Joint co-clustering and prediction of microbes and metabolites

Liat Shenhav, Jennifer Dawkins, Georg Gerber

July 20, 2022

1 Model

Let N be the number of samples, J the number of metabolites and M the number of taxa. Let Y be an N $\times$ J matrix of samples and metabolites, where each entry y_{ij} represents the normalized abundance of metabolite j in sample i (e.g., z score). Let X be an N $\times$ M matrix of samples and microbial abundances, where each entry x_{mi} represents the abundance of microbe m in sample i. Microbes are combined into L latent features, while metabolites are divisively clustered into K clusters.

1.1 Learning microbial clusters

Based on the embedding location for each microbe $\boldsymbol{a}_m^{\text{taxa}}$, where $\boldsymbol{a}_m^{\text{taxa}}$ is a vector of length D where D is the embedding size, microbes are first aggregated into L latent features g_{il}. Each feature has a mean $\boldsymbol{\mu}_l^{\text{taxa}} \in \mathbb{R}^D$ and radius $r_l^{\text{taxa}} \in \mathbb{R}_+^1$ in the embedding space. We denote the euclidean distance between each taxa location $\boldsymbol{a}_m^{\text{taxa}}$ and microbial feature center $\boldsymbol{\mu}_l^{\text{taxa}}$ as κ_{lm}

$$\begin{aligned}
\kappa_{lm} &= \sqrt{\sum_d^D (\mu_{ld}^{\text{taxa}} - a_{md}^{\text{taxa}})^2} \\
\omega_{lm} &= \text{logistic}((r_l^{\text{taxa}} - \kappa_l^{\text{taxa}})/\tau_\omega)) \\
g_{il} &= \sum_m^M \omega_{lm} X_{im}
\end{aligned}$$

Here, τ_ω is initially set to 1, and decreases to 10^{-1} linearly throughout training.

If we want to run the model without prior locations, we have to change how we model ω. If this is the case, ω is a learned parameter with a prior.

$$\begin{aligned}\omega_{kl} &\sim \text{BinaryConcrete}(\gamma_\omega, \tau_\omega)\\ &\text{where}\\ \gamma_\omega &= 0.1 N_{bug}/N_{bug}\\ \tau_\omega &= 0.5 \rightarrow 10^{-2}\end{aligned}$$

The priors probabilities for the taxa feature centers and radii (after scaling locations to have mean zero and standard-deviation 1):

$$\begin{aligned}\boldsymbol{\mu}_l^{\text{taxa}} &\sim \text{MVN}(0, \text{I} * 100)\\ r_l^{\text{taxa}} &\sim \text{Scale-inv-}\chi^2(\text{dof}_r, \tau_r^2)\\ &\text{where}\\ \tau_r^2 &= \text{mean}_f(r_f^{\text{taxa}})\\ \text{dof}_r &= 0.1\end{aligned}$$

Here, r_f^{taxa} are the radii of each family f in the phylogenetic tree.

**Note: the priors for the radii (microbial and metabolomic) need work, as the degrees of freedom shouldn't be less than 1, though for now the distribution looks to be in the range of what we'd want for a prior. Alternatively, we could change this to truncated normal

1.2 Learning metabolic clusters

Metabolites are clustered divisively into K clusters based on indicator variable $\boldsymbol{z}$, where $\boldsymbol{z}_j$ for each metabolite j is has a categorical prior over the cluster weights $\boldsymbol{\pi}$. The posterior of $\boldsymbol{z}$ given the metabolite locations $\boldsymbol{a}^{\text{metab}}$ is a gaussian mixture model.

$$\begin{aligned}\boldsymbol{z}_j &\sim \text{Categorical}(\pi_0, ..., \pi_k)\\ p(\boldsymbol{z}_j|\boldsymbol{a}^{\text{metab}}, \boldsymbol{\mu}^{\text{metab}}, \boldsymbol{r}^{\text{metab}}, \boldsymbol{\pi}) &\propto \text{Categorical}(\boldsymbol{\pi}) \sum_k z_{jk} \text{MVN}(\boldsymbol{a}_j^{\text{metab}}; \boldsymbol{\mu}_k^{\text{metab}}, \text{I} r_k^{\text{metab}})\end{aligned}$$

