Tradict - mathematical details

Surojit Biswas, Konstantin Kerner, Paulo José Pereira Lima Texeira,
Jeffery L. Dangl, Vladimir Jojic, Philip A. Wigge

Contents

This document describes the full mathematical details for the concepts presented in the "Tradict algorithm" section, "Building a predictive Multivariate Normal Continuous-Poisson hierarchical model" subsection of the Materials and Methods in the Supplemental Information. Specifically, we present exactly how Tradict uses a selected set of markers to 1) complete the encoding, and 2) to perform decoding.

1 Preliminaries

For a matrix A, $A_{:i}$ and $A_{i:}$ index the i^{th} column and row, respectively. For a set of indices, q, we use $-q$ to refer to all indices not specified by q.

2 Model

Tradict uses a Continuous-Poisson Multivariate Normal (CP-MVN) hierarchical model to model the expression of transcriptional programs and all genes in the transcriptome. Multivariate Normal hierarchies have been explored in the past as a means of modeling correlation structure among count based random variables [REF]. However, given we will be working with abundances as transcripts per million (TPM), which are non-negative (can equal zero) and fractional, we relax the integral assumption of the Poisson so it is continuous on $[0, \infty)$. Specifically, we define the continuous relaxation of the Poisson distribution (hereafter, Continuous-Poisson) to have the following density function:

$$f(x|\lambda) = C_\lambda \frac{e^{-\lambda}\lambda^x}{\Gamma(x+1)}$$

where C_λ is a normalization constant [REF]. The mean of this distribution is given by λ, just as the Poisson.

We begin by building a predictive model of gene expression, and thereafter discuss a predictive model for the expression of transcriptional programs. Let z_j denote the log-*latent abundance* of gene j, such that

$\exp(z_j)$ is the *latent abundance* of that gene (in TPM) whose measured abundance is given by t_j. Let $T_j = t_j o$ be the measured total number of transcripts of gene j. Here o is the sequencing depth in millions of reads of the sample under consideration. We assume then,

$$\begin{aligned} z &\sim \mathcal{N}(\mu, \Sigma) \\ T_j &\sim \text{Continuous-Poisson}(\exp(z_j)o) \end{aligned}$$

where μ and Σ are of dimension $1 \times$ #-genes and #-genes $\times$ #-genes, respectively. In effect, we are assuming that the measured number of transcripts for gene j is a noisy realization of a latent abundance $\exp(z_j)$ times the sequencing depth, o. The dependencies between log-latent abundances (the z_j's) are then encoded by the covariance matrix of the Multivariate Normal layer of the model.

Note that we could model the TPM measurements directly in the second layer by assuming $t_j \sim$ Continuous-Poisson($\exp(z_j)$); however, this formulation does not consider sequencing depth, which can be a valuable source of information when inferring latent abundances for rare/poorly sampled genes [1].

During decoding, we are interested in building a predictive model between markers and all genes in the transcriptome. Therefore, we need to consider a conditional model of the transcriptome given the log-latent abundances of the markers. Let m be the set of indices for the given panel of selected markers, which are the subset of genes Tradict selects as representative of the transcriptome. To perform prediction we therefore need $p(z_{-m}|z_m)$, and given this we would like to ultimately compute as our estimate of the abundance of all genes in the transcriptome $\hat{T} = \text{argmax}_T\, p(T|z_m)$. We have,

$$\begin{aligned} z_m &\sim \mathcal{N}(\mu^{(m)}, \Sigma^{(m)}) \\ z_{-m}|z_m &\sim \mathcal{N}(\mu_{z_{-m}|z_m}, \Sigma_{z_{-m}|z_m}) \\ T_j &\sim \text{Continuous-Poisson}(\exp(z_j)o) \end{aligned}$$

Here, $\mu^{(m)}$ and $\Sigma^{(m)}$ refer to mean vector and covariance matrix of z_m. Given these, the conditional mean of the log-latent abundances for all non-marker genes can be obtained through Gaussian conditioning. Specifically, for two normally distributed row-vector variables a and b the conditional mean of b given a is given by $\mu_{b|a} = \mu_b + (a - \mu_a)\Sigma_a^{-1}\sigma_{ab}$ and $\Sigma_{b|a} = \Sigma_b - \sigma_{ab}^T\Sigma_a^{-1}\sigma_{ab}$, where σ_{ab} is the cross-covariance between a and b, and Σ_a and Σ_b are the covariance matrices of a and b, respectively.

