)), \epsilon(\mathbf{q}_s(k))) \text{Normal}(\mathbf{Q}(k); \sum_i \mathbf{q}_{si}(k), \sigma_{Q_s}^2(k)) \quad (42)$$

where ϕ and ϵ were defined in section 2.6. The unnormalized target distribution conditional on all other parameters Ω is:

$$\begin{aligned} p(\mathbf{x}_{si}(k), \mathbf{q}_{si}(k); \Omega) &\propto \text{Uniform}(\mathbf{q}_{si}(k); 0, \infty) \\ &\times l(\mathbf{y}_{si}(k), \mathbf{Q}(k); \mathbf{q}_{si}(k)) \\ &\times \text{Normal}(\mathbf{x}_{si}(k) - \mathbf{q}_{si}(k); 0, \sigma_q^2) \\ &\times \text{Normal}(\mathbf{x}_{si}(k); \mu_{si}(k), \Delta_k \sigma_{w,si}^2(k)) \\ &\times \text{Normal}(\mathbf{x}_{si}(k+1); \mu_{si}(k+1), \Delta_{k+1} \sigma_{w,si}^2(k+1)) \end{aligned} \quad (43)$$

Note that when we are sampling the last time point ($k = n_T$) we do not calculate the likelihood of the future point (set the log-likelihood to 0). When $k = 0$, we cannot calculate the probability from the previous time point so we sample from the prior of \mathbf{x} . The acceptance probability is calculated as the ratio:

$$r^{\text{accept}} = \frac{p(\mathbf{x}_{si}^{(*)}(k), \mathbf{q}_{si}^{(*)}(k); \Omega) \text{Normal}\left(\begin{pmatrix} \mathbf{x}_{si}(k)^{(j-1)} \\ \mathbf{q}_{si}(k)^{(j-1)} \end{pmatrix}; \begin{pmatrix} \mathbf{x}_{si}(k)^{(*)} \\ \mathbf{q}_{si}(k)^{(*)} \end{pmatrix}, \Sigma_{\text{prop}}(s, k, i)\right)}{p(\mathbf{x}_{si}^{(j-1)}(k), \mathbf{q}_{si}^{(j-1)}(k); \Omega) \text{Normal}\left(\begin{pmatrix} \mathbf{x}_{si}(k)^{(*)} \\ \mathbf{q}_{si}(k)^{(*)} \end{pmatrix}; \begin{pmatrix} \mathbf{x}_{si}(k)^{(j-1)} \\ \mathbf{q}_{si}(k)^{(j-1)} \end{pmatrix}, \Sigma_{\text{prop}}(s, k, i)\right)} \quad (44)$$

Because our proposal distribution is symmetric, the forward and reverse jumping likelihoods are equal to each other, so our acceptance probability simplifies to

$$r^{\text{accept}} = \frac{p(\mathbf{x}_{si}^{(*)}(k), \mathbf{q}_{si}^{(*)}(k); \Omega)}{p(\mathbf{x}_{si}^{(j-1)}(k), \mathbf{q}_{si}^{(j-1)}(k); \Omega)} \quad (45)$$

where

$$p(\mathbf{x}_{si}(k), \mathbf{q}_{si}(k); \Omega) \propto \begin{cases} \left(\sigma_Q \sigma_q \sigma_w(k) \sigma_w(k+1) \sqrt{\Delta_k \Delta_{k+1}} \right)^{-1} \\ \cdot \exp\left(-\frac{1}{2} \left(\frac{(\mathbf{Q}(k) - \sum_i \mathbf{q}_{si}(k))^2}{\sigma_Q^2} + \frac{(\mathbf{x}_{si}(k) - \mathbf{q}_{si}(k))^2}{\sigma_q^2} \right. \right. \\ \left. \left. + \frac{(\mathbf{x}_{si}(k) - \mu_{si}(k))^2}{\Delta_k \sigma_{w,si}^2(k)} + \frac{(\mathbf{x}_{si}(k+1) - \mu_{si}(k+1))^2}{\Delta_{k+1} \sigma_{w,si}^2(k+1)} \right) \right) & \text{if } \mathbf{q}_{si}(k) \geq 0 \\ \cdot \frac{\Gamma(r + \mathbf{y}_{si}(k))}{\mathbf{y}_{si}(k)! \Gamma(r)} \left(\frac{\phi(\mathbf{q}_{si}(k))}{r + \phi(\mathbf{q}_{si}(k))} \right)^{\mathbf{y}_{si}(k)} \left(\frac{r}{r + \phi(\mathbf{q}_{si}(k))} \right)^r & \text{otherwise} \\ 0 & \end{cases}$$

where $r = 1/\epsilon(\mathbf{q}_{si}(k))$. If we eliminate variables that are not dependent on $\mathbf{x}_{si}(k)$ or $\mathbf{q}_{si}(k)$, we are able to simplify to

