AMSI SUMMERRESEARCH SCHOLARSHIPS 2023–24

SET YOUR SIGHTS ON RESEARCH THIS SUMMER

Cell type deconvolution methods for Spatial Transcriptomics (ST) data

Yichen Jiang

Supervised by Dr Heejung Shim

The University of Melbourne

February 2024

AMSI AUSTRALIAN MATHEMATICAL SCIENCES INSTITUTE

Abstract

The advent of Spatial Transcriptomics (ST) technology enables the measurement of gene or isoform expressions at pixel resolution, where each pixel contains multiple cells. Hence, the identification of cell types within the tissue's spatial organization is hindered. Cell type deconvolution aims to estimate the proportion of different cell types for each pixel from the ST data. It is an important step for downstream analysis, such as the interpretation of identified transcript programs from spatial data in cell type context. Existing methods include the reference-based approach, which uses potentially imperfect cell-type signature genes from single cell RNA sequencing (scRNA-seq) as assistance; or the reference-free approach, which only uses the ST data for cell-type proportion estimation. This research project aims to utilise the cell-type signature genes as prior in the deconvolution process, thus giving ST data power the power to correct it if the scRNA-seq data is inaccurate. This approach demonstrates greater accuracy in deconvolved cell types when an appropriate prior is given, with the performance approaching the reference-free result as the prior became less suitable.

1 Introduction

Biological tissues often contain a mixture of different cell types, the spatial structure of which can be used to identify diseases. The advancement of short-read and long-read sequencing technologies allows the capturing of gene and isoform expression on various pixels of a tissue sample (Figure 1), hence preserving the spatial context of gene activity. However, the fact that for each pixel, spatial data comes from multiple cells makes the discovery of cellular structure difficult.

Cell type deconvolution endeavours to understand compositions of cell types within tissue samples, and multiple methods have been developed in this area. Reference-based methods use cell type signature gene information obtained from scRNA-seq to help distinguish cell types within pixels. Two examples of this approach are as follows: Robust Cell Type Decomposition (RCTD) [2] uses mean signature gene expression profiles to construct a poisson model while SPOTlight [3] initializes a *Non-Negative Matrix Factorisation* using single cell reference and *Non-Negative Least Squares* to build a pixel-wise cell type proportion profile. The alternative approach is reference-free, which uses the spatial data directly without the single cell reference. An example of such method is STdeconvolve [1], which models the dataset using Latent Dirichlet Allocation (LDA) [4].

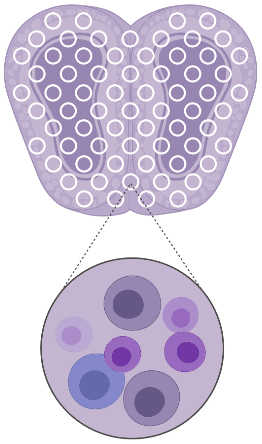


Figure 1: Spatial Transcriptomics data visualisation from [1].

However, using single cell reference data has two drawbacks, namely its potential inaccuracy and unavailability. The true reference dataset may vary between different tissues or organisms, and is often limited for diseased tissues. The goal of this research project is to tackle this issue by incorporating the single cell signature gene dataset as a prior, and thus the deconvolution process does not rely completely on the reference.

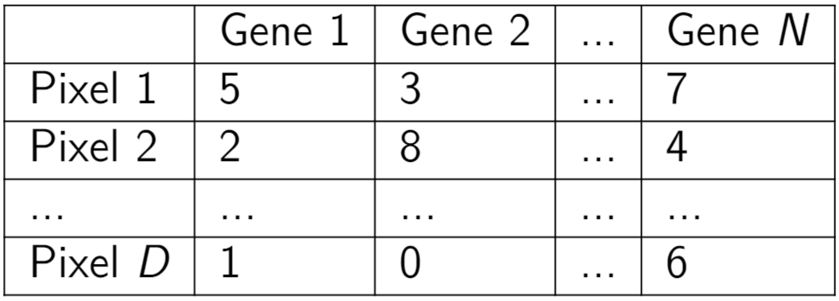
Statement of Authorship. This research project was completed under the supervision of Dr Heejung Shim. All mathematical derivations, program implementations and analysis were done with consistent guidance from my supervisor throughout the duration of this project.

2 Datasets

As described previously, two input datasets are required for deconvolution: the Spatial Transcriptomic data which captures pixel resolution gene activity, and the single cell reference data which contains cell-type specific gene expression profiles.

2.1 Spatial Transcriptomic Data

A count matrix (Figure 2) of size $D \times N$, entry with row d, column n represents the count of mRNA copies corresponding to gene n in pixel d. It will be referred to as 'spatial data' in short.

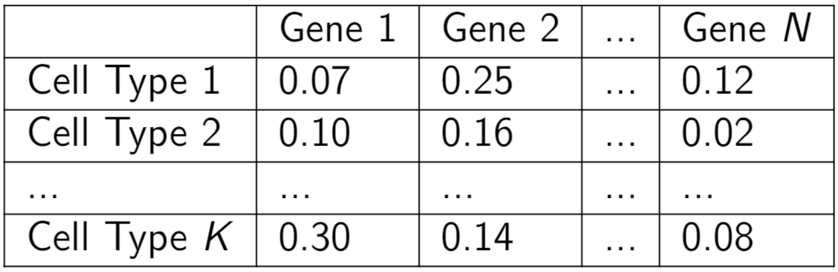


	Gene 1	Gene 2	...	Gene N
Pixel 1	5	3	...	7
Pixel 2	2	8	...	4
...	...	...	...	...
Pixel D	1	0	...	6

Figure 2: Gene expression profile for each pixel.

2.2 Single Cell Reference Data

A matrix (Figure 3) of size $K \times N$, entry with row k, column n represents the likelihood of occurrence of gene n in cell type k. Each row of this matrix should sum to 1 because every entry represents a proportion.



	Gene 1	Gene 2	...	Gene N
Cell Type 1	0.07	0.25	...	0.12
Cell Type 2	0.10	0.16	...	0.02
...	...	...	...	...
Cell Type K	0.30	0.14	...	0.08

Figure 3: Gene expression profile from scRNA-seq.

Note that this dataset will be used as prior in the inference step and later referred to as η.

3 Methods

The modelling component of this project is based on the *Latent Dirichlet Allocation* (LDA) from [1] with modifications. LDA was first proposed in [4], under the context of Natural Language Processing (NLP). We will give a brief description of LDA in the original setting and focus on how it is applied and modified in cell type deconvoluion.

Using notations from the original LDA paper, a *word* w represents the finest grain of data which comes from a set of distinct words called the *vocabulary*, with index $\{1, 2, ..., V\}$. Furthermore, each *word* w_n is assumed to be generated from one single *topic* z_n, which is an unobserved latent variable. Lastly, each *document* $\mathbf{w}$ consists of a sequence of *words* $(w_1, w_2, ..., w_n)$ and a corpus $\mathbf{M}$ consists of a collection of documents $(\mathbf{w}_1, \mathbf{w}_2, ..., \mathbf{w}_m)$. Note that order of words and documents are irrelevant, meaning they can be exchanged without affecting the inference result.

Two key challenges that arise naturally are: what is the topic distribution per document? And how likely is each word to occur in a particular topic z (i.e, can $p(w_i|z_j)$ be recovered)? LDA and many other algorithms such as the Probabilistic Latent Sematic Analysis [5] aim to address these questions.

3.1 Latent Dirichlet Allocation in Cell Type Deconvolution

Notations stated below will be used throughout this report:

- D represents the number of pixels in the spatial dataset.
- K represents the number of cell types in the reference dataset.
- N represents the number of genes in both datasets.

LDA is a generative statsitical model, with the following generative process given in [1] when applied to cell type deconvolution.
For each pixel $d \in [1 : D]$:

1. Generate $\theta_d \sim Dir(\alpha)$ (θ_d is K dimensional)
2. For each molecule $m \in [1 : M_d]$ from pixel d:
   (a) Given θ_d, assign cell type $z_{d,m} \sim Categorical\left(\theta_d\right)$
   (b) Given $z_{d,m}$, assign gene $w_{d,m} \sim Categorical\left(\beta_{z_{d,m}}\right)$ [1]

Where M_d represents the total gene count per pixel. The modification proposed in this paper in comparison to the LDA model described originally in [1] is that we fit a prior for β from the single cell reference dataset, instead of treating it as a parameter to be estimated. Let η denote the single cell reference matrix in section

[1] Note that β represents the true underlying cell type gene proportions.

2.2, where η_{kn} represents the proportion of gene n in cell type k. Then, prior for β is:

$$p(\beta_k) = Dir\left(b\eta_k\right), \text{ for } k = 1, 2, ..., K$$

where η_k represents the kth row of the single cell reference matrix. b is a hyperparameter which controls the strength of prior in the inference step, it will be tuned as part of the algorithm. Note that for dirichlet distribution, scaling up the parameter does not change the overall mean, but decreases variance. Hence, if η is strongly correlated with β, then a high value of b would be suitable. However, if the cell type specific gene expression profile lacks accuracy, then a low value of b would be adequate.