The metabolite cluster weights, centers, and radii are distributed as:

$$\begin{aligned}\boldsymbol{\pi} &\sim \text{Dirichlet}(e_0, ...e_0)\\ &\text{where}\\ e_0 &\sim \text{Gamma}(10, 10K)\\ \boldsymbol{\mu}_k^{\text{metab}} &\sim \text{MVN}(0, \text{I} * 100)\\ r_k^{\text{metab}} &\sim \text{Scale-inv-}\chi^2(\text{dof}_r, \tau_r^2)\\ &\text{where}\\ \text{dof}_r &= 0.1\\ \tau_r^2 &= \text{mean}_c(r_c^{\text{metab}})\end{aligned}$$

Here, r_c^{metab} is the radii of each metabolic class/category c. Categories can be defined by several different metrics, but for now we use the classy fire sub-classes

1.3 Learning the relationship between microbial clusters and metabolic clusters

We define weights and selectors between the microbial features and the metabolic clusters with priors:

$$\begin{aligned}
\beta_{kl} &\sim \text{Normal}(0, 1000)\\
\alpha_{kl} &\sim \text{BinaryConcrete}(\gamma, \tau_\alpha)\\
&\quad\text{where}\\
&\quad\gamma = 1/(L * K)\\
&\quad\tau_\alpha = 10^{-0.5} \rightarrow 10^{-3}(\text{decreasing linearly every epoch})
\end{aligned}$$

If we want to learn the number of microbial clusters, we set L to be much larger than the expected number of clusters and add an additional negative binomial prior to to decrease the number of active microbial clusters:

$$P(\alpha_{k1}, ..., \alpha_{kL}) \propto \prod_l \text{NegativeBinomial}(\sum_l \alpha_{kl}; K, 0.1)\text{BinaryConcrete}(\alpha_{kl}; 1/(L * K), \tau_\alpha)$$

We generate metabolic data from microbial features with:

$$y_{ij} = z_{jk}\left[\beta_{0k} + \sum_l \beta_{lk}\alpha_{lk}g_{il} + \epsilon_y\right]$$

Implying that the full data likelihood is:

$$P(\boldsymbol{y}|\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{x}, \boldsymbol{z}) = \prod_{ij}\sum_k z_{jk}\text{Normal}(y_{ij}; \beta_{0k} + \sum_l \beta_{kl}\alpha_{kl}g_{il}, \sigma_y^2)$$

The measurement variance, σ_y^2 is distributed as:

$$\sigma_y^2 \sim \text{HalfNormal}(10)$$

**Note: This prior probably needs work; truncated normal may be better but pytorch doesn't have a truncated normal so I used half normal for now

The full model posterior is as follows:

$$\begin{gathered}
p(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{z}, \boldsymbol{r}^{\text{taxa}}, \boldsymbol{r}^{\text{metab}}, \boldsymbol{\mu}^{\text{taxa}}, \boldsymbol{\mu}^{\text{metab}}, \boldsymbol{\pi}\boldsymbol{\sigma_y^2} \mid \boldsymbol{x}, \boldsymbol{y}, \boldsymbol{a}^{\text{taxa}}, \boldsymbol{a}^{\text{metab}}) =\\
p(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{z}, \boldsymbol{r}^{\text{taxa}}\boldsymbol{\mu}^{\text{taxa}})p(\boldsymbol{\alpha})p(\boldsymbol{\beta})p(\boldsymbol{r}^{\text{taxa}})p(\boldsymbol{\mu}^{\text{taxa}}) \times\\
p(\boldsymbol{z} \mid \boldsymbol{\pi}, \boldsymbol{\mu}^{\text{metab}}, \boldsymbol{r}^{\text{metab}}, \boldsymbol{a}^{\text{metab}})p(\boldsymbol{\mu}_{\text{metab}})p(\boldsymbol{r}^{\text{metab}})p(\boldsymbol{\pi}^{\text{metab}})p(\boldsymbol{\sigma_y^2})
\end{gathered}$$

1.4 Marginalizing over z

Because $\boldsymbol{z}$ is parameterized with a discrete, and non-differentiable, prior, we marginalize out $\boldsymbol{z}$. This gives us the following posterior (here we treat z_j as a cluster indicator rather than a one-hot vector):