$$p(\mathbf{x}_{si}(k), \mathbf{q}_{si}(k); \Omega) \propto \begin{cases} \left(\sigma_q \sigma_w(k) \sigma_w(k+1) \right)^{-1} \\ \cdot \exp\left(-\frac{1}{2} \left(\frac{(\mathbf{Q}(k) - \sum_i \mathbf{q}_{si}(k))^2}{\sigma_Q^2} + \frac{(\mathbf{x}_{si}(k) - \mathbf{q}_{si}(k))^2}{\sigma_q^2} \right. \right. \\ \left. \left. + \frac{(\mathbf{x}_{si}(k) - \mu_{si}(k))^2}{\Delta_k \sigma_{w,si}^2(k)} + \frac{(\mathbf{x}_{si}(k+1) - \mu_{si}(k+1))^2}{\Delta_{k+1} \sigma_{w,si}^2(k+1)} \right) \right) & \text{if } \mathbf{q}_{si}(k) \geq 0 \\ \cdot \left(\frac{\phi(\mathbf{q}_{si}(k))}{r + \phi(\mathbf{q}_{si}(k))} \right)^{\mathbf{y}_{si}(k)} \left(\frac{r}{r + \phi(\mathbf{q}_{si}(k))} \right)^r & \text{otherwise} \\ 0 & \end{cases}$$

The next values for $\mathbf{x}_{si}^{(j)}(k)$ and $\mathbf{q}_{si}^{(j)}(k)$ are set given by:

$$\begin{pmatrix} \mathbf{x}_{si}^{(j)}(k) \\ \mathbf{q}_{si}^{(j)}(k) \end{pmatrix} = \begin{cases} \begin{pmatrix} \mathbf{x}_{si}^{(*)}(k) \\ \mathbf{q}_{si}^{(*)}(k) \end{pmatrix} & \text{with probability } \min(1, r^{\text{accept}}) \\ \begin{pmatrix} \mathbf{x}_{si}^{(j-1)}(k) \\ \mathbf{q}_{si}^{(j-1)}(k) \end{pmatrix} & \text{otherwise} \end{cases} \quad (46)$$

3.5 Interaction and perturbation indicator

A Gibbs step is used to update the interaction indicator $\mathbf{z}_{\mathbf{c}_i, \mathbf{c}_j}^{(\mathbf{b})}$ and perturbation indicator $\mathbf{z}_{\mathbf{c}_i}^{(\gamma_j)}$ variable for whether or not an interactions is present. Due to conjugacy, the interactions and perturbations (\mathbf{a}_2 , $\mathbf{b}_{\mathbf{c}_i, \mathbf{c}_j}$, and γ_{j, \mathbf{c}_i}) can be integrated out, conditioned on the growth rates \mathbf{a}_1 . A derivation of this marginalization can be seen in appendix A. In the following equations, let $F_u(\mathbf{z}_{\mathbf{c}_i, \mathbf{c}_j}^{(\mathbf{b})} | \mathbf{a}_1)$, $u \in \{0, 1\}$ be the marginal likelihood that $\mathbf{z}_{\mathbf{c}_i, \mathbf{c}_j}^{(\mathbf{b})} = u$.

$$F_u(\mathbf{z}_{\mathbf{c}_i, \mathbf{c}_j}^{(\mathbf{b})} | \mathbf{a}_1) = \int \prod_s \prod_k \text{Normal}(\mathbf{x}_{si}(k); \mu_{si}(k, \mathbf{c}_i, \mathbf{c}_j, u), \Delta_k \sigma_{w, si}^2) d(\mathbf{a}_2, \gamma_j, \mathbf{b}) \quad (47)$$

where $\mu_{si}(k, \mathbf{c}_i, \mathbf{c}_j, u)$ indicates the mean (defined in equation 11) computed when $\mathbf{z}_{\mathbf{c}_i, \mathbf{c}_j}^{(\mathbf{b})} = u$. Let

$$P(\mathbf{z}_{\mathbf{c}_i, \mathbf{c}_j}^{(\mathbf{b})} = 0) = F_0(\mathbf{z}_{\mathbf{c}_i, \mathbf{c}_j}^{(\mathbf{b})} | \mathbf{a}_1) \text{Bernoulli}(0; \pi_{\mathbf{b}}) \quad (48)$$

$$P(\mathbf{z}_{\mathbf{c}_i, \mathbf{c}_j}^{(\mathbf{b})} = 1) = F_1(\mathbf{z}_{\mathbf{c}_i, \mathbf{c}_j}^{(\mathbf{b})} | \mathbf{a}_1) \text{Bernoulli}(1; \pi_{\mathbf{b}}) \quad (49)$$

be the posterior probabilities of assigning $\mathbf{z}_{\mathbf{c}_i, \mathbf{c}_j}^{(\mathbf{b})}$. Given both likelihoods, we sample $u | \Omega$.

3.6 Cluster assignment

The cluster assignment parameter \mathbf{c} is sampled using Gibbs sampling according to Algorithm 8 described in [2]. First, we sample the assignments of OTU i being assigned to each cluster, marginalizing over the interactions and the perturbations. In the following equations, $F_m(\mathbf{x}_{si} | \mathbf{a}_1)$ denotes marginal likelihood for OTU i assigned to cluster m . A derivation of the marginalization can be seen in appendix A. Note that the marginalization must be conditioned on the growth rates \mathbf{a}_1 because they are multiplying the perturbation parameters.