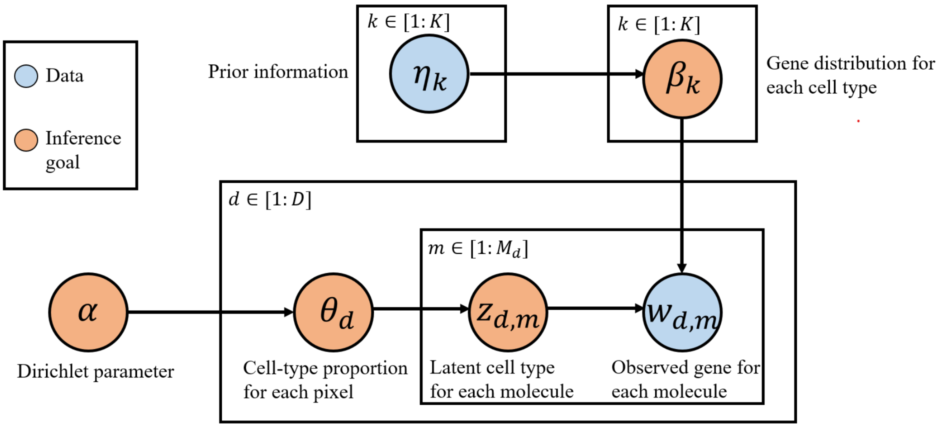


Figure 4: Generative process of the LDA model [4] applied in Cell Type Deconvolution with prior for β incorporated, each box represents replications.

4 Variational Inference

Variational Inference (VI) is used to obtain posterior distributions for the inference goals in Figure 4. A general introduction to this method will be given below.

In Bayesian statistics, we are often interested in the posterior distribution $p(\theta|x)$, where θ is the unknown parameters and x is the data.

$$p(\theta|x) = \frac{p(\theta)p(x|\theta)}{p(x)} = \frac{p(\theta)p(x|\theta)}{\int_\Theta p(\theta)p(x|\theta)d\theta}$$

However, with θ being high dimensional in many cases, the integral $\int_\Theta p(\theta)p(x|\theta)d\theta$ cannot be evaluated and thus the posterior is intractable (or equivalently, it is only known up to a constant). *Variational Inference* approximates this posterior distribution through an optimisation approach and is well known for its speed comparing to alternative methods such as *Markov Chain Monte Carlo* (MCMC).

4.1 Evidence Lower Bound

VI approximates the target posterior $p(\theta|x)$ by some member $q(\theta)$ from a given family of distributions Q. The best member $q^*(\theta)$ is found by minimizing the Kullback-Leiber(KL) Divergence between the variational distribution and the target posterior:

$$q^*(\theta) = \arg\min_{q(\theta)\in Q} KL\left(q(\theta)||p(\theta|x)\right) = \arg\min_{q(\theta)\in Q} \int_\Theta q(\theta)\log\frac{q(\theta)}{p(\theta|x)}\,d\theta$$

However, because $p(\theta|x)$ exists in the KL Divergence argument, it cannot be computed directly. This issue is resolved by first defining the Evidence Lower Bound function which can be computed.

$$\text{ELBO}(q(\theta)) = \int_\Theta q(\theta)\log\frac{p(\theta,x)}{q(\theta)}\,d\theta$$

Then the KL Divergence can be rewritten as follows,

$$\text{KL}\left(q(\theta)||p(\theta|x)\right) = -\text{ELBO}(q(\theta)) + \log p(x)$$

Because VI aims to find the best $q(\theta)$, the $\log p(x)$ term is irrelevant and thus,

$$q^*(\theta) = \arg\min_{q(\theta)\in Q} \text{KL}\left(q(\theta)||p(\theta|x)\right) = \arg\max_{q(\theta)\in Q} \text{ELBO}(q(\theta))$$

A quick note that KL Divergence is strictly non-negative (Appendix A.1), we have,

$$\begin{aligned}\text{KL}\left(q(\theta)||p(\theta|x)\right) &\geq 0\\ -\text{ELBO}(q(\theta)) + \log p(x) &\geq 0\\ \text{ELBO}(q(\theta)) &\leq \log p(x)\end{aligned}$$

and hence the name 'Evidence Lower Bound'.

4.2 Mean Field Variational Family

A famous and widely used variational family is the *Mean Field Variational Family*, which assumes independence between variational parameters.

$$q(\theta) = \prod_{j=1}^{J} q_j(\theta_j)$$

The benefit of independence is that the integral $\int_\Theta q(\theta)\log\frac{p(\theta,x)}{q(\theta)}\,d\theta$ can be factorized for each θ_j, thus allowing the integrals to be computed independently for each variational parameter. Let w represent the count matrix containing spatial data (Figure 2), and thus the target posterior is $p(\theta, z, \beta|w, \alpha)$[2]. Using the mean field

[2] Note that α is treated as a parameter to be updated in each iteration, thus no variational distribution is assigned.

variational family yields the following variational distributions,

$$q(\theta, z, \beta) = \prod_{d=1}^{D} q(\theta_d) \times \prod_{d=1}^{D} \prod_{m=1}^{M_d} q(z_{d,m}) \times \prod_{k=1}^{K} q(\beta_k)$$

where M_d represents the total number of molecule counts in pixel d. Furthermore, the following variational distributions were chosen for each parameter[6].

- $q(\theta_d) = Dir(\gamma_d)$
- $q(z_{d,m}) = Categorical(\phi_{d,m})$
- $q(\beta_k) = Dir(\tau_k)$

With the variational distributions specified, the ELBO function can be expressed as follows,

$$\begin{aligned} \text{ELBO}(\gamma, \phi, \tau, \alpha | w, \eta, b) = & \mathbb{E}_q [\log p(\beta)] + \mathbb{E}_q [\log p(\theta|\alpha)] + \mathbb{E}_q [\log p(z|\theta)] + \mathbb{E}_q [\log p(w|z, \beta)] \\ & - \mathbb{E}_q [\log q(\theta)] - \mathbb{E}_q [\log q(z)] - \mathbb{E}_q [\log q(\beta)] \end{aligned}$$

Each expectation term is taken with respect to the underlying variational distribution, detail derivations will be given in the appendix (Appendix A.2). Parameter updates for γ, ϕ and τ are obtained by setting the partial derivative of the ELBO function with respect to these parameters to 0 (e.g, setting $\frac{\partial \text{ELBO}}{\partial \gamma_{dk}} = 0$ for pixel d and cell type k). However, for α, an straight forward update formula cannot be obtained in the same manner, instead, Newton-Raphson method is used in each iteration to find the optimal α (Appendix A.3).

In terms of model specifications, cell type proportion for each molecule is considered separately. Nevertheless, in practice, estimated cell type proportions for molecules within the same pixel that have the same gene will have no difference because their corresponding spatial and reference data are identical. Hence, the space complexity can be greatly reduced in when constructing the 3D matrix ϕ, where matrix size goes from $\sum_{d=1}^{D} M_d \times K$ to $D \times N \times K$.

Note that in the pseudocode, ψ is the digamma function, w_{dn} represents the count of gene n in pixel d, matrices γ and τ are normalized because they are parameters for dirichlet distributions where proportions are obtained from (cell type proportion per pixel and gene proportion per cell type respectively).

4.3 Interpretations for parameter update

Interpretations for updates of the three main parameters ϕ, γ and τ are examined because they enhance credibility of the LDA model, particularly for parameter τ, which is appended from the original deconvolution model. Because γ and τ are both parameters for dirichlet distributions, they are discussed together.

Interpretations for updates of γ and τ. In the update formulas for both γ and τ, involvements from both the prior and spatial data can be easily spotted. For τ_{kn}, the first term $b\eta_{kn}$ is the prior parameter (reason why b controls power of the prior), while the second term $\sum_{d=1}^{D} \phi_{dnk} \cdot w_{dn}$ represents the effective count of cell

Algorithm 1 Variational Inference Parameter Update for the LDA model

```
Input: $w$, $b$, $\eta$, $D$, $K$, $N$, $\epsilon$
Initialize $\alpha_i = 50/K$ for all $i$
Initialize $\gamma_{dk} = 1$ for all $d$, $k$
Initialize $\phi_{dnk} = 1/K$ for all $d$, $n$(gene), $k$
Initialize $\tau_{kn} = 1$ for all $k$, $n$
while $\Delta$ELBO $> \epsilon$ do
    Update $\phi_{dnk} \propto \exp\left\{\psi(\gamma_{dk}) - \psi\left(\sum_{j=1}^{K}\gamma_{dj}\right) + \psi(\tau_{kn}) - \psi\left(\sum_{j=1}^{N}\tau_{kj}\right)\right\}$
    Update $\gamma_{dk} = \alpha_k + \sum_{n=1}^{N}\phi_{dnk}\cdot w_{dn}$
    Update $\tau_{kn} = b\eta_{kn} + \sum_{d=1}^{D}\phi_{dnk}\cdot w_{dn}$
    Update $\alpha$ using Newton-Raphson Method.
    Compute and store current ELBO.
end while
return $\alpha$, normalized $\gamma$, $\phi$, normalized $\tau$
```

type k assigned to molecules with gene n across all pixels(Figure 5). For γ_{dk}, contributions from both prior and data can be identified. Although α_k is not a specified by the user, it can be considered as the overall proportion of cell type k across all pixels, so it plays the role of a prior. On the other hand, $\sum_{n=1}^{N}\phi_{dnk}\cdot w_{dn}$ represents the effective count of cell type k assigned to pixel d across all molecules in that pixel.

Interpretations for update of ϕ. Firstly, the update formula can be rewritten,

$$\begin{aligned}\phi_{dnk} &\propto \exp\left\{\mathbb{E}_q\left[\log\theta_{dk}\right] + \mathbb{E}_q\left[\log\beta_{kn}\right]\right\}\\ &\propto \exp\left\{\mathbb{E}_q\left[\log\theta_{dk}\right]\right\}\times\exp\left\{\mathbb{E}_q\left[\log\beta_{kn}\right]\right\}\end{aligned}$$

Although expectation and exponential cannot be swapped, this formula can still be interpreted as $\theta_{dk}\times\beta_{kn}$ - relative proportion of cell type k in pixel d multiplied by the likelihood for gene n to appear in cell type k gives the proportion of cell type k for all molecules with gene n in pixel d. Hence, instead of updating directly from the spatial data, convergence result of ϕ relies on the correctness of both τ and β.

