

# Tradict - mathematical details

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

## 3 Contents

4	1 Preliminaries	1
5	2 Model	1
6	3 Encoding	2
7	3.1 Inference of $z_m$ given $\mu^{(m)}$ and $\Sigma^{(m)}$	3
8	3.2 Complete inference of $\mu^{(m)}$ , $\Sigma^{(m)}$ , and $z_m$	4
9	4 Decoding	5
10	5 References	6

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.

## 15 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  $q$ .

## 18 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  $C_\lambda$  is a normalization constant [REF]. The mean of this distribution is given by  $\lambda$ , 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

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

$$z \sim \mathcal{N}(\mu, \Sigma)$$

$$T_j \sim \text{Continuous-Poisson}(\exp(z_j)o)$$

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

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

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

$$z_m \sim \mathcal{N}(\mu^{(m)}, \Sigma^{(m)})$$

$$z_{-m}|z_m \sim \mathcal{N}(\mu_{z_{-m}|z_m}, \Sigma_{z_{-m}|z_m})$$

$$T_j \sim \text{Continuous-Poisson}(\exp(z_j)o)$$

45 Here,  $\mu^{(m)}$  and  $\Sigma^{(m)}$  refer to mean vector and covariance matrix of  $z_m$ . Given these, the conditional  
 46 mean of the log-latent abundances for all non-marker genes can be obtained through Gaussian conditioning.  
 47 Specifically, for two normally distributed row-vector variables  $a$  and  $b$  the conditional mean of  $b$  given  $a$  is  
 48 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  $\sigma_{ab}$  is the cross-covariance between  
 49  $a$  and  $b$ , and  $\Sigma_a$  and  $\Sigma_b$  are the covariance matrices of  $a$  and  $b$ , respectively.

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

$$z_m \sim \mathcal{N}(\mu^{(m)}, \Sigma^{(m)})$$

$$s|z_m \sim \mathcal{N}(\mu_{s|z_m}, \Sigma_{s|z_m})$$

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

### 57 3 Encoding

58 As described in the Supplemental Information, given an estimate of  $z_m$ ,  $\hat{z}_m$ , inference of  $\mu_s$ ,  $\mu_{z_{-m}}$ ,  $\sigma_{z_m, s}$  and  
 59  $\sigma_{z_m, z_{-m}}$  is straightforward. In lag transforming the entire training TPM expression matrix,  $t \in \mathbb{R}^{\text{samples} \times \text{genes}}$ ,  
 60 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  
 61 mean of  $\hat{z}_{-m}$ .

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

65 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  
 66  $\hat{z}_m$  and  $\hat{s}$  and between  $\hat{z}_m$  and  $\hat{z}_{-m}$ , respectively.

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

### 75 3.1 Inference of $z_m$ given $\mu^{(m)}$ and $\Sigma^{(m)}$

76 Suppose Tradict has estimates of  $\mu^{(m)}$  and  $\Sigma^{(m)}$  given by  $\hat{\mu}^{(m)}$  and  $\hat{\Sigma}^{(m)}$ , and let  $T_m = t_m(o \times \mathbf{1}_{1 \times \text{markers}})$   
 77 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  
 78 of sample sequencing depths in millions of reads. Given these, we would like to calculate the maximum *a*  
 79 *posteriori* (MAP) estimate of  $\hat{z}_m = \text{argmax}_{z_m} p(z_m | o, T_m, \hat{\mu}^{(m)}, \hat{\Sigma}^{(m)})$ .

80 The posterior distribution over  $z_m$  is given by

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

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

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

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

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

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

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

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

89 Differentiating we get,

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

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

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

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

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

### 97 3.2 Complete inference of $\mu^{(m)}$ , $\Sigma^{(m)}$ , and $z_m$

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

- 99 • Initialize  $T_m = t_m(o \times \mathbf{1}_{1 \times \text{markers}})$ ,  $\hat{z}_m = \text{lag}(t_m)$ .  
100 • 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:  
101 – Update  $\hat{\mu}^{(m)}$  and  $\hat{\Sigma}^{(m)}$ :

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

- 102 – Update  $\hat{z}_m = \text{MAP\_Z}(t, o, \hat{\mu}^{(m)}, \hat{\Sigma}^{(m)})$ .

Here  $\text{diag}(x)$  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.  $\text{cov}(\cdot)$  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  $\lambda$  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

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

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

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

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,

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

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.

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

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

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

129 where,

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

130 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  
131 between  $z_m$  and  $z_j$ . These were learned from the training data during encoding.

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

## 134 5 References

135 [1] Surojit Biswas. The latent logarithm. *arXiv*, pages 1–11, 2016.

136 [2] Julian Besag. On the Statistical Analysis of Dirty Pictures. *Journal of the Royal Statistical Society*,  
137 48(3):259–302, 1986.