$$F_m(\mathbf{x}_i | \mathbf{a}_1) = \int \prod_s \prod_k \text{Normal}(\mathbf{x}_{si}(k); \mu_{si}^{(m)}(k), \Delta_k \sigma_{w, si}^2) d(\mathbf{a}_2, \gamma_j, \mathbf{b}) \quad (50)$$

where $\mu_{si}^{(m)}$ indicates the mean computed with interaction and perturbatoin parameters from cluster m

$$\begin{aligned} \mu_{si}^{(m)}(k) &= \mathbf{x}_{si}(k-1) + \mathbf{x}_{si}(k-1)(\mathbf{a}_{i,1}(1 + \sum_j h_j \mathbf{z}_m^{(\gamma_j)} \gamma_{j,m}) + \mathbf{a}_{i,2} \mathbf{x}_{si}(k-1) \\ &\quad + \sum_{c_j \neq m} \mathbf{z}_{m, \mathbf{c}_j}^{(\mathbf{b})} \mathbf{b}_{m, \mathbf{c}_j} \mathbf{x}_{sj}(k-1)) \Delta_{k-1} \end{aligned} \quad (51)$$

Conditional on all other parameters Ω , the posterior distribution for cluster assignment \mathbf{c}_i is

$$P(m = c_i | c_{-i}, \Omega) \propto \begin{cases} n_{-i, m} F_m(\mathbf{x}_{si}) & \text{for } m \in \mathbf{c} \\ \theta F_m(\mathbf{x}_{si}) & \text{for } m \text{ new cluster} \end{cases} \quad (52)$$

where $n_{-i, m}$ is the number of OTUs in cluster m without OTU i , and c_{-i} are all the clusters without OTU i in them. Given all these likelihoods, then sample $m | \Omega$. In the case when calculating the likelihood of staying in the OTU's current cluster, if it is a singleton ($n_{-i, m} = 0$), then we set $n_{-i, m} = \theta$.

3.7 Cluster concentration

The cluster concentration parameter θ is sampled according to the auxiliary variable method described in [7]. Our posterior is a mixture of Gamma distributions with auxiliary variable η :

$$P(\theta|\eta, n_c) \sim \pi_\eta \text{Gamma}(\alpha_\theta + n_c, \hat{\beta}_\theta) + (1 - \pi_\eta) \text{Gamma}(\alpha_\theta + n_c - 1, \hat{\beta}_\theta) \quad (53)$$

where $\hat{\beta}_\theta = 1/(1/\beta_\theta + \log(\eta))$, $(\pi_\eta)/(1 - \pi_\eta) = (\alpha_\theta + n_c - 1)/(n(\beta_\theta - \log(\eta)))$, n_c are the current number of clusters, and

$$P(\eta|\theta, n_o) \sim \text{Beta}(\theta + 1, n_o) \quad (54)$$

where n_o are the number of OTUs. At each Gibbs iteration, using the current number of clusters n_c and current value of the cluster concentration θ , we can draw a new value of θ by

1. Sample η using the beta distribution conditioned on current the current value of θ
2. Sample a new θ from the mixture (53) with the current mixture weights

We repeat the above steps within a single Gibbs step until θ settles on a value. In MDSINE 2.0 we repeat the above steps 20 times per Gibbs step.

3.8 Interaction and perturbation indicator probability

The variables $\pi_{\mathbf{c}}$ and π_{γ_j} each have conjugate beta priors, so they can be sampled using straightforward Gibbs steps.

4 Setting hyperparameters and initializations

4.1 Cluster assignment and concentration

We initialize the clusters based on taxonomic similarity, with the number of clusters set to the expected number of clusters using a Dirichlet Process ($\log(n_o)$). We set the prior of the concentration parameter θ to be a Gamma distribution

$$\theta \sim \text{Gamma}(10^{-5}, 10^5) \quad (55)$$

We set the initial value of θ to the mean of the prior.

4.2 Interactions and perturbations

There are three default settings for initializing the hyperparameters of the indicators, depending on the application:

1. Weak 50/50: We set $b_{z,1} = b_{z,2} = b_{\gamma_j,1} = b_{\gamma_j,2} = 0.5$. These hyperparameters correspond to a weak, diffuse prior and have an equal probability of being on/off. This is useful for general, predictive applications.
2. Strong sparse: For applications where you want to push sparse interaction structure, set $b_{z,2} = N(N-1)$, $b_{\gamma_j,2} = N$, and $b_{z,1} = b_{\gamma_j,1} = 0.5$, where $N = \log(n_o)$, the expected number of clusters. This encodes the assumption that there are not even a single interaction or perturbation in the absence of evidence. This is a much stronger prior.

3. Strong dense: For applications where you want to push dense interaction structure and where you expect many OTUs to be affected by the perturbations, set $b_{z,1} = N(N-1)$, $b_{\gamma_j,1} = N$, and $b_{z,2} = b_{\gamma_j,2} = 0.5$, where $N = \log(n_o)$, which encodes the exact opposite of the prior point: In the absence of evidence it assumes that nearly all clusters interact and get affected by the perturbation.