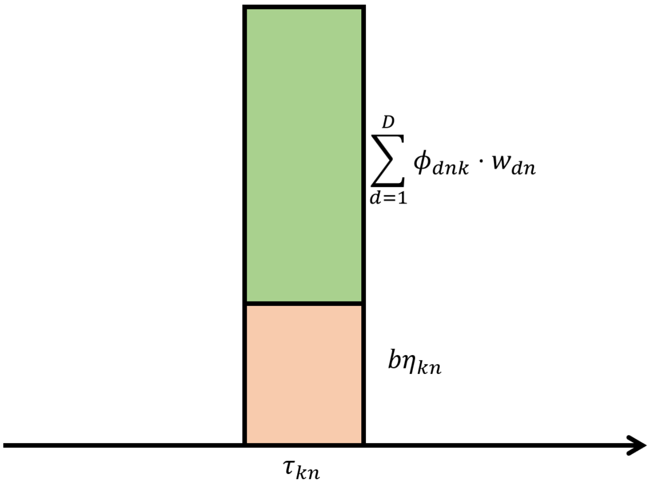


Figure 5: Visual understanding of the update formula for τ_{kn}

5 Results

Performance of our proposed model is tested against both simulated and test dataset. We first confirm the validity of our model on data with strong signal, accompanied by an informative prior; then we assess the benefit of having informative prior when the data signal is weak. We also evaluate how our model performs when prior is less informative, with the expectation that it should approach the reference-free method performance. Finally, we show that our model can also be applied to real dataset.

5.1 Simulated Datasets

For simulation, we started by generating two sets of cell type specific gene expression profiles (η), each having $K = 3$ cell types and $N = 150$ genes. Cell types from strong signal data is characterized by a distinct group of genes of size 50. While for the weak signal data (Figure 7), differences of gene expressions between cell types is minor, in particular, cell type 2 and 3 only differ by the middle 10 genes.

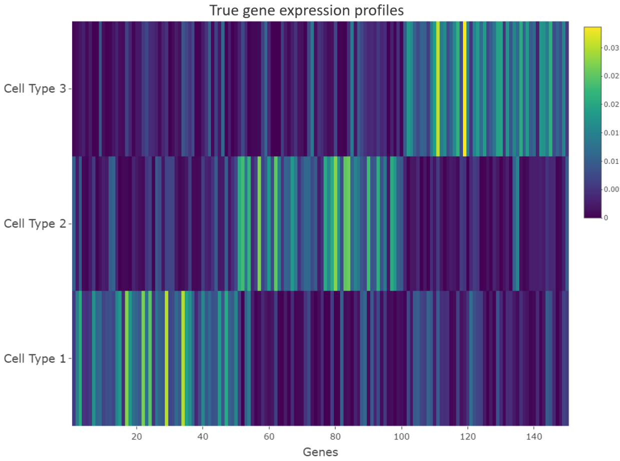


Figure 6: Strong signal from gene expression data for each cell type.

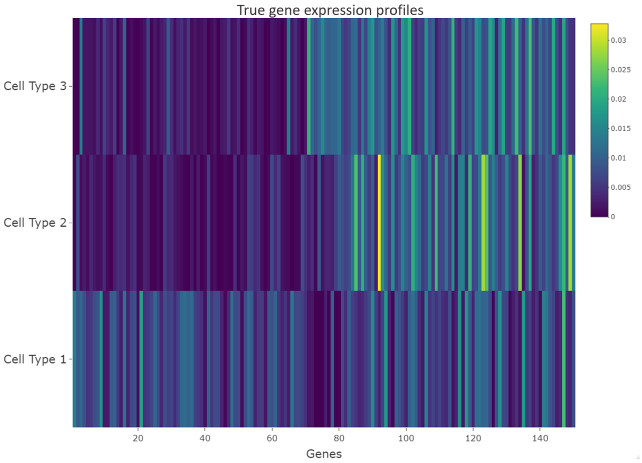


Figure 7: Weak signal from gene expression data for each cell type.

With the prior η being generated, the ground true β is then generated with $\beta_k \sim Dir(\eta_k)$, thus we obtain an informative prior as η is closely correlated with β (check Appendix B.1 for their correlation plots). Two sets of spatial data are then generated with the two different β, we fixed number of pixels $D = 196$, each pixel contains number of molecules $M_d \sim \text{Poisson}(2000)$. When generating pixel-specific cell-type proportion $\theta_d \sim Dir(\alpha)$, we used $\alpha = (0.4, 0.35, 0.25)$ for cell types 1, 2 and 3 respectively. The entire simulated dataset follow the generative process specified by LDA (Figure 4).

Prior constants. For each set of data (spatial and reference), we ran the VI algorithm with prior constant $b = (1, 20, 100, 500, 750, 1000, 1500)$ and compare performances for each VI run. It is expected that with a very low prior constant, the performance may be similar to the reference-free method, because the prior contribution is small in the update formulas, parameter convergence will largely depend on the data. However, drop in performance is also expected if the prior constant is set to be too large (prior too powerful), because even though the prior is correlated with ground truth β, it is not perfect and we still want the data to be involved in identifying the correct posterior.

Result visualisation. In the generative process of the LDA model, no assumptions are being made on the spatial location of pixels. Hence, deconvolution results will be identical if positions for pixels are randomly reassigned. We begin analysis by randomly selecting 10 pixels, and give visual comparison of deconvolved cell types proportions for every VI run with different prior constants, against the true cell type proportions and the reference-free (STdeconvolve) result (Figure 8). The visualisation tool is based on the STdeconvolve package[1].

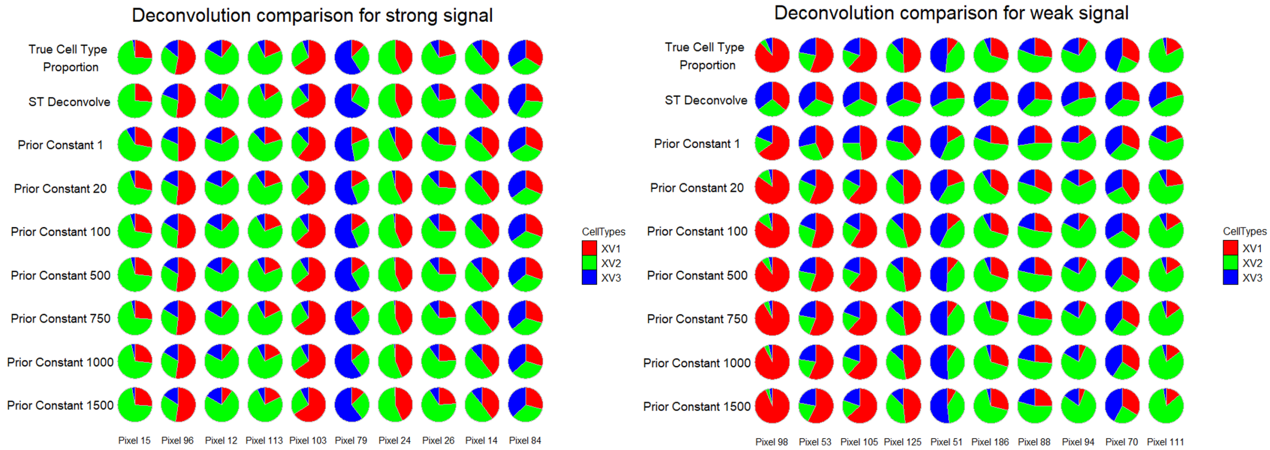


Figure 8: Deconvolved Cell Type proportions from 10 randomly selected pixels for different prior constants.

From figure 8, when the data signal is strong (left), the reference-free method is able to correctly identify proportions of each cell type, hence incorporating the prior into the model brings little improvements. Nevertheless, we do see that when the prior constant is low, our model may potentially misidentify cell types with low proportions (e.g. pixel 15, 24). This behavior is gradually corrected as we increase the prior constants. However, when the data signal is weak, STdeconvolve fails to identify correct cell type proportions, and it instead predicts approximate uniform distribution for all pixels. When the prior is included, improvements are evident as it no longer predicts uniform distributions. In the case when a low prior constant ($b = 1$) is used, the model still tends to overestimate cell type proportions that are low (e.g. pixel 98). This inaccuracy is significantly corrected as prior constants are increased. Overall, we observe that as we increase prior constants, performance changes positively from one similar to STdeconvolve to one almost identical to the ground truth, which matches our initial expectation.

Numerical comparisons. Instead of sampling 10 pixels at random, now we aim to give an overall numerical comparison of model performances over all pixels. For each VI run (with different prior constants), we compute the Root Mean Squared Error(RMSE) for each pixel d:

$$\text{RMSE}(\theta_d, \hat{\theta}_d) = \sqrt{\frac{\sum_{k=1}^{K} \left(\theta_{dk} - \hat{\theta}_{dk}\right)^2}{K}}$$

where θ_d represents ground true proportion for each cell type in pixel d and $\hat{\theta}_d$ represents estimated proportion for each cell in pixel d. Note that $\hat{\theta}$ is obtained after normalizing variational parameter γ.