Given the expression of a transcriptional program is a linear combination of the latent abundances of its constituent genes, they will be normally distributed given 1) Central Limit Theorem, and 2) the latent abundances themselves are normally distributed (convolutions of normals are normals). Let s be the expression of all transcriptional programs. We posit the following model,

$$\begin{aligned} z_m &\sim \mathcal{N}\left(\mu^{(m)}, \Sigma^{(m)}\right) \\ s|z_m &\sim \mathcal{N}(\mu_{s|z_m}, \Sigma_{s|z_m}) \end{aligned}$$

To use these models for prediction, we must learn their parameters from training data. This would complete the process of encoding described in the Supplemental Information. Specifically, we need to learn $\mu^{(m)}$, $\Sigma^{(m)}$, μ_s, $\mu_{z_{-m}}$, $\sigma_{z_m,s}$ and $\sigma_{z_m,z_{-m}}$.

3 Encoding

As described in the Supplemental Information, given an estimate of z_m, $\hat{z}_m$, inference of μ_s, $\mu_{z_{-m}}$, $\sigma_{z_m,s}$ and $\sigma_{z_m,z_{-m}}$ is straightforward. In lag transforming the entire training TPM expression matrix, $t \in \mathbb{R}^{\text{samples}\times\text{genes}}$, we have an estimate of z, $\hat{z} = \text{lag}(t)$ [1]. Thus, an estimate of $\mu_{z_{-m}}$ is given by the usual column-wise sample mean of $\hat{z}_{-m}$.

Let $\Lambda \in \mathbb{R}^{\text{genes}\times\text{transcriptional programs}}$ be a matrix of principal component 1 coefficients over genes for each transcriptional program. Note, that $\Lambda_{ij} = 0$ if gene i is not in transcriptional program j. An estimate of s is given by $\hat{s} = \hat{z}\Lambda$, and so an estimate for μ_s, $\hat{\mu}_s$, is given by the usual column-wise mean of $\hat{s}$.

Given $\hat{z}_m$ the cross-covariances, $\sigma_{z_m,s}$ and $\sigma_{z_m,z_{-m}}$, are given by the usual sample cross-covariance between $\hat{z}_m$ and $\hat{s}$ and between $\hat{z}_m$ and $\hat{z}_{-m}$, respectively.

Now, though we could use the lag-transformed values of t_m as our estimate for z_m, we have an opportunity to improve this estimate by virtue of having to estimate $\mu^{(m)}$ and $\Sigma^{(m)}$. More specifically, given z_m, estimates of $\mu^{(m)}$ and $\Sigma^{(m)}$ are given by – up to some regularization – the usual sample mean and covariance of z_m. Furthermore, given $\mu^{(m)}$ and $\Sigma^{(m)}$, we can update our estimate of z_m to the maximum of its posterior distribution. This suggests an alternating iterative procedure in which we iterate 1) estimation of $\mu^{(m)}$ and $\Sigma^{(m)}$, and 2) maximum *a posteriori* inference of z_m until convergence of their joint likelihood. It is the $\hat{z}_m$ that we obtain from this procedure that we use in the cross-covariance calculations above. The following section details this procedure.

3.1 Inference of z_m given $\mu^{(m)}$ and $\Sigma^{(m)}$

Suppose Tradict has estimates of $\mu^{(m)}$ and $\Sigma^{(m)}$ given by $\hat{\mu}^{(m)}$ and $\hat{\Sigma}^{(m)}$, and let $T_m = t_m(o \times \mathbf{1}_{1\times\text{markers}})$ be a matrix of the total measured number of transcripts for each marker. Here $o \in \mathbb{R}^{\text{samples}\times 1}$ is a vector of sample sequencing depths in millions of reads. Given these, we would like to calculate the maximum *a posteriori* (MAP) estimate of $\hat{z}_m = \text{argmax}_{z_m}\, p(z_m|o, T_m, \hat{\mu}^{(m)}, \hat{\Sigma}^{(m)})$.