We then set the value of $\pi_{\mathbf{b}}$ and π_{γ_j} to the mean of the prior. We initially assume that there are no interactions ($\mathbf{z}_{\mathbf{c}_i, \mathbf{c}_j}^{(\mathbf{b})} = 0, \forall \mathbf{c}_i \neq \mathbf{c}_j$) and no clusters are affected by the perturbations ($\mathbf{z}_{\mathbf{c}_i}^{(\gamma_j)} = 0, \forall \mathbf{c}_i$). We set the perturbation magnitude prior hyperparameters to be diffuse:

$$\mu_{\gamma_j} = 0 \quad (56)$$

$$\sigma_{\gamma_j}^2 = 100 \quad (57)$$

4.3 Process variance parameters

We set the initial values to $\mathbf{v}_1 = 0.09$ and $\mathbf{v}_2 = 1.0$. \hat{c}_w is set $\hat{c}_w = 10^6$ so that we get 10^6 reintroduction of bacteria each day and there is about 30% variance on the abundances at higher abundance levels. We set the prior hyperparameters of \mathbf{v}_1 and \mathbf{v}_2 to be diffuse:

$$\mu_{\mathbf{v}_1} = 0.09 \quad (58)$$

$$\sigma_{\mathbf{v}_1}^2 = 100 \quad (59)$$

$$\mu_{\mathbf{v}_2} = 1 \quad (60)$$

$$\sigma_{\mathbf{v}_2}^2 = 100 \quad (61)$$

The MH proposal variance is set to be $\sigma_{(\text{prop})}^2(\mathbf{v}_i) = \mu_{\mathbf{v}_i}^2$.

4.4 Logistic growth parameters and prior variances

Our goal is to initialize the hyperparameters of $\sigma_{\mathbf{a}_1}^2$ and $\sigma_{\mathbf{a}_2}^2$ such that the mean is proportional to the data. We first compute a rough estimate for the parameters $\mathbf{a}_{i,1}$ and $\mathbf{a}_{i,2}$ based on the data by solving via linear regression $f = X\Theta$ using the pseudo-inverse solution, indexing out the times that there are perturbations. We enforce positive growth and negative self-interaction parameters as $\hat{\mathbf{a}}_1 = |\mathbf{a}_1|$ and $\hat{\mathbf{a}}_2 = -|\mathbf{a}_2|$. We then set the values $\sigma_{\mathbf{a}_1}^2 = (\frac{1}{n_o} \sum_i \hat{\mathbf{a}}_{i1})^2$ and $\sigma_{\mathbf{a}_2}^2 = (\frac{1}{n_o} \sum_i \hat{\mathbf{a}}_{i2})^2$, the squared empirical mean of those parameters.

Depending if we want weak priors, we set the shape parameters such that $\alpha_{\sigma_{\mathbf{a}_1}^2} = \alpha_{\sigma_{\mathbf{a}_2}^2} = 2$. If we want strong priors, we set $\alpha_{\sigma_{\mathbf{a}_1}^2} = \alpha_{\sigma_{\mathbf{a}_2}^2} = \frac{n_o}{2}$ so that the prior has equal weight as the data during inference. We then $\beta_{\sigma_{\mathbf{a}_1}^2}$ and $\beta_{\sigma_{\mathbf{a}_2}^2}$ so that the expected value of the prior is equal to the initial value:

$$\mathbb{E}[\sigma_{\mathbf{a}_l}^2] = \frac{\beta_{\sigma_{\mathbf{a}_l}^2}}{\alpha_{\sigma_{\mathbf{a}_l}^2} - 1} = \mathbf{a}_l \quad (62)$$

for $l = 1, 2$

4.5 Latent and auxiliary trajectories

If we are simulating intermediate time-points, we initialize the intermediate data by doing a linear interpolation from the nearest given data.

We initialize the values of both the latent (\mathbf{x}) and auxiliary (\mathbf{q}) trajectories by sampling a truncated normal distribution ($[0, \infty)$) where the mean is the data and the variance is the coupling variance σ_q^2 . We set the parameters for the coupling variance to be

$$v_1^{\text{aux}} = 10e^{-3} \quad (63)$$

$$v_2^{\text{aux}} = 10e^{-4} \quad (64)$$

and we fix them during inference.

Additionally, we set the negative binomial dispersion scaling parameters to fixed numbers before had:

$$a_0 = 0.0025 \quad (65)$$

$$a_1 = 0.025 \quad (66)$$

The priors of the latent and auxiliary trajectory are set to be diffuse:

$$\mathbf{q}_{si}(k) \sim \text{Uniform}(0, \infty) \quad (67)$$

$$\mathbf{x}_{si}(k) \sim \text{Normal}(10e^7, 10e^{20}) \quad (68)$$

We initialize the the proposal covariance for each data point to be strongly correlated matrix scaled to the squared auxiliary trajectory without a scaling:

$$\Sigma_{\text{prop}}(s, k, i) = \frac{1}{10} \begin{bmatrix} \mathbf{q}_{si}^2(k) & 0.9\mathbf{q}_{si}^2(k) \\ 0.9\mathbf{q}_{si}^2(k) & \mathbf{q}_{si}^2(k) \end{bmatrix} \quad (69)$$

$$c_{si}(k) = 1 \quad (70)$$

4.6 Staged initialization

4.6.1 First version

Hold clustering Fixed for 100 Gibbs steps from initialization, hold process variance fixed for 150 Gibbs steps from initialization, hold growth and self-interactions fixed for 200 Gibbs steps from initialization. Allows interactions and perturbations to settle on decent values before we start full mixing.

Positives:

- Works very well if growth and self-interactions have a good initialization

Negatives:

- Works very bad if growth and self-interactions dont have a good initialization

4.6.2 Second (current) version

Hold process variance fixed for 150 Gibb steps, hold clustering fixed for 200 Gibb steps. **Positives:**

- More robust to bad initializations of the growth and self-interactions

Negatives:

- Longer period that it is just mixing around not collapsing

Both methods become equivalent after 1000 Gibb steps.