With the metric defined, we used boxplot to visualise the distribution of RMSE for each VI run, which illustrates how performance varies with different data signal and prior constants (Figure 9). The conclusion we can draw from this diagram is similar to the previous visualisation that, after incorporating the prior, reduction in error is much more significant when deconvolving weak signal data. We also confirmed our expectation in this diagram that as the prior constant becomes closer to 0 (i.e, prior power decreasing), performance approaches the reference-free method result (two left-most boxplots). Finally, although it is not obvious in the graph, the optimal prior constant that gives lowest RMSE is 500 instead of 1500. A slight increase in the error can be observed as the prior constant becomes greater than 500. We again expects this behavior because prior is not perfect both in simulation and in practice.

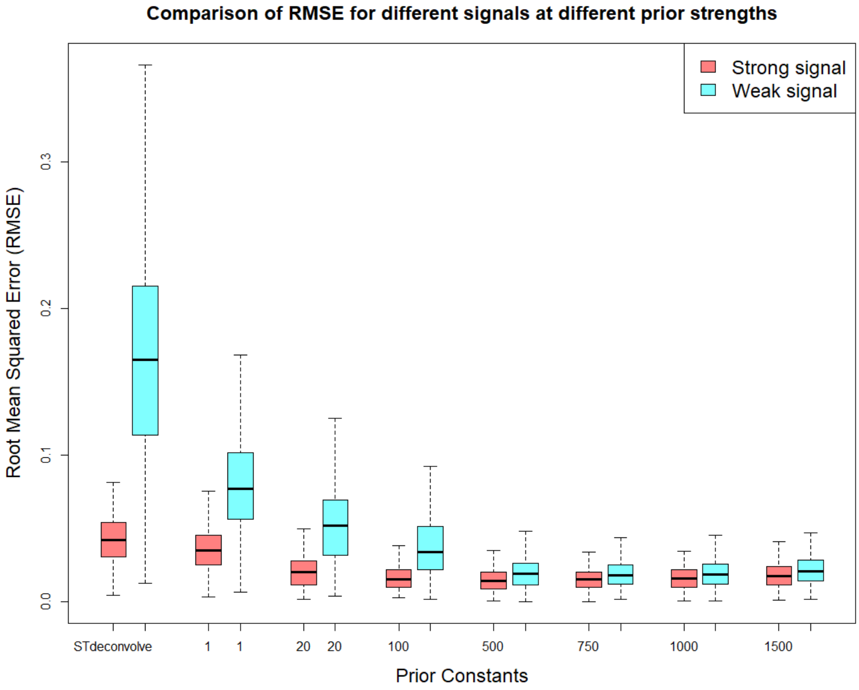
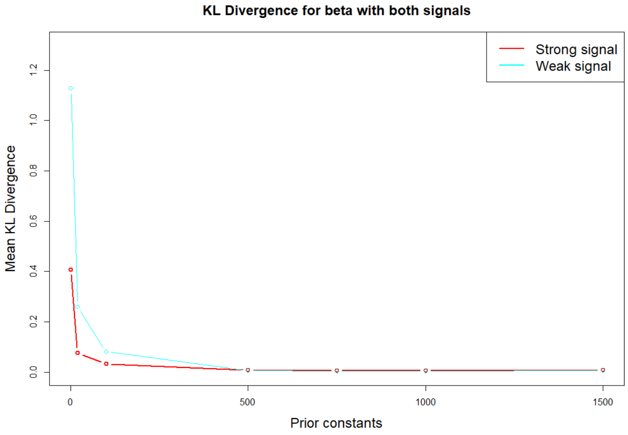


Figure 9: Performance comparison for both signals at various prior strength



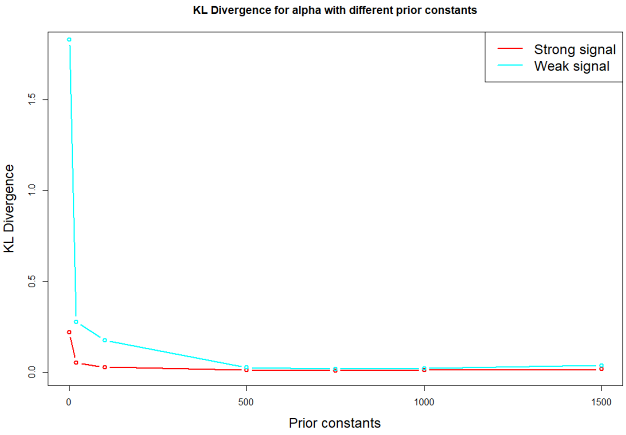


Figure 10: Mean KL Divergence between estimated and true β for both signals.

Figure 11: KL Divergence between estimated and true α for both signals.

Similar results for other parameters. As a final confirmation, we plotted KL divergences between estimated β, α to their respective ground truths (Figures 9, 10). It is apparent that previous conclusion can be applied to other parameters that for weak signal spatial data, it is more difficult for LDA to converge to the correct posterior, hence significant improvements can be expected from having a suitable prior.

Effects of changing prior informativeness. From the simulated spatial dataset with weak signal, we investigate effects of changing suitability of the prior by the following method:

1. Fix prior constant to be 20.
2. Mask different proportions of reference data matrix η.
3. Compute RMSE of VI result.

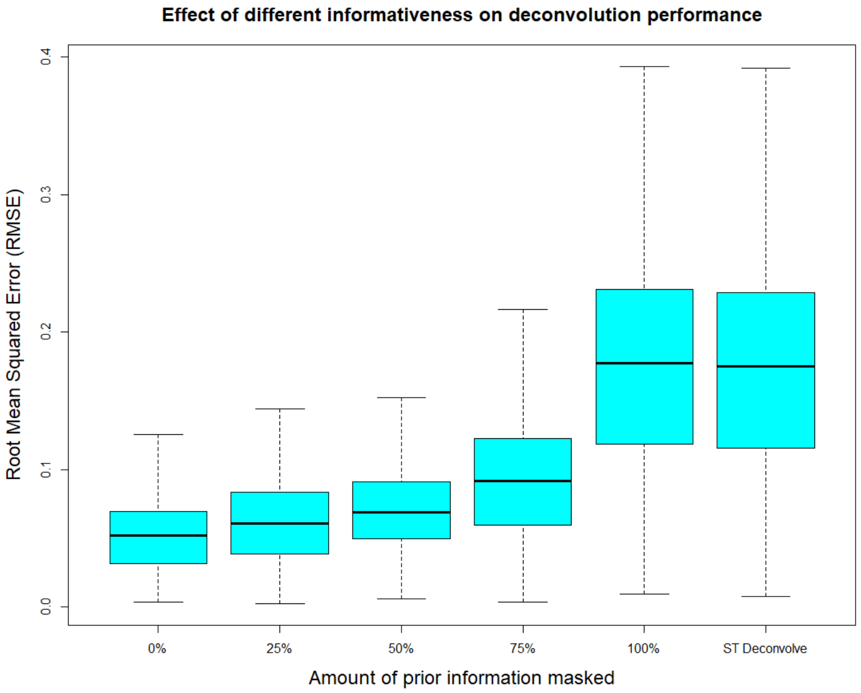


Figure 12: Performance comparison for different proportion of reference data masked away.

Figure 12 shows that as greater amount of prior information are hidden, which represents the prior becoming less informative, deconvolution performance decreases and approaches the reference-free performance. When all the information are hidden, the prior becomes uniform, the performance is almost identical to that of STdeconvolve. This diagram suggests that there is still benefit in utilizing the single cell reference, even if it lacks accuracy, the prior constant can always be tuned to find the optimal deconvolved proportions.

5.2 Biological Dataset

In this section we test our model performance when applied to real biological dataset. We selected the Visium Hippocampus ST dataset provided within the RCTD package [2] (the reference dataset was also provided), which was originally presented in [7]. For this dataset, there are $D = 313$ pixels, $N = 307$ genes and $K = 17$ cell types identified in RCTD. We still use RMSE as the comparison metric, employing the RCTD result as ground truth; performances are compared both visually and numerically between different prior constants, including the reference-free method.

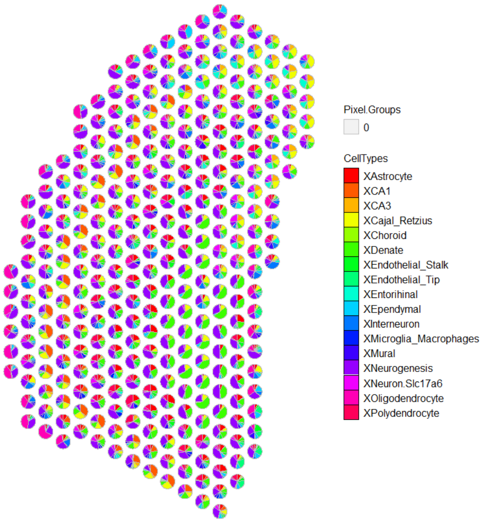


Figure 13: Deconvolution result by RCTD on Visium dataset.

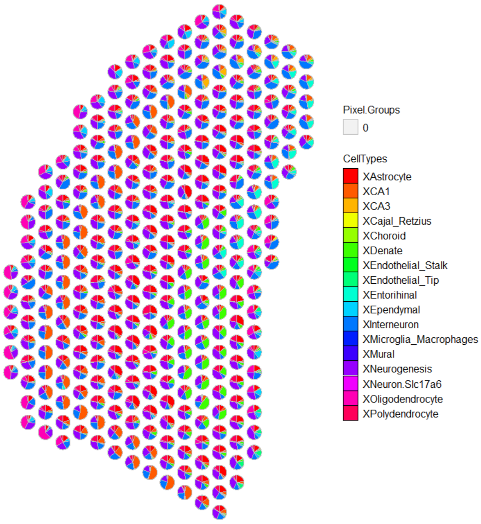


Figure 14: Deconvolution result by modified LDA (our model) on Visium dataset.