The posterior distribution over z_m is given by

$$\begin{aligned}
p(z_m|o, T_m, \hat{\mu}^{(m)}, \hat{\Sigma}^{(m)}) &= \frac{p(T_m|o, z_m, \hat{\mu}^{(m)}, \hat{\Sigma}^{(m)})p(z_m|\hat{\mu}^{(m)}, \hat{\Sigma}^{(m)})}{\int_k p(T_m|o, k, \hat{\mu}^{(m)}, \hat{\Sigma}^{(m)})p(k|\hat{\mu}^{(m)}, \hat{\Sigma}^{(m)})\mathrm{d}k} \\
&\propto \prod_{i=1}^{n} p(T_{im}|o, z_{im}, \hat{\mu}^{(m)}, \hat{\Sigma}^{(m)})p(z_{im}|\hat{\mu}^{(m)}, \hat{\Sigma}^{(m)}) \\
&= \prod_{i=1}^{n} \left[\prod_{j=1}^{|m|} C_{[\exp(z_{ij})o_i]} [\exp(z_{ij})o_i]^{T_{ij}} e^{-[\exp(z_{ij})o_i]} / \Gamma(T_{ij}+1) \right] \\
&\times \frac{1}{\sqrt{2\pi|\hat{\Sigma}^{(m)}|}^{|m|}} \exp\left(-\frac{1}{2}(z_{i:} - \hat{\mu}^{(m)}) \text{inv}\left(\hat{\Sigma}^{(m)}\right)(z_{i:} - \hat{\mu}^{(m)})^T \right)
\end{aligned}$$

where for notational clarity we have used $\text{inv}(\cdot)$ to represent matrix inverse.

Given z is a matrix parameter, this may be difficult to solve directly. However, note that given z_{ij}, T_{ij} is conditionally independent of $T_{i,-j}$. Additionally, given $z_{i,-j}$, z_{ij} is normally distributed with mean and covariance

$$\begin{aligned}
a_{ij} &= \mu_j^{(m)} + \left(z_{i,-j} - \mu_{-j}^{(m)}\right) \text{inv}\left(\Sigma_{-j,-j}^{(m)}\right) \Sigma_{-j,j}^{(m)} \\
\sigma_{m(j)} &= \Sigma_{j,j}^{(m)} - \Sigma_{j,-j}^{(m)} \text{inv}\left(\Sigma_{-j,-j}^{(m)}\right) \Sigma_{-j,j}^{(m)}
\end{aligned}$$

respectively. Taken together, this suggests an iterative conditional modes algorithm [2] in which we maximize the posterior one column of z at a time, while conditioning on all others.

Let $\hat{z}_m$ denote our current estimate of z_m. Let $m(j)$ denote the index of the j^{th} marker and let $m(-j)$

denote the indices of all markers but the j^{th} one. The above sub-objective is given by,

$$\begin{aligned}
\hat{z}_{im(j)} &= \underset{z_{im(j)}|z_{im(-j)}}{\operatorname{argmax}} \log p(z_{im(j)}|T_{im(j)}, o_i, \hat{z}_{im(-j)}, \hat{\mu}^{(m)}, \hat{\Sigma}^{(m)}) \\
&= \underset{z_{im(j)}|z_{im(-j)}}{\operatorname{argmax}} \log p(T_{im(j)}|z_{im(j)}, o_i, \hat{z}_{im(-j)}, \hat{\mu}^{(m)}, \hat{\Sigma}^{(m)}) p(z_{im(j)}|\hat{z}_{im(-j)}, \hat{\mu}^{(m)}, \hat{\Sigma}^{(m)}) \\
&= \underset{z_{im(j)}|z_{im(-j)}}{\operatorname{argmax}} \log p(T_{im(j)}|z_{im(j)}, o_i) p(z_{im(j)}|\hat{z}_{im(-j)}, \hat{\mu}^{(m)}, \hat{\Sigma}^{(m)}) \\
&= \underset{z_{im(j)}|z_{im(-j)}}{\operatorname{argmax}} \log \left[[\exp(z_{im(j)}) o_i]^{T_{im(j)}} e^{-[\exp(z_{im(j)}) o_i]} \exp\left(-\frac{1}{2\sigma_{m(j)}} (z_{im(j)} - a_{im(j)})^2 \right) \right] \\
&= \underset{z_{im(j)}|z_{im(-j)}}{\operatorname{argmax}} T_{im(j)} \exp(z_{im(j)}) o_i - \exp(z_{im(j)}) o_i - \frac{1}{2\sigma_{m(j)}} (z_{im(j)} - a_{im(j)})^2
\end{aligned}$$