5 Benchmarking

To benchmark the predictive capabilities and test the model, ground truth information is necessary. We use simulated data from generated artificial microbial dynamical systems to benchmark MDSINE 2.0. We seek to mimic key features of real microbiome time-series data with noisy count-based and qPCR measurements, sparse temporal sampling. The simulated data was used to benchmark the performance of the model such as calculating the accuracy of the growth rates and interaction coefficient estimates, recapitulating the the underlying network structure, accuracy of noise estimates, and the accuracy of predicting trajectories given initial conditions not previously seen by the algorithm.

5.1 Data simulation

1. Sample each OTU steady-state with replacement from the given input vector
2. Growth values sampled from Uniform(0.85, 1.5). Self-interactions = $-ss^{-1}$
3. Specify cluster assignment for each OTU
 - (a) Specify cluster evenness: even, user specified. (TODO: Heavy-tail?)
4. Sample topology
 - (a) Specify sparsity: dense ($p = 0.75$), sparse ($p = 0.25$), even ($p = 0.5$)
 - (b) $z_{c_i, c_j} = \text{Bernoulli}(p)$ for each $c_i \neq c_j$
5. Sample interaction matrix with strict stability constraints. For each cluster c :
 - (a) Calculate the budget \hat{v} : The budget is the maximum allowed amount of interactions going into a cluster for the system to be strictly stable. $\hat{v} = \min_{i \in c}(|ss_i|)$
 - (b) For each interaction with a positive indicator:
 - i. Sample magnitude of the interaction: $\hat{b}_{c_i, c_j} = \text{Uniform}(0, \hat{v})$. If the current interaction is the last positive interaction going into the cluster, the rest of the budget is assigned to it: $\hat{b}_{c_i, c_j} = \hat{v}$
 - ii. Sample the sign of the interaction with given probability of being False π_n : $\hat{s} = (-1)^{\text{Bernoulli}(\pi_n)}$.
 - iii. Set the interaction: $b_{c_i, c_j} = \hat{s}\hat{b}_{c_i, c_j}$
 - iv. Decrease the budget for next iteration: $\hat{v} = \hat{v} - \hat{b}_{c_i, c_j}$
 - (c) Optional: Scale the off diagonal interactions by a scalar to push the strict stability constraint: $b_{c_i, c_j} = mb_{c_i, c_j}$, $m > 1$, $c_i \neq c_j$
6. Validation of system: Steady state orthant check ($-A^{-1}r$). Resample if greater than 10% negative/equal to zero

5.2 Simulating noise

We simulate gLV trajectories from section 5.1 and then we need to simulate noisy reads and biomass measurements. We simulate each by the following:

5.2.1 Reads

We simulate the read depth $r_s(k)$ of our system with a negative binomial distribution that was fitted to real data. Reads are then sampled from effectively a Dirichlet-multinomial distribution. We calculate the relative abundance of each of the trajectories generated $\pi_{si}(k) = x_{si}(k) / \sum_i x_{si}(k)$ and then sample the concentration of each of the OTUs with a Dirichlet distribution $\hat{\pi}_{si}(k) \sim \text{Dirichlet}(\pi_{si}(k), \theta)$. The dispersion parameter θ was set to 286 as done previously in [6]. The reads $y_{si}(k)$ are then sampled from a multinomial with the given read depth $r_s(k)$ and concentrations $\hat{\pi}_{si}(k)$: $y_{si}(k) \sim \text{Multinomial}(r_s(k), \hat{\pi}_{si}(k))$.

5.3 qPCR

Biomass data was sampled from a log-normal distribution that was fitted to a lognormal distribution, where the mean is the sum of the abundances of the trajectories generated from section 5.1: $Q_s(k) \sim \text{Lognormal}(\sum_i x_{si}(k))$.

5.4 Evaluation metrics

TODO

A Derivation of marginalization

Let

$$y \sim \text{Normal}(Xw, \Sigma_1) \quad (71)$$

with the prior on w given as

$$w \sim \text{Normal}(\mu_2, \Sigma_2) \quad (72)$$

The design matrix X has dimensions $n \times d$.

$$p_{y|w}p_w = \frac{1}{(2\pi)^{n/2}|\Sigma_1|^{1/2}} \frac{1}{(2\pi)^{d/2}|\Sigma_2|^{1/2}} \exp\left(-\frac{1}{2}(y - Xw)^T \Sigma_1^{-1}(y - Xw) - \frac{1}{2}(w - \mu_2)^T \Sigma_2^{-1}(w - \mu_2)\right) \quad (73)$$

$$= \frac{1}{(2\pi)^{n/2}|\Sigma_1|^{1/2}} \frac{1}{(2\pi)^{d/2}|\Sigma_2|^{1/2}} \quad (74)$$

$$\cdot \exp\left(-\frac{1}{2}(w^T(X^T \Sigma_1^{-1} X + \Sigma_2^{-1})w - (y^T \Sigma_1^{-1} X + \mu_2^T \Sigma_2^{-1})w)\right) \quad (75)$$

$$\cdot \exp\left(-\frac{1}{2}(-w^T(X^T \Sigma_1^{-1} y + \Sigma_2^{-1} \mu_2) + \mu_2^T \Sigma_2^{-1} \mu_2 + y^T \Sigma_1^{-1} y)\right) \quad (76)$$