In Figures 13 and 14, both methods were able to identify structures that exist in the spatial data of the Hippocampus dataset. RCTD is more likely to assign pixels to a wider variety of cell types, allowing some to have low proportions but significantly different from 0. Whereas for our model, most of the pixels are dominated by 3 to 4 major cell types. To be more specific, when comparing major cell types that occur in both diagrams, both methods recognize cell types Neurogenesis, Oligodendrocyte, Denate and CA3 in the same locations. However, our method appears to have identified cell type Interneuron (blue) for Cajal Retzius (yellow) found in RCTD. Moreover, as a general observation, if a cell type is rare across all pixels (Endothelial Stalk, Microglia Macrophages etc), the LDA model tends to predict their non-existence (Appendix B.2). This behavior of over predicting major cell types and neglecting minor ones can be explained by the fact that Variational Inference inclines to underestimate variance. The KL Divergence can be rewritten as expectation with respect to q, thus enforcing q to match the peak of p and place less emphasis on regions where p is lowly expressed - cell types with low overall proportions.

In Figure 15, we used RCTD's deconvolution result as ground truth and visualized RMSE distribution for different VI runs with prior constants $b = (100, 500, 2500, 5000)$. The magnitude of prior constants tested is increased to improve efficiency of running VI algorithm, as every matrix is enlarged. We again observed an optimal prior constant $b = 500$ which achieves minimal RMSE, similar to the simulation result. However, the difference is, we see a sharp increase in RMSE as prior constant is set to $b = 5000$, signalling too much power is given to the prior for its level of accuracy.

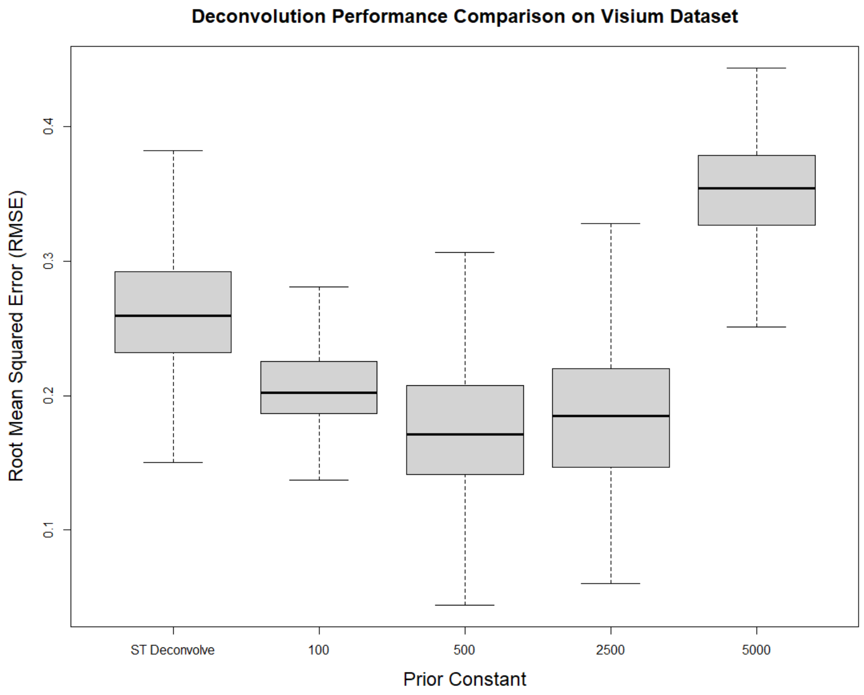


Figure 15: RMSE comparison for VI runs with different prior constant including reference-free result.

6 Discussion

In this section, we will address potential improvements that can be made to this model.

Optimal value of b. In simulated data, we have full knowledge of the ground truth and it can be exploited to obtain the optimal prior constant b. However, it is not the case for real biological dataset, where the accuracy of the reference is not known exactly. We hypothesize that this optimal value of b is highly relevant to sizes of the spatial data and the reference, but more investigation is required to make a conclusion.

Spatial data with isoform-resolution. This research project is entirely based on ST dataset with gene-level information for each pixel. With the advancement of long-read sequencing technology, ST data with isoform-resolution is more and more popular. It would not be excessively challenging to extend our model to allow ST data with isoform-resolution, and the expected improvements are on the estimation of ϕ (cell type proportions for each molecule).

More prior scenarios. So far we only tested the outcome of masking away certain proportions of informative prior. However, it is possible that in a reference data, some gene expression profiles are suitable for the tissue sample while others are not. Hence, instead of simply replacing by uniform values, another way of modifying the prior would be replacing gene expression profile for certain cell types by other similar cell types, whether our model would have similar performance in this scenario is yet to be confirmed.

7 Conclusion

In this project, I modified existing reference-free model which uses LDA, by incorporating the single cell reference data as prior. I also tested the performance of proposed model by applying it on both simulated and real dataset. Results suggest that incorporating the prior indeed improves performance, particularly when the spatial data signal is weak. For the Visium data analysis, deconvolution performance indicates that many deconvolved cell types can be matched, but discrepancies still exist.

Acknowledgement. I would like to thank my supervisor, Dr Heejung Shim, for her consistent support and guidance throughout the duration of this project. Mathematical derivations, model interpretations, results analysis and the writing of this report were completed with close assistance from my supervisor.

Appendix A Derivations

Appendix A.1 Proof KL Divergence is non-negative

$$\begin{aligned}
-\mathrm{KL}(q(\theta)||p(\theta)) &= \int q(\theta)\log\frac{p(\theta)}{q(\theta)} \\
&= \mathbb{E}_q\left[\log\frac{p(\theta)}{q(\theta)}\right] \\
&\leq \log\left(\mathbb{E}_q\left[\frac{p(\theta)}{q(\theta)}\right]\right) \quad \text{by Jensen's inequality} \\
&= \log\left(\int p(\theta)\,d\theta\right) \\
&= 0
\end{aligned}$$

Hence, $\mathrm{KL}(q(\theta)||p(\theta)) \geq 0$.

Appendix A.2 ELBO derivations

Recall that the ELBO function can be expressed as,

$$\begin{aligned}
\mathrm{ELBO}(\gamma,\phi,\tau,\alpha|w,\eta,b) = &\,\mathbb{E}_q\left[\log p(\beta)\right] + \mathbb{E}_q\left[\log p(\theta|\alpha)\right] + \mathbb{E}_q\left[\log p(z|\theta)\right] + \mathbb{E}_q\left[\log p(w|z,\beta)\right] \\
&- \mathbb{E}_q\left[\log q(\theta)\right] - \mathbb{E}_q\left[\log q(z)\right] - \mathbb{E}_q\left[\log q(\beta)\right]
\end{aligned}$$

We first note the following:

Let $X \sim Dir(\beta)$, X is a k dimensional random vector. We borrow the following result,

$$\mathbb{E}_X\left[\log X_i\right] = \psi(\beta_i) - \psi\left(\sum_{j=1}^{k}\beta_j\right)$$

Where $\psi(\cdot)$ is the digamma function. This result can be proved using the fact that $\log X$ is the sufficient statistic for the dirichlet distribution, which is in the exponential family.

$$\begin{aligned}
\mathbb{E}_X\left[\log f_X(X)\right] &= \mathbb{E}_X\left[\log\left(\frac{\Gamma\left(\sum_{i=1}^{k}\beta_i\right)}{\prod_{i=1}^{k}\Gamma(\beta_i)}\prod_{i=1}^{k}X_i^{\beta_i-1}\right)\right] \\
&= \log\Gamma\left(\sum_{i=1}^{k}\beta_i\right) - \sum_{i=1}^{k}\log\Gamma(\beta_i) + \mathbb{E}\left[\sum_{i=1}^{k}(\beta_i-1)\log X_i\right] \\
&= \log\Gamma\left(\sum_{i=1}^{k}\beta_i\right) - \sum_{i=1}^{k}\log\Gamma(\beta_i) + \sum_{i=1}^{k}(\beta_i-1)\cdot\left(\psi(\beta_i) - \psi\left(\sum_{j=1}^{k}\beta_j\right)\right)
\end{aligned}$$

In the original ELBO function, given that we are taking expectation w.r.t the variational distribution q, we have β, $\theta|\alpha$, $q(\theta)$ and $q(\beta)$ all following dirichlet distribution, and so we can use the form from above. In particular,

1. $p(\beta_{i\cdot}) \sim Dir(b\eta_{i\cdot})$, while under expectation q, $\beta_{i\cdot} \sim Dir(\tau_{i\cdot})$
2. $p(\theta|\alpha) \sim Dir(\alpha)$, while under expectation q, $\theta_i \sim Dir(\gamma_{i\cdot})$
3. $q(\theta_{i\cdot}) \sim Dir(\gamma_{i\cdot})$
4. $q(\beta_{i\cdot}) \sim Dir(\tau_{i\cdot})$