$$p(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{r}^{\text{taxa}}, \boldsymbol{r}^{\text{metab}}, \boldsymbol{\mu}^{\text{taxa}}, \boldsymbol{\mu}^{\text{metab}}, \boldsymbol{\pi} \mid \boldsymbol{x}, \boldsymbol{y}, \boldsymbol{a}^{\text{taxa}}, \boldsymbol{a}^{\text{metab}}) =$$
$$\prod_j \sum_{z_j=k} \left[p(\boldsymbol{y}_j|\boldsymbol{x}, \boldsymbol{\alpha}, \boldsymbol{\beta}, z_j = k, \boldsymbol{r}^{\text{taxa}}\boldsymbol{\mu}^{\text{taxa}})p(z_j = k \mid \boldsymbol{\pi}, \boldsymbol{\mu}^{\text{metab}}, \boldsymbol{r}^{\text{metab}}, \boldsymbol{a}_j^{\text{metab}})\right]$$
$$\times p(\boldsymbol{\alpha})p(\boldsymbol{\beta})p(\boldsymbol{r}^{\text{taxa}})p(\boldsymbol{\mu}^{\text{taxa}}) \times p(\boldsymbol{\mu}_{\text{metab}})p(\boldsymbol{r}^{\text{metab}})p(\boldsymbol{\pi}^{\text{metab}})p(\boldsymbol{\sigma_y^2})$$

Where

$$\prod_j \sum_{z_j=k} p(\boldsymbol{y}_j|\boldsymbol{x}, \boldsymbol{\alpha}, \boldsymbol{\beta}, z_j = k, \boldsymbol{r}^{\text{taxa}}\boldsymbol{\mu}^{\text{taxa}})p(z_j = k|\boldsymbol{\pi}, \boldsymbol{\mu}^{\text{metab}}, \boldsymbol{r}^{\text{metab}}, \boldsymbol{a}^{\text{metab}}) =$$

$$\prod_j \sum_k \left[\prod_i [\text{Normal}(y_{ij}; \beta_{0k} + \sum_l \beta_{kl}\alpha_{kl}g_{il}, \sigma_y^2)]\pi_k \text{MVN}(a_j^{\text{metab}}; \mu_k^{\text{metab}}, \text{I}r_k^{\text{metab}})\right]$$

1.5 Learning the number of metabolite clusters

To learn the number of clusters, we set K to be some number much larger than the number of clusters we expect to have, and attempt to obtain a sparse solution where the number of non-empty clusters is equal to K^{true}.

If we have information on metabolite classes or categories, we set K equal to the number of unique metabolic categories. Otherwise, we set $K = N_{\text{metab}}/3$

Following the method outlined in Malsiner-Walli et al. (2014), we add a hierarichal prior to the cluster weights $\boldsymbol{\pi}$.

We model $\boldsymbol{\pi}$ with a Dirichlet distribution over concentration parameters e_0. e_0 is parameterized with a gamma distribution to have the expectation $\mathbf{E}[e_0] = 1/K$ and with a variance $1/(aK^2)$ small enough to restrict large values of e_0 and allow emptying of superfluous components. To do this, we set $a = 10$.

$$\boldsymbol{\pi} \sim \text{Dirichlet}(e_0, ..., e_0)$$
$$e_0 \sim \text{Gamma}(a, a \times K)$$

1.6 Learning the number of microbial features

To learn the number of microbial features L^{true} from a large number of features L, we can use a negative binomial prior and either introduce a feature selector vector of length L, $\boldsymbol{\eta}$, or add a negative binomial prior to $\sum_l \alpha_{kl}$. This last method is the simplest and results in the prior for $\boldsymbol{\alpha}_k$:

$$P(\alpha_{k1}, .., \alpha_{kL}) \propto \prod_l \text{NegativeBinomial}(\sum_l \alpha_{kl}; L, 0.1)$$
$$\text{BinaryConcrete}(\alpha_{kl}; \lambda, \tau)$$

We set $\lambda = \frac{1}{L \times K}$

2 Initializations

$$\begin{aligned}
\beta_{lk}^{(0)} &\sim \text{N}(0,1)\\
\alpha_{lk}^{*(0)} &\sim \text{Normal}(0,1)\\
\alpha_{lk}^{(0)} &= \text{round}(\text{sigmoid}(\alpha_{lk}^{*(0)}))\\
\sigma_y^{(0)} &= \sigma_{est}
\end{aligned}$$

Here, σ_{est} is the estimated measurement variance from the data.

To initialize the microbe cluster centers and radii, we perform K-means clustering with L clusters. We set:

$$\begin{aligned}
\mu_l^{taxa,(0)} &= [\text{Kmeans center}]_l\\
r_l^{taxa,(0)} &= 1.5 * ([\text{Kmeans radii}]_l + 0.01\text{max}_l([\text{Kmeans radii}])
\end{aligned}$$

We add a small number to the Kmeans radii so that no radii are initialized to zero, and we multiply by 1.5 because in experiments with synthetic data, larger radii helped the model converge by reducing the likelihood that microbial radii didn't encompass any microbes

We initialize the metabolite cluster centers and radii in the same way.