Using the following definitions

$$\Sigma_3^{-1} = X^T \Sigma_1^{-1} X + \Sigma_2^{-1} \quad (77)$$

$$\mu_3 = \Sigma_3(X^T \Sigma_1^{-1} y + \Sigma_2^{-1} \mu_2) \quad (78)$$

we can simplify the above to

$$p_{y|w}p_w = \frac{1}{(2\pi)^{n/2}|\Sigma_1|^{1/2}} \frac{1}{(2\pi)^{d/2}|\Sigma_2|^{1/2}} \exp\left(-\frac{1}{2}(w^T \Sigma_3^{-1} w - \mu_3^T \Sigma_3^{-1} w - w^T \Sigma_3^{-1} \mu_3)\right) \quad (79)$$

$$\cdot \exp\left(-\frac{1}{2}(\mu_2^T \Sigma_2^{-1} \mu_2 + y^T \Sigma_1^{-1} y)\right) \quad (80)$$

Completing the square on the first line exponent

$$p_{y|w}p_w = \frac{1}{(2\pi)^{n/2}|\Sigma_1|^{1/2}} \frac{1}{(2\pi)^{d/2}|\Sigma_2|^{1/2}} \exp\left(-\frac{1}{2}(w - \mu_3)^T \Sigma_3^{-1}(w - \mu_3)\right) \quad (81)$$

$$\cdot \exp\left(-\frac{1}{2}(-\mu_3^T \Sigma_3^{-1} \mu_3 + \mu_2^T \Sigma_2^{-1} \mu_2 + y^T \Sigma_1^{-1} y)\right) \quad (82)$$

Before we integrate out w , we need to multiply and divide by $|\Sigma_3|^{1/2}$ to make this normal with respect to w

$$p_{y|w}p_w = \frac{1}{(2\pi)^{n/2}|\Sigma_1|^{1/2}} \frac{|\Sigma_3|^{1/2}}{|\Sigma_2|^{1/2}} \frac{1}{(2\pi)^{d/2}|\Sigma_3|^{1/2}} \exp\left(-\frac{1}{2}(w - \mu_3)^T \Sigma_3^{-1}(w - \mu_3)\right) \quad (83)$$

$$\cdot \exp\left(-\frac{1}{2}(-\mu_3^T \Sigma_3^{-1} \mu_3 + \mu_2^T \Sigma_2^{-1} \mu_2 + y^T \Sigma_1^{-1} y)\right) \quad (84)$$

Now we have an exact multivariate Gaussian $\frac{1}{(2\pi)^{d/2}|\Sigma_3|^{1/2}} \exp\left(-\frac{1}{2}(w - \mu_3)^T \Sigma_3^{-1}(w - \mu_3)\right)$. Now if we integrate out w the multivariate normal integral is 1 and

$$\int p_{y|w}p_w dw = \frac{1}{(2\pi)^{n/2}|\Sigma_1|^{1/2}} \frac{|\Sigma_3|^{1/2}}{|\Sigma_2|^{1/2}} \exp\left(-\frac{1}{2}(\mu_2^T \Sigma_2^{-1} \mu_2 + y^T \Sigma_1^{-1} y - \mu_3^T \Sigma_3^{-1} \mu_3)\right) \quad (85)$$

B Learning negative binomial dispersion parameters

We learn the negative binomial dispersion parameters a_0 and a_1 offline before we learn the other parameters of the model and keep them fixed during inference. Inference of a_0 and a_1 is done using MCMC with Metropolis-Hastings steps with the same error model that was defined in section 2.6. The fundamental difference in the inference is that we assume that the latent trajectory \mathbf{x} was generated from the replicate data instead of dynamics. We use Metropolis-Hastings MCMC updates to update \mathbf{x} , \mathbf{q} , a_0 , and a_1 .

B.1 Model Details

We assume that the latent state \mathbf{x}_{si} for OTU i and sample s was generated from a normal distribution:

$$\mathbf{x}_{si} \sim \text{Normal}(\mu_{si}, \sigma_{si}^2) \quad (86)$$

and the auxiliary trajectory \mathbf{q}_i has a uniform prior and is tightly coupled to \mathbf{x}_i the same way as in equation 24:

$$\mathbf{q}_{si} \sim \text{Uniform}(0, \infty) \quad (87)$$

$$\mathbf{q}_{si} \mid \mathbf{x}_{si} \sim \text{Normal}(\mathbf{x}_{si}, \sigma_q^2) \quad (88)$$

$$\sigma_q^2 = v_1^{\text{aux}} \mathbf{q}_{si}^2 + v_2^{\text{aux}} \quad (89)$$

where v_1^{aux} and v_2^{aux} are small values.