Note that for 1. and 2., $p(\beta_{i\cdot})$ and $p(\theta|\alpha)$ represent prior distributions specified in the LDA model, instead of the variational distribution of β and θ respectively. The four terms can now be identified as follows,

$$\begin{aligned}
\mathbb{E}_q\left[\log p(\beta)\right] &= \sum_{k=1}^{K}\left[\log\Gamma\left(\sum_{n=1}^{N} b\eta_{kn}\right) - \sum_{n=1}^{N}\log\Gamma(b\eta_{kn})) + \sum_{n=1}^{N}(b\eta_{kn}-1)\cdot\left(\psi(\tau_{kn}) - \psi\left(\sum_{j=1}^{N}\tau_{kj}\right)\right)\right] \\
\mathbb{E}_q\left[\log p(\theta|\alpha)\right] &= \sum_{d=1}^{D}\left[\log\Gamma\left(\sum_{k=1}^{K}\alpha_k\right) - \sum_{k=1}^{K}\log\Gamma(\alpha_k)) + \sum_{k=1}^{K}(\alpha_k-1)\cdot\left(\psi(\gamma_{dk}) - \psi\left(\sum_{j=1}^{K}\gamma_{dj}\right)\right)\right] \\
\mathbb{E}_q\left[\log q(\theta)\right] &= \sum_{d=1}^{D}\left[\log\Gamma\left(\sum_{k=1}^{K}\gamma_{dk}\right) - \sum_{k=1}^{K}\log\Gamma(\gamma_{dk})) + \sum_{k=1}^{K}(\gamma_{dk}-1)\cdot\left(\psi(\gamma_{dk}) - \psi\left(\sum_{j=1}^{K}\gamma_{dj}\right)\right)\right] \\
\mathbb{E}_q\left[\log q(\beta)\right] &= \sum_{k=1}^{K}\left[\log\Gamma\left(\sum_{n=1}^{N}\tau_{kn}\right) - \sum_{n=1}^{N}\log\Gamma(\tau_{kn})) + \sum_{n=1}^{N}(\tau_{kn}-1)\cdot\left(\psi(\tau_{kn}) - \psi\left(\sum_{j=1}^{N}\tau_{kj}\right)\right)\right]
\end{aligned}$$

For $\mathbb{E}_q\left[\log p(z|\theta)\right]$:

$$\begin{aligned}
\mathbb{E}_q\left[\log p(z|\theta)\right] &= \mathbb{E}_q\left[\log\left(\prod_{d=1}^{D}\prod_{m=1}^{M_d}\prod_{k=1}^{K} p(z_{dm}=k|\theta)^{\mathbb{I}(z_{dm}=k)}\right)\right] \\
&= \sum_{d=1}^{D}\sum_{m=1}^{M_d}\sum_{k=1}^{K}\mathbb{E}_q\left[\mathbb{I}(z_{dm}=k)\log\theta_{dk}\right] \\
&= \sum_{d=1}^{D}\sum_{m=1}^{M_d}\sum_{k=1}^{K}\phi_{dmk}\left[\psi(\gamma_{dk}) - \psi\left(\sum_{j=1}^{k}\gamma_{dj}\right)\right] \text{ by parameter independence} \\
&= \sum_{d=1}^{D}\sum_{n=1}^{N} w_{dn}\sum_{k=1}^{K}\phi_{dnk}\left[\psi(\gamma_{dk}) - \psi\left(\sum_{j=1}^{k}\gamma_{dj}\right)\right]
\end{aligned}$$

Where w_{dn} represents the count of gene n in pixel d in spatial data. The reason we can compute the last step is because there is no difference between two molecules having the same gene in the same pixel. We slightly abuse the notation here that in the second last line, ϕ_{dmk} represents the proportion of mth molecule in pixel d belonging to cell type k, while ϕ_{dnk} in the last line represents the proportion of molecules with gene n in pixel d belonging to cell type k.

For $\mathbb{E}_q\left[\log p(w|z,\beta)\right]$:

$$
\begin{aligned}
\mathbb{E}_q\left[\log p(w|z,\beta)\right] &= \mathbb{E}_q\left[\log\left(\prod_{d=1}^{D}\prod_{m=1}^{M_d}\prod_{k=1}^{K}\prod_{n=1}^{N} p(v_{dm}^n = 1|z_{dm}=k,\beta)^{\mathbb{I}(z_{dm}=k)v_{dm}^n}\right)\right] \\
&= \sum_{d=1}^{D}\sum_{m=1}^{M_d}\sum_{k=1}^{K}\sum_{n=1}^{N} \mathbb{E}_q\left[\mathbb{I}(z_{dm}=k)v_{dm}^n p(v_{dm}^n = 1|z_{dm}=k,\beta)\right] \\
&= \sum_{d=1}^{D}\sum_{m=1}^{M_d}\sum_{k=1}^{K}\sum_{n=1}^{N} \phi_{dmk}v_{dm}^n\,\mathbb{E}_q\left[\beta_{kn}\right] \text{ by parameter independence} \\
&= \sum_{d=1}^{D}\sum_{m=1}^{M_d}\sum_{k=1}^{K}\phi_{dmk}\sum_{n=1}^{N} v_{dm}^n\left(\psi(\tau_{kn}) - \psi\left(\sum_{j=1}^{N}\tau_{kj}\right)]\right)
\end{aligned}
$$

In this equation, v_{dm}^n is the indicator that pixel d molecule m has gene n, because this is observed data so it becomes a constant. ϕ_{dmk} represents the estimated proportion of molecule m in pixel d belonging to cell type k. Again, because molecules in the same pixel having same gene are equivalent, we can rewrite the expression above in terms of ϕ_{dnk}, which represents the estimated proportion of molecules with gene n in pixel d belonging to cell type k, as well as letting w_{dn} represent counts of gene n in pixel d.

$$
\mathbb{E}_q\left[\log p(w|z,\beta)\right] = \sum_{d=1}^{D}\sum_{n=1}^{N}\sum_{k=1}^{K}\phi_{dnk}w_{dn}\left(\psi(\tau_{kn}) - \psi\left(\sum_{j=1}^{N}\tau_{kj}\right)\right)
$$

Finally, for $\mathbb{E}_q\left[q(z)\right]$:

$$
\begin{aligned}
\mathbb{E}_q\left[q(z)\right] &= \mathbb{E}_q\left[\log\left(\prod_{d=1}^{D}\prod_{m=1}^{M_d}\prod_{k=1}^{K} q(z_{dm}=k)^{\mathbb{I}(z_{dm}=k)}\right)\right] \\
&= \sum_{d=1}^{D}\sum_{m=1}^{M_d}\sum_{k=1}^{K}\mathbb{E}_q\left[\mathbb{I}(z_{dm}=k)\log\phi_{dmk}\right] \\
&= \sum_{d=1}^{D}\sum_{m=1}^{M_d}\sum_{k=1}^{K}\phi_{dmk}\log\phi_{dmk} \\
&= \sum_{d=1}^{D}\sum_{n=1}^{N}\sum_{k=1}^{K}w_{dn}\phi_{dnk}\log\phi_{dnk}
\end{aligned}
$$

Again, note the difference between ϕ_{dmk} and ϕ_{dnk}. Definition of w_{dn} remain unchanged.

Appendix A.3 Parameter Update

To obtain parameter updates for ϕ, γ and τ. We solve the partial derivative of the ELBO function w.r.t these parameters equal to 0. For all derivations below, λ represents the Lagrange multiplier.

For ϕ_{dnk}:

$$\begin{aligned}
\text{ELBO}[\phi] &= \sum_{d=1}^{D}\sum_{n=1}^{N}\sum_{k=1}^{K}\{\phi_{dnk}w_{dn}\left[\psi(\tau_{kn})-\psi\left(\sum_{j=1}^{N}\tau_{kj}\right)\right]+w_{dn}\phi_{dnk}\left[\psi(\gamma_{dk})-\psi\left(\sum_{j=1}^{k}\gamma_{dj}\right)\right] \\
&\quad - w_{dn}\phi_{dnk}\log\phi_{dnk}\} + \sum_{n=1}^{N}\lambda_{dn}\left(\sum_{k=1}^{K}\phi_{dnk}-1\right) \\
\frac{\partial \text{ELBO}[\phi]}{\partial \phi_{dnk}} &= w_{dn}\left[\psi(\tau_{kn})-\psi\left(\sum_{j=1}^{N}\tau_{kj}\right)\right]+w_{dn}\left[\psi(\gamma_{dk})-\psi\left(\sum_{j=1}^{k}\gamma_{dj}\right)\right]-w_{dn}(\log\phi_{dnk}+1)+\lambda_{dn}
\end{aligned}$$

Solve $\frac{\partial \text{ELBO}[\phi]}{\partial \phi_{dnk}}=0$:

$$\begin{aligned}
\log\phi_{dnk}+1 &= \psi(\tau_{kn})-\psi\left(\sum_{j=1}^{N}\tau_{kj}\right)+\psi(\gamma_{dk})-\psi\left(\sum_{j=1}^{k}\gamma_{dj}\right)+\frac{\lambda_{dn}}{w_{dn}} \\
\phi_{dnk} &\propto \exp\left[\psi(\tau_{kn})-\psi\left(\sum_{j=1}^{N}\tau_{kj}\right)+\psi(\gamma_{dk})-\psi\left(\sum_{j=1}^{k}\gamma_{dj}\right)\right]
\end{aligned}$$

Note that we normalize such that $\sum_{k=1}^{K}\phi_{dnk}=1$ and so λ_{dn} and w_{dn} are constants that can be safely ignored.