If we are learning the number of metabolite and microbe clusters AND we have input classes of metabolites and/or phylogenetic families, we set:

$$K = \text{Number of unique metabolite classes} \tag{1}$$
$$L = \text{Number of unique microbe families} \tag{2}$$

if we don't have this information but are learning the number of clusters, we set:

$$K = N_{metab}/3 \tag{3}$$
$$L = N_{bug}/3 \tag{4}$$

If we don't have any prior embedded locations for the microbes, we initialize:

$$\begin{aligned}
\omega_{kl}^{*(0)} &\sim \text{Normal}(0,1)\\
\omega_{kl}^{(0)} &= \text{round}(\text{sigmoid}(\omega_{kl}^{*(0)}))
\end{aligned}$$

If we are learning the number of metabolic clusters, we initialize:

$$\begin{aligned}
e_0^{(0)} &= 1/K\\
\boldsymbol{\pi}^{(0)} &\sim \text{Dirichlet}(e_0, ..., e_0)
\end{aligned}$$

Because several parameters are constrained to a certain range, we have to initialize these parameters in an unconstrained form for the model to learn, and then transform them to their constrained form in the loss function.

$$\begin{aligned}
r_k^{*(0),\text{metab}} &= \log(r_k^{\text{metab},(0)}) \\
r_l^{*(0),\text{taxa}} &= \log(r_l^{\text{taxa},(0)}) \\
\sigma_y^{*(0)} &= \log(\sigma_y^{(0)}) \\
e_0^{*(0)} &= \log(e_0^{(0)}) \\
\pi^{*(0)} &= \log(\boldsymbol{\pi}^{(0)})
\end{aligned}$$

3 Inference

We perform inference using RMS prop with learning rates adapted to each parameters' size, or the range in which we expect the parameter to stay. Estimated parameter scales are calculated from the mean and standard deviation of the priors. (i.e., estimated parameter size is found by taking 1000 samples from the prior and taking the mean + standard deviation minus the mean - standard deviation. For parameters that are transformed to be within a certain range, we take the reverse transformation of the sample)

Per-parameter learning rates are calculated by scaling the learning rate by the parameter size, as relative to the regression parameter learning rates β. The estimated scale per parameter are: $\beta = 8, \alpha^* = 4, \mu = 40, r^* = 1.5, \pi^* = 1.5, e^* = 0.5, \sigma^{2*} = 0.1$. The overall learning rate is set to 0.01 and the per-parameter learning rates are scaled accordingly.

In the future, we may adjust and simplify this method given the scale of real data, but for now we leave it general.

4 Data generation

1. Given the number of metabolites and the number of clusters, randomly select metabolite indices for each cluster
2. Do the same for microbes, but allow microbes to be in more than one cluster if repeat_clusters = True; if this is the case, randomly choose i indices that have already been chosen to be in the next cluster, where $i = \text{len(chosen)} \times \text{overlap}^{\cdot}$
3. For both microbes and metabolites, generate a distance matrix where the distance between features in the same cluster is 1, and the distance between features not in the same cluster is 1 + dist_frac.
4. Generate a d-dimensional embedding from the generated distance matrix using multi-dimensional scaling

5. Set values for β where the only non-zero terms are on the diagonal
6. Set values for $alpha$, where $alpha = 1$ everywhere
7. Set microbial cluster sum ranges for up to 10 clusters as the following: (100,250), (350,410), (510,550), (650, 770), (870, 900), (1000, 1100), (1200, 1300), (1400, 1500), (1600,1700), (1800,1900)
8. For each microbe cluster sum, sample $g_{ls} \sim \text{Uniform}(c_{l0}, c_{l1})$ where c is the cluster range above, for all s samples
9. For each cluster i, generate the microbe values in the cluster by $x_s \sim \text{Multinomial}(g_{sl}, p)$, where $p \sim \text{Dir}(g_{sl}/L, ..., g_{sl}/L)$
10. Divide each sample in x by the sum of the microbes in the sample to obtain relative abundance, and do the same for g
11. Generate y from $y \sim N(\sum_k \beta_{0k} + \beta_{lk} g_{lk} \alpha_{lk}, \sigma^2_{meas})$ if we are generating linear data
12. If we are generating non-linear data, replace g_{lk} above by $f(g_{lk})$ where f is a non-linear function