We assume that the reads, \mathbf{y}_{si} , were generated from the same negative binomial distribution as specified in equation 31:

$$\mathbf{y}_{si} \mid \mathbf{q}_{si} \sim \text{NegBin}(\phi(\mathbf{q}_{si}), \epsilon(\mathbf{q}_{si}, a_0, a_1)) \quad (90)$$

$$\phi(\mathbf{q}_{si}) = r_s \frac{\mathbf{q}_{si}}{\sum_{si} \mathbf{q}_{si}} \quad (91)$$

$$\epsilon(\mathbf{q}_{si}, a_0, a_1) = \frac{a_0}{\mathbf{q}_{si} / \sum_i \mathbf{q}_{si}} + a_1 \quad (92)$$

where r_s is the read depth for sample s . We assume a_0 and a_1 have uniform priors [6]:

$$a_0, a_1 \sim \text{Uniform}(a_{\text{low}}, a_{\text{high}}) \quad (93)$$

B.2 Initialization

We set μ_i and σ_i^2 as:

$$V_{s_k i} = \mu_{Q_s} \frac{r_{s_k i}}{\sum_{i=1}^{n_o} r_{s_k i}} \quad (94)$$

$$\mu_{si} = \text{mean}_k(V_{s_k i}) \quad (95)$$

$$\sigma_{si}^2 = \text{var}_k(V_{s_k i}) \quad (96)$$

$$(97)$$

where s_k is the k^{th} replicate of the sample s , $r_{s_k i}$ are the reads for OTU i for sample replicate s_k , n_o are the number of OTUs, and μ_{Q_s} is the mean of the triplicate qPCR measurements of the sample s . We initialize \mathbf{x}_i and \mathbf{q}_i sampling around the prior of the latent state with the coupling variance:

$$\mathbf{q}_{si} \sim \text{Normal}_{(+)}(\mu_{si}, \sigma_q^2) \quad (98)$$

$$\mathbf{x}_{si} \sim \text{Normal}_{(+)}(\mu_{si}, \sigma_q^2) \quad (99)$$

We set $a_{\text{low}} = 0$ and $a_{\text{high}} = 10^5$, the same prior that was used in [6] because it is sufficiently diffuse. We set the initial value of $a_0 = 0.001$ and $a_1 = 0.01$.

B.3 Model Inference

B.3.1 Latent and auxiliary trajectory

We use a similar inference method that was used in section 3.4, the major difference is that we assume the latent state is generated from replicate data instead of dynamics. Everything else is very similar but adapted to deal with replicate data instead of time series data.

For each set of samples s , OTU i , and MH step j , the following proposal is used:

$$\begin{pmatrix} \mathbf{x}_{si}^{(*)} \\ \mathbf{q}_{si}^{(*)} \end{pmatrix} \sim \text{Normal} \left(\begin{pmatrix} \mathbf{x}_{si}^{(j-1)} \\ \mathbf{q}_{si}^{(j-1)} \end{pmatrix}, \Sigma_{\text{prop}}(s, i) \right) \quad (100)$$

The proposal variance $\Sigma_{\text{prop}}(s, i)$ is set to the empirical covariance of the accepted auxiliary and latent trajectory over all the previous MH steps for sample s and OTU i , multiplied by a scalar $c(s, i)$:

$$v_x(s, i) = [\mathbf{x}_{si}^{(0)}, \mathbf{x}_{si}^{(1)}, \dots, \mathbf{x}_{si}^{(j-1)}] \quad (101)$$

$$v_q(s, i) = [\mathbf{q}_{si}^{(0)}, \mathbf{q}_{si}^{(1)}, \dots, \mathbf{q}_{si}^{(j-1)}] \quad (102)$$

$$\Sigma_{\text{prop}}(s, i) = c(s, i) \text{Cov}(v_x(s, i), v_q(s, i)) \quad (103)$$

During the first half of the burnin period, the parameter $c(s, i)$ is tuned to adjust the acceptance rate towards 0.234, the optimal acceptance rate for a multivariate MH step [1]. After the tuning is finished, $c(s, i)$ and $\Sigma_{\text{prop}}(s, i)$ are fixed for the rest of inference. Note that there is a separate proposal covariance for each pair of \mathbf{q}_{si} and \mathbf{x}_{si} for each s , and i . We initialize $\Sigma_{\text{prop}}(s, i)$ and $c(s, i)$ to:

$$\Sigma_{\text{prop}}(s, i) = \frac{1}{100} \begin{bmatrix} \mathbf{q}_{si}^2(k) & 0.9\mathbf{q}_{si}^2(k) \\ 0.9\mathbf{q}_{si}^2(k) & \mathbf{q}_{si}^2(k) \end{bmatrix} \quad (104)$$

$$c(s, i) = 1 \quad (105)$$

The unnormalized target distribution conditional on all other parameters Ω is:

$$\begin{aligned} p(\mathbf{x}_{si}, \mathbf{q}_{si}; \Omega) &\propto \text{Uniform}(\mathbf{q}_{si}; 0, \infty) \\ &\times \text{Normal}(\mathbf{x}_{si} - \mathbf{q}_{si}; 0, \sigma_q^2) \\ &\times \text{Normal}(\mathbf{x}_{si}; \mu_{si}, \sigma_{si}^2) \\ &\times \prod_k^{n_s} \text{NegBin}(\mathbf{y}_{s_k i}; \phi(\mathbf{q}_{si}), \epsilon(\mathbf{q}_{si}, a_0, a_1)) \\ &\times \text{Normal}(\sum_i \mathbf{q}_{si}; \mu_{Q_s}, \sigma_{Q_s}^2) \end{aligned} \quad (106)$$

where n_s are the number of replicates for sample s . The acceptance probability is calculated as the ratio:

$$r^{\text{accept}} = \frac{p(\mathbf{x}_{si}^{(*)}, \mathbf{q}_{si}^{(*)}; \Omega) \text{Normal} \left(\begin{pmatrix} \mathbf{x}_{si}^{(j-1)} \\ \mathbf{q}_{si}^{(j-1)} \end{pmatrix}; \begin{pmatrix} \mathbf{x}_{si}^{(*)} \\ \mathbf{q}_{si}^{(*)} \end{pmatrix}, \Sigma_{\text{prop}}(s, i) \right)}{p(\mathbf{x}_{si}^{(j-1)}, \mathbf{q}_{si}^{(j-1)}; \Omega) \text{Normal} \left(\begin{pmatrix} \mathbf{x}_{si}^{(*)} \\ \mathbf{q}_{si}^{(*)} \end{pmatrix}; \begin{pmatrix} \mathbf{x}_{si}^{(j-1)} \\ \mathbf{q}_{si}^{(j-1)} \end{pmatrix}, \Sigma_{\text{prop}}(s, i) \right)} \quad (107)$$