For γ_{dk}:

$$\begin{aligned}
\text{ELBO}[\gamma] &= \sum_{d=1}^{D}\sum_{k=1}^{K}\left\{(\alpha_k-1)\cdot\left(\psi(\gamma_{dk})-\psi\left(\sum_{j=1}^{K}\gamma_{dj}\right)\right)+\sum_{n=1}^{N}w_{dn}\phi_{dnk}\left[\psi(\gamma_{dk})-\psi\left(\sum_{j=1}^{K}\gamma_{dj}\right)\right]\right\} \\
&\quad -\sum_{d=1}^{D}\left[\log\Gamma\left(\sum_{k=1}^{K}\gamma_{dk}\right)-\sum_{k=1}^{K}\log\Gamma(\gamma_{dk}))+\sum_{k=1}^{K}(\gamma_{dk}-1)\cdot\left(\psi(\gamma_{dk})-\psi\left(\sum_{j=1}^{K}\gamma_{dj}\right)\right)\right] \\
\frac{\partial \text{ELBO}[\gamma]}{\partial \gamma_{dk}} &= (\alpha_k-1)\psi'(\gamma_{dk})-\sum_{i=1}^{K}(\alpha_i-1)\psi'\left(\sum_{j=1}^{K}\gamma_{dj}\right)+\psi'(\gamma_{dk})\sum_{n=1}^{N}w_{dn}\phi_{dnk}-\sum_{i=1}^{K}\sum_{n=1}^{N}w_{dn}\phi_{dni}\psi'\left(\sum_{j=1}^{K}\gamma_{dj}\right) \\
&\quad -\psi\left(\sum_{i=1}^{K}\gamma_{di}\right)+\psi(\gamma_{dk})-\psi(\gamma_{dk})+\psi\left(\sum_{i=1}^{K}\gamma_{di}\right)-(\gamma_{dk}-1)\psi'(\gamma_{dk})+\sum_{i=1}^{K}(\gamma_{di}-1)\psi'\left(\sum_{j=1}^{K}\gamma_{dj}\right) \\
&= \psi'(\gamma_{dk})\left(\alpha_k+\sum_{n=1}^{N}w_{dn}\phi_{dnk}-\gamma_{dk}\right)-\psi'\left(\sum_{j=1}^{K}\gamma_{dj}\right)\sum_{i=1}^{K}\left(\alpha_i+\sum_{n=1}^{N}w_{dn}\phi_{dni}-\gamma_{di}\right)
\end{aligned}$$

Hence, solving $\frac{\partial \text{ELBO}[\gamma]}{\partial \gamma_{dk}}=0$ gives

$$\gamma_{dk}=\alpha_k+\sum_{n=1}^{N}w_{dn}\phi_{dnk}$$

For τ_{kn}:

$$\begin{aligned}
\text{ELBO}[\tau] &= \sum_{k=1}^{K}\sum_{n=1}^{N}\left\{(b\eta_{kn}-1)\cdot\left(\psi(\tau_{kn})-\psi\left(\sum_{j=1}^{N}\tau_{kj}\right)\right)+\sum_{d=1}^{D}w_{dn}\phi_{dnk}\left[\psi(\tau_{kn})-\psi\left(\sum_{j=1}^{N}\gamma_{kj}\right)\right]\right\}\\
&\quad-\sum_{k=1}^{K}\left[\log\Gamma\left(\sum_{n=1}^{N}\tau_{kn}\right)-\sum_{n=1}^{N}\log\Gamma(\tau_{kn}))+\sum_{n=1}^{N}(\tau_{kn}-1)\cdot\left(\psi(\tau_{kn})-\psi\left(\sum_{j=1}^{N}\tau_{kj}\right)\right)\right]\\
\frac{\partial\text{ELBO}[\tau]}{\partial\tau_{kn}} &= (b\eta_{kn}-1)\psi'(\tau_{kn})-\sum_{i=1}^{N}(\eta_{ki}-1)\psi'\left(\sum_{j=1}^{N}\tau_{kj}\right)+\psi'(\tau_{kn})\sum_{d=1}^{D}w_{dn}\phi_{dnk}-\sum_{i=1}^{N}\sum_{d=1}^{D}w_{di}\phi_{dik}\psi'\left(\sum_{j=1}^{N}\tau_{dj}\right)\\
&\quad-\psi\left(\sum_{i=1}^{N}\tau_{ki}\right)+\psi(\tau_{kn})-\psi(\tau_{kn})+\psi\left(\sum_{i=1}^{N}\tau_{ki}\right)-(\tau_{kn}-1)\psi'(\tau_{kn})+\sum_{i=1}^{N}(\tau_{ki}-1)\psi'\left(\sum_{j=1}^{N}\tau_{kj}\right)\\
&= \psi'(\tau_{kn})\left(b\eta_{kn}+\sum_{d=1}^{D}w_{dn}\phi_{dnk}-\tau_{kn}\right)-\psi'\left(\sum_{j=1}^{N}\tau_{kj}\right)\sum_{i=1}^{N}\left(b\eta_{kn}+\sum_{d=1}^{D}w_{dn}\phi_{dni}-\tau_{ki}\right)
\end{aligned}$$

Hence, solving $\frac{\partial\text{ELBO}[\tau]}{\partial\tau_{kn}}=0$ gives

$$\tau_{kn}=b\eta_{kn}+\sum_{d=1}^{D}w_{dn}\phi_{dnk}$$

Newton-Raphson Method. Finally, update for α will be based on Newton-Raphson method as described in [4]. It is an optimization technique which finds the maximum of a function $f(\alpha)$ by iterating:

$$\alpha_{i+1}=\alpha_i-H(\alpha_i)^{-1}g(\alpha_i)$$

Where $H(\alpha_i)$ and $g(\alpha_i))$ are the Hessian matrix and gradient vector respectively. If the Hessian matrix can be expressed as $H(\alpha_i)=diag(h)+\mathbf{1}z\mathbf{1}^\top$, where $diag(h)$ is a diagonal matrix with the diagonal elements being vector h, then the update formula can be simplified to:

$$\begin{aligned}
(H(\alpha_i)^{-1}g(\alpha_i))_j &= \frac{g_j-c}{h_j}\\
\text{where } c &= \frac{\sum_{m=1}^{K}g_m/h_m}{z^{-1}+\sum_{m=1}^{K}h_m^{-1}}
\end{aligned}$$

In our derivation below, we change the meaning of notation α_i to represent the ith component of vector α, instead of the ith iteration.

$$
\begin{aligned}
\text{ELBO}[\alpha] &= \sum_{d=1}^{D}\left[\log\Gamma\left(\sum_{k=1}^{K}\alpha_k\right) - \sum_{k=1}^{K}\log\Gamma(\alpha_k)) + \sum_{k=1}^{K}(\alpha_k - 1)\cdot\left(\psi(\gamma_{dk}) - \psi\left(\sum_{j=1}^{K}\gamma_{dj}\right)\right)\right] \\
\frac{\partial\text{ELBO}[\alpha]}{\partial\alpha_i} &= \sum_{d=1}^{D}\left[\psi\left(\sum_{k=1}^{K}\alpha_k\right) - \psi(\alpha_i) + \psi(\gamma_{di}) - \psi\left(\sum_{j=1}^{K}\gamma_{dj}\right)\right] \\
&= D\left[\psi\left(\sum_{k=1}^{K}\alpha_k\right) - \psi(\alpha_i)\right] + \sum_{d=1}^{D}\left[\psi(\gamma_{di}) - \psi\left(\sum_{j=1}^{K}\gamma_{dj}\right)\right] = g_i \\
\frac{\partial^2\text{ELBO}[\alpha]}{\partial\alpha_i\partial\alpha_j} &= D\left(\psi'\left(\sum_{k=1}^{K}\alpha_k\right) - \mathbb{I}(i=j)\psi'(\alpha_i)\right)
\end{aligned}
$$

Hence, the Hessian matrix is of the form required for simplification, with

$$
\begin{aligned}
[z]_{ij} &= D\psi'\left(\sum_{k=1}^{K}\alpha_k\right) \\
h_i &= -D\psi'(\alpha_i)
\end{aligned}
$$

Appendix B Supplementary Figures

Appendix B.1 Correlation between β and η

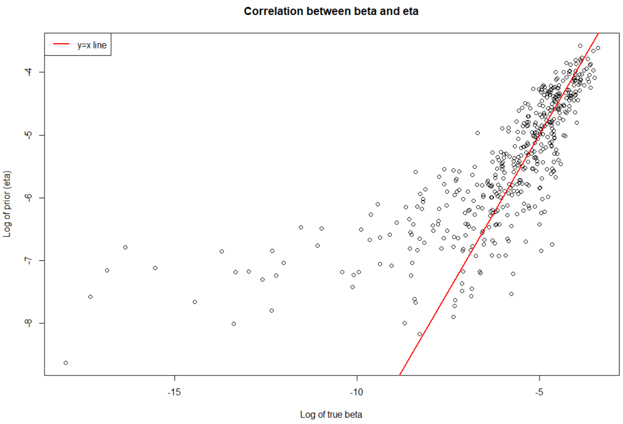


Figure 16: Comparison of β and η on a log scale for strong signal data.

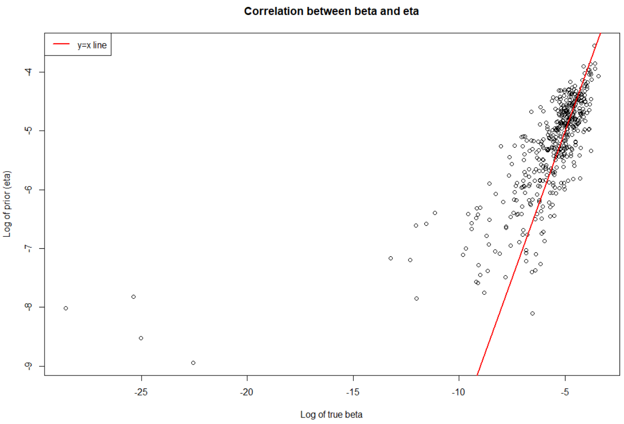


Figure 17: Comparison of β and η on a log scale for weak signal data.