Differentiating we get,

$$\begin{aligned}
&\frac{\partial}{\partial z_{im(j)}} T_{im(j)} z_{im(j)} o_i - \exp(z_{im(j)}) o_i - \frac{1}{2\sigma_{m(j)}} (z_{im(j)} - a_{im(j)})^2 \\
&\qquad = T_{im(j)} o_i - \exp(z_{im(j)}) o_i - \frac{1}{\sigma_{m(j)}} (z_{im(j)} - a_{im(j)})
\end{aligned}$$

Because $z_{im(j)}$ appears as a linear and exponential term, we cannot solve this gradient analytically. We therefore utilize Newton-Raphson optimization. For this we also require the Hessian, which is given by,

$$\begin{aligned}
&\frac{\partial}{\partial z_{im(j)}} T_{im(j)} o_i - \exp(z_{im(j)}) o_i - \frac{1}{\sigma_{m(j)}} (z_{im(j)} - a_{im(j)}) \\
&\qquad = -\exp(z_{im(j)}) o_i - \frac{1}{\sigma_{m(j)}} < 0
\end{aligned}$$

Notice the Hessian is always negative-definite, which implies each update has a single, unique optimum.

In practice, the Newton-Raphson updates can be performed in vectorized fashion iteratively for each column of z. We generally find that this optimization takes 5-15 iterations (full passes over all columns of z) and less than a minute to converge. We refer to the program that performs these calculations as $\hat{z}_m = \texttt{MAP\_z}\left(t, o, \hat{\mu}^{(m)}, \hat{\Sigma}^{(m)}\right)$.

3.2 Complete inference of $\mu^{(m)}$, $\Sigma^{(m)}$, and z_m

For complete inference we use the following iterative conditional modes algorithm [2]:

- Initialize $T_m = t_m(o \times \mathbf{1}_{1\times\text{markers}})$, $\hat{z}_m = \text{lag}(t_m)$.
- Until convergence of $\log p(T_m|o, \hat{z}_m, \hat{\mu}^{(m)}, \hat{\Sigma}^{(m)}) + \log p(\hat{z}_m|\hat{\mu}^{(m)}, \hat{\Sigma}^{(m)})$, iterate:
  - Update $\hat{\mu}^{(m)}$ and $\hat{\Sigma}^{(m)}$:

$$\begin{aligned}
\hat{\mu}^{(m)} &= \frac{1}{\#\text{samples}} \sum_i \hat{z}_{im} \\
\hat{\Sigma}^{(m)} &= \frac{1}{\#\text{samples} - 1} \sum_i (\hat{z}_{im} - \hat{\mu}^{(m)})^T (\hat{z}_{im} - \hat{\mu}^{(m)}) + \lambda \text{diag}\left[\text{cov}\left(\hat{z}_m^{(\text{init})}\right)\right]
\end{aligned}$$

  - Update $\hat{z}_m = \texttt{MAP\_z}\left(t, o, \hat{\mu}^{(m)}, \hat{\Sigma}^{(m)}\right)$.

Here diag(x) of the square matrix x returns an equivalently sized matrix with only the diagonal of x preserved and 0's for the off-diagonal terms. cov$(\cdot)$ denotes the usual sample covariance matrix.

Note that in this algorithm we have added a regularization to the estimate of the covariance matrix. This is done in order to ensure stability and to avoid infinite-data-likelihood singularities that arise from singular covariance matrices. This is most often happens when a genes TPM abundance is mostly zero (i.e. there is little data for the gene), giving the multivariate normal layer an opportunity to increase the data likelihood (via the determinant of the covariance matrix) by tightly coupling this genes latent abundance to that of another gene, thereby producing a singularity. This regularization is probabilistically equivalent to adding an Inverse-Wishart prior over $\Sigma^{(m)}$. The parameter λ controls the strength of the regularization. In practice, we find $\lambda = 0.1$ leads to good predictive performance, stable (non-singular) covariance matrices, and reasonably quick convergence.

4 Decoding

During decoding we are given new measured TPM measurements for our markers, $t^*_m \in \mathbb{R}^{\text{query samples}\times|m|}$, and we must make predictions about the expression of all transcriptional programs and the remaining non-marker genes. To do this we first need an estimate of the log-latent abundances $\hat{z}^*_m$ associated with t^*_m. Given the estimates $\hat{\mu}^{(m)}$ and $\hat{\Sigma}^{(m)}$ obtained from the training data, we obtain these estimates as

$$\hat{z}^*_m = \texttt{MAP\_z}\left(t^*_m, \mathbf{1}_{\text{query samples}\times 1}, \hat{\mu}^{(m)}, \hat{\Sigma}^{(m)}\right)$$

Given the inferred marker latent abundances, we let our estimates of s^* and t^*_m be the maximizers of their probability distribution. In other words, $\hat{s}^* = \text{argmax}_{s^*}\, p(s^*|\hat{z}^*_m)$ and $\hat{t}^*_m = \text{argmax}_{t^*_m}\, p(t^*_m|\hat{z}^*_m)$.