Because our proposal distribution is symmetric, the forward and reverse jumping likelihoods are equal to each other, so our acceptance probability simplifies to

$$r^{\text{accept}} = \frac{p(\mathbf{x}_{si}^{(*)}, \mathbf{q}_{si}^{(*)}; \Omega)}{p(\mathbf{x}_{si}^{(j-1)}, \mathbf{q}_{si}^{(j-1)}; \Omega)} \quad (108)$$

The values for \mathbf{x}_{si} and \mathbf{q}_{si} at MH step j are set given by:

$$\begin{pmatrix} \mathbf{x}_{si}^{(j)} \\ \mathbf{q}_{si}^{(j)} \end{pmatrix} = \begin{cases} \begin{pmatrix} \mathbf{x}_{si}^{(*)} \\ \mathbf{q}_{si}^{(*)} \end{pmatrix} & \text{with probability } \min(1, r^{\text{accept}}) \\ \begin{pmatrix} \mathbf{x}_{si}^{(j-1)} \\ \mathbf{q}_{si}^{(j-1)} \end{pmatrix} & \text{otherwise} \end{cases} \quad (109)$$

B.3.2 Negative binomial dispersion parameters

We use Metropolis-Hastings updates with an adaptive proposal to update the values of a_0 and a_1 . Since the inference scheme is the same between a_0 and a_1 , we will use a_m to describe both.

The following jumping proposal is used at MH step j :

$$a_m^{(*)} \sim \text{Normal}_{(+)}(a_m^{(j-1)}, \sigma_{a_m}^2) \quad (110)$$

The proposal variance $\sigma_{a_m}^2$ is initialized from the initial value of a_m

$$\sigma_{a_m}^2 = \frac{1}{10} a_m^2 \quad (111)$$

During the first half of the burnin period, $\sigma_{a_m}^2$ is tuned to adjust the acceptance rate towards 0.44, the optimal value of a scalar MH step [1]. After the tuning is finished, $\sigma_{a_m}^2$ is fixed for the rest of inference. The unnormalized target distribution conditional on all other parameters Ω is:

$$\begin{aligned} p(a_m; \Omega) &\propto \text{Uniform}(a_m; 0, 10^5) \\ &\times \prod_k^{n_s} \text{NegBin}(\mathbf{y}_{s_k i}; \phi(\mathbf{q}_{si}), \epsilon(\mathbf{q}_{si}, a_0, a_1)) \end{aligned} \quad (112)$$

where n_s are the number of replicates for sample s . The acceptance probability is calculated as the ratio

$$r^{\text{accept}} = \frac{p(a_m^{(*)}; \Omega) \text{Normal}_{(+)}(a_m^{(j-1)}; a_m^{(*)}, \Sigma_{a_m})}{p(a_m^{(j-1)}; \Omega) \text{Normal}_{(+)}(a_m^{(*)}; a_m^{(j-1)}, \Sigma_{a_m})} \quad (113)$$

The value for a_m at MH step j is set by:

$$a_m^{(j)} = \begin{cases} a_m^{(*)} & \text{with probability } \min(1, r^{\text{accept}}) \\ a_m^{(j-1)} & \text{otherwise} \end{cases} \quad (114)$$

References

- [1] A. Gelman, H. S. Stern, J. B. Carlin, D. B. Dunson, A. Vehtari, and D. B. Rubin, *Bayesian Data Analysis Third Edition*. Chapman and Hall/CRC, 2013.
- [2] R. M. Neal, "Markov chain sampling methods for dirchlet process mixture models," *Journal of computational and graphical models*, 9(2):249-265, 2000.
- [3] C. E. Ramussen, "The infinite gaussian mixture model," *Advances in Information Processing Systems 12*, 2000.
- [4] P.J. McMurdie and S. Holmes, "Waste not, want not: why rarefying microbiome data is inadmissible," *PLOS Computational Biology*, vol. 10, no. 4, p. e1003531, 2014.

- [5] T. E. Gibson and G. K. Gerber, "Robust and scalable models of microbiome dynamics", *International Conference on Machine Learning*, pp. 1758-1767, 2018.
- [6] B. Vanni, B. Tzen, N. Ling, M. Simmons, T. Tanoue, E. Bogart, L. Deng, V. Yeliseyev, M. L. Delaney, Q. Liu, B. Olle, R. R. Stein, K. Honda, L. Bry, G. K. Gerber, "Mdsine: Microbial dynamical systems inference engine for microbiome time-series analysis," *Genome Biology*, 17(1):121, 2016.
- [7] M. D. Escobar and M. West, "Bayesian density estimation and inference using mixtures," *Journal of the American Statistical Association*, vol. 90, no. 430, pp 577-588, 1995.