Appendix B.2 Visium Dataset cell types comparison.

Visualization tool used in this section is developed in the STdeconvolve paper [1].

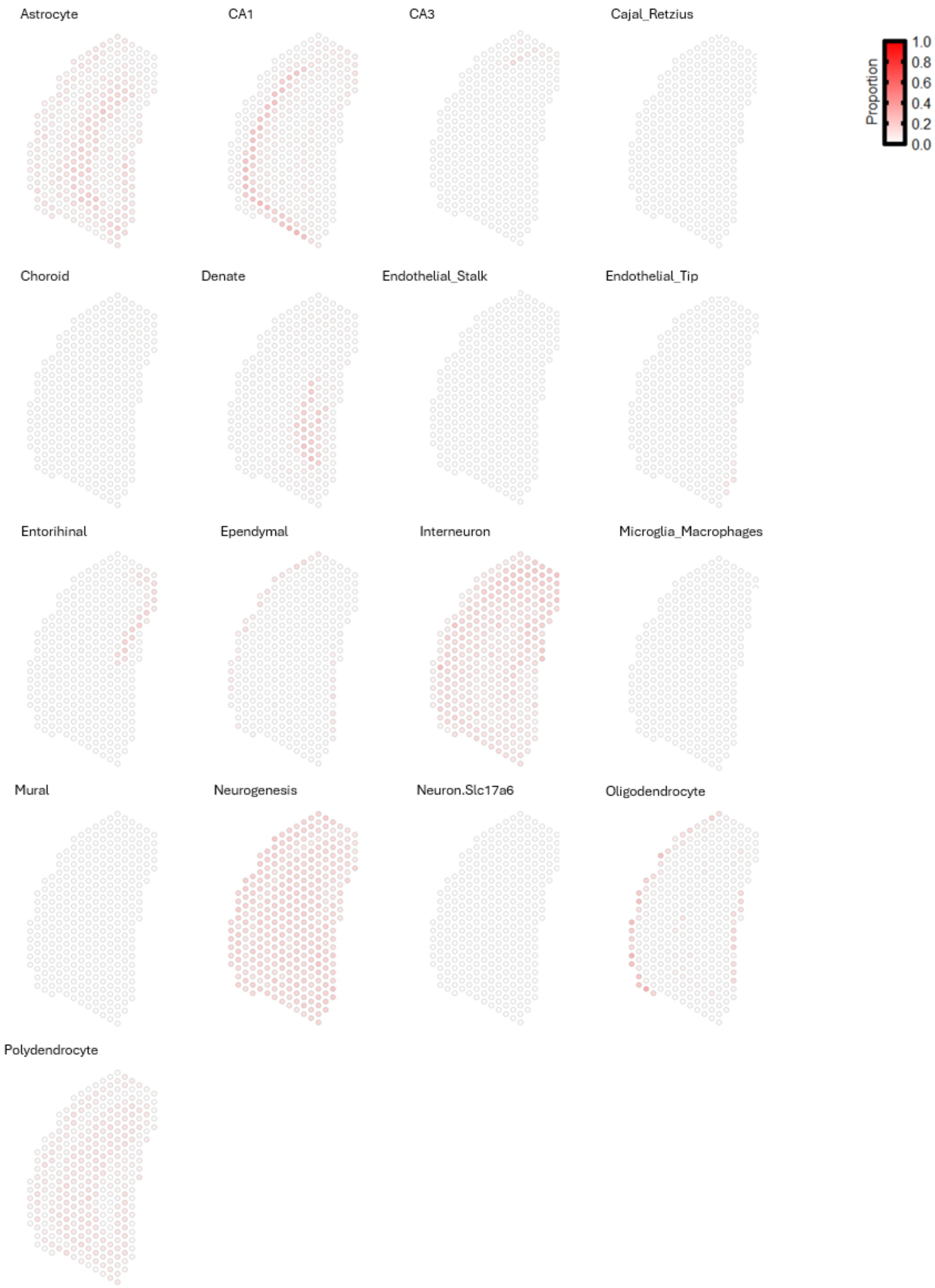


Figure 18: Deconvolved proportions for each cell type using RCTD.

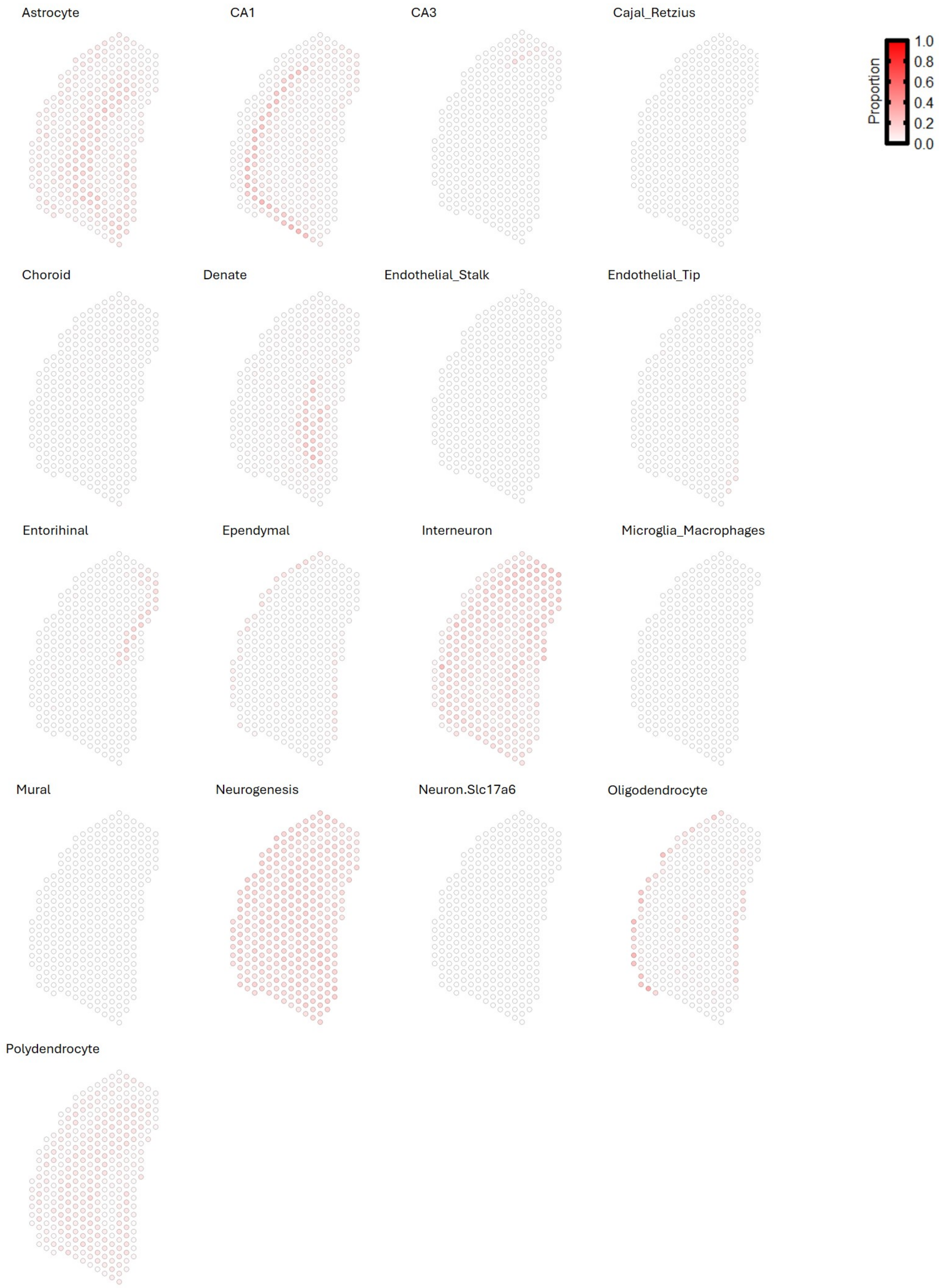


Figure 19: Deconvolved proportions for each cell type using our model with prior constant 500.

References

[1] B. F. Miller, F. Huang, L. Atta, A. Sahoo, and J. Fan, "Reference-free cell type deconvolution of multi-cellular pixel-resolution spatially resolved transcriptomics data," *Nature communications*, vol. 13, no. 1, 2022.

[2] D. M. Cable, E. Murray, L. S. Zou, A. Goeva, E. Z. Macosko, F. Chen, and R. A. Irizarry, "Robust decomposition of cell type mixtures in spatial transcriptomics," *Nat. Biotechnol.*, vol. 40, pp. 517–526, Apr. 2022.

[3] M. Elosua-Bayes, P. Nieto, E. Mereu, I. Gut, and H. Heyn, "SPOTlight: seeded NMF regression to deconvolute spatial transcriptomics spots with single-cell transcriptomes," *Nucleic Acids Res.*, vol. 49, p. e50, May 2021.

[4] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *Journal of Machine Learning Research*, 2003.

[5] T. Hofmann, "Probabilistic latent semantic analysis," 2013.

[6] A. Raj, M. Stephens, and J. K. Pritchard, "fastSTRUCTURE: Variational Inference of Population Structure in Large SNP Data Sets," *Genetics*, vol. 197, pp. 573–589, 06 2014.

[7] Y. Vanrobaeys, U. Mukherjee, L. Langmack, S. E. Beyer, E. Bahl, L.-C. Lin, J. J. Michaelson, T. Abel, and S. Chatterjee, "Mapping the spatial transcriptomic signature of the hippocampus during memory consolidation," *Nature Communications*, vol. 14, Sept. 2023.