Our estimate for the expression of all transcriptional programs is given by

$$\underset{s^*}{\text{argmax}}\, p(s^*|\hat{z}^*_m) = \mathbb{E}[s^*|\hat{z}^*_m] = \mu_{s^*|\hat{z}^*_m} = \hat{\mu}_s + \left(\hat{z}^*_m - \hat{\mu}^{(m)}\right)\text{inv}\left(\hat{\Sigma}^{(m)}\right)\hat{\sigma}_{z_m,s}.$$

Here, $\hat{\mu}_s$ and $\hat{\sigma}_{z_m,s}$ represent estimates of the unconditional mean of s and the cross-covariance matrix between z_m and s previously learned during encoding.

Similarly, for the entire transcriptome we have,

$$\hat{t}^*_{ij} = \underset{t}{\text{argmax}}\, p(t|\hat{z}^*_{im}) = \exp\left(\mu_{z_{ij}|\hat{z}^*_{im}}\right).$$

where,

$$\mu_{z_{ij}|\hat{z}^*_{im}} = \hat{\mu}_j + \left(\hat{z}^*_{im} - \hat{\mu}^{(m)}\right)\text{inv}\left(\hat{\Sigma}^{(m)}\right)\hat{\sigma}_{z_m,z_j}$$

We could also use the expected value of t as our estimate.

$$\begin{aligned}
\mathbb{E}[t^*_{ij}|\hat{z}^*_{im}] &= \int_{-\infty}^{\infty} \mathbb{E}[t^*_{ij}|z^*_{ij}]p(z_{ij}|\hat{z}^*_{im})\mathrm{d}z_{ij} \\
&= \int_{-\infty}^{\infty} \exp(z_{ij})\mathcal{N}(z_{ij}|\mu_{z_{ij}|\hat{z}^*_{im}}, \Sigma_{z_{ij}|\hat{z}^*_{im}})\mathrm{d}z_{ij} \\
&= \mathbb{E}_{\mathcal{N}}[\exp(z_{ij})|\hat{z}^*_{im}]
\end{aligned}$$

The Moment Generating Function of a Normal random variable X with mean μ and variance σ^2 is given by $M(t) = \mathbb{E}[\exp(tX)] = \exp(\mu t + \sigma^2 t^2/2)$. Therefore we have,

$$\mathbb{E}[t^*_{ij}|\hat{z}^*_{im}] = \mathbb{E}_{\mathcal{N}}[\exp(z_{ij})|\hat{z}^*_{im}] = M(1) = \exp\left(\mu_{z_{ij}|\hat{z}^*_{im}} + \frac{1}{2}\Sigma_{z_{ij}|\hat{z}^*_{im}}\right)$$

where,

$$\begin{aligned}\mu_{z_{ij}|\hat{z}^*_{im}} &= \hat{\mu}_j + \left(\hat{z}^*_{im} - \hat{\mu}^{(m)}\right) \mathrm{inv}\left(\hat{\Sigma}^{(m)}\right) \hat{\sigma}_{z_m,z_j} \\ \Sigma_{z_{ij}|\hat{z}^*_{im}} &= \hat{\sigma}_{jj} - \hat{\sigma}^T_{z_m,z_j} \mathrm{inv}\left(\hat{\Sigma}^{(m)}\right) \hat{\sigma}_{z_m,z_j}\end{aligned}$$

Here, $\hat{\mu}_j$ and $\hat{\sigma}_{z_m,z_j}$ represent estimates of the unconditional mean of z_j and the cross-covariance matrix between z_m and z_j. These were learned from the training data during encoding.

Though this predictor is unbiased, it does not produce a good prediction for most samples. This is due to the right-skew of the Poisson, which drags its mean away from the most likely values.

5 References

[1] Surojit Biswas. The latent logarithm. *arXiv*, pages 1–11, 2016.

[2] Julian Besag. On the Statistical Analysis of Dirty Pictures. *Journal of the Royal Statistical Society*, 48(3):259–302, 1986.