Robust decomposition of cell type mixtures in spatial transcriptomics

Dylan M. Cable[1,2,3], Evan Murray[2], Luli S. Zou[2,3,4], Aleksandrina Goeva[2], Evan Z. Macosko[2,5], Fei Chen[2,6,*], and Rafael A. Irizarry[3,4,*]

[1]Department of Electrical Engineering and Computer Science, MIT, Cambridge, MA, 02139
[2]Broad Institute of Harvard and MIT, Cambridge, MA, 02142
[3]Department of Data Sciences, Dana-Farber Cancer Institute, Boston, MA, 02215
[4]Department of Biostatistics, Harvard University, Boston, MA, 02115
[5]Department of Psychiatry, Massachusetts General Hospital, Boston, MA, 02114
[6]Department of Stem Cell and Regenerative Biology, Harvard University, Cambridge MA 02138
[*]These authors contributed equally
Correspondence to: rafa@ds.dfci.harvard.edu, chenf@broadinstitute.org

Supplementary Methods

Estimation of cell type means

We estimate $\hat{\mu}_{k,j}$ as the empirical mean of normalized expression of gene i in cell type $j \in J$ within the scRNA-seq reference:

$$\hat{\mu}_{k,j} \equiv \frac{1}{I_k} \sum_{i=1}^{I_k} \frac{Y_{i,k,j}}{N_{i,k}} \tag{1}$$

Here, I_k is the number of cells in reference of cell type k, $N_{i,k}$ is the number of UMIs of cell i of cell type k, and $Y_{i,k,j}$ is observed counts of gene j in this cell. If we assume there are many cells, then this will be an accurate and unbiased estimator of the true mean $\mu_{k,j}$.

Probabilistic model

In this section, we revisit our probabilistic model of spatial transcriptomics, deriving the Poisson sampling model from a more detailed independently-sampled unique molecular identifier (UMI) model. For each UMI, the source cell type is first probabilistically determined based on the cell type mixture proportions. Second, the gene is determined according to the probability vector of that cell type. Formally, for each pixel $1 \leq i \leq I$ and for each read $1 \leq n \leq N_i$, we select cell type $1 \leq \theta_{n,i} \leq K$ and gene $1 \leq z_{n,i} \leq J$ as:

$$P(\theta_{n,i} = k \mid \beta) = \beta_{k,i}, \quad P(z_{n,i} = j \mid \theta_{n,i}, \gamma, \varepsilon) = \delta_{i,\theta_{n,i},j} \tag{2}$$

Here, we define:

$$\log(\delta_{i,k,j}) = \alpha_i + \log(\hat{\mu}_{k,j}) + \gamma_j + \varepsilon_{i,j}, \tag{3}$$

$$\text{where } \gamma_j \sim \text{Normal}(0, \sigma_\gamma^2), \quad \varepsilon_{i,j} \sim \text{Normal}(0, \sigma_\varepsilon^2), \quad \alpha_i \text{ fixed} \tag{4}$$

Similar to the probabilistic model of Latent Dirichlet Allocation [1], in the RCTD model, individual reads are sampled from a randomly chosen gene from a randomly chosen cell type. RCTD extends the LDA model with its introduction of platform effects λ_j and random effects $\varepsilon_{i,j}$ accounting for gene-specific overdispersion.

We define $\lambda_{i,j}$ (predicted gene probability) as:

$$\lambda_{i,j} \equiv P(z_{n,i} = j \mid \gamma, \varepsilon) = \sum_{k=1}^{K} \beta_{i,k}\mu_{k,j} \tag{5}$$

Multinomial to Poisson approximation

As a consequence of our model, conditional on $\lambda_{i,j}$, $Y_{i,j}$ is distributed as the Multinomial distribution:

$$\begin{gathered} Y_{i,j} = \sum_{n=1}^{N_i} \mathbb{I}[z_{n,i} = j], \\ Y_{i,1}, Y_{i,2}, \ldots, Y_{i,J} \mid \lambda \sim \text{Multinomial}(N_i, \lambda_{i,1}, \lambda_{i,2}, \ldots, \lambda_{i,J}). \end{gathered} \tag{6}$$

We will now argue to approximate this distribution using:

$$Y_{i,j} \overset{\text{ind}}{\sim} \text{Poisson}(N_i\lambda_{i,j}), \tag{7}$$

thus justifying the use of the Poisson model by RCTD.

Given N_i reads, and each read occurs independently with $\lambda_{i,j}$ representing the probability of gene j, there are two ways of arriving at a Poisson approximation. First, we can assume that the number of UMIs N_i is a Poisson random variable rather than fixed. That is $N_i \sim \text{Poisson}(\mu_i)$, and $Y_{i,1}, Y_{i,2}, \ldots, Y_{i,J} \mid N_i \sim \text{Multinomial}(N_i, \lambda_{i,1}, \lambda_{i,2}, \ldots, \lambda_{i,J})$. It turns out that this formulation is equivalent to:

$$Y_{i,j} \overset{\text{ind}}{\sim} \text{Poisson}(\mu_i \cdot \lambda_{i,j}) \tag{8}$$

where the Poissonization of N_i leads to independence across genes [2,3]. Although μ_i is not known, it is approximately correct to estimate $\hat{\mu}_i = \sum_j Y_{i,j} = N_i$. This approximation is valid when μ_i is large because Poisson(μ_i) will have standard deviation $\sqrt{\mu_i}$ which will lead to a small percentage error. We are in the regime of large μ_i because we only consider pixels with $N_i \geq 100$.

A second approach is to assume that N_i is fixed (observed), and that,

$$Y_{i,1}, Y_{i,2}, \ldots, Y_{i,J} \sim \text{Multinomial}(N_i, \lambda_{i,1}, \lambda_{i,2}, \ldots, \lambda_{i,J}). \tag{9}$$

For counts-based gene expression data, this expression has been previously approximated as a Poisson distribution. Details have previously been provided in [4]. In particular, this approximation is accurate when N_i is large and $\lambda_{i,j}$ is small. We do have that $N_i \geq 100$, and that for most genes $\lambda_{i,j}$ is small ($< 10^{-3}$ for e.g. 98.7% of genes used in the cerebellum Slide-seq from snRNA-seq predictions).

Role of α_i

We now provide further insight into the role of α_i. Briefly, the parameter α_i controls the overall rate of gene expression per pixel, and is equivalent to treating the number of UMIs, N_i, as a free parameter. While constraining $\alpha_i = -\sigma^2/2$ would enforce the mean gene counts to be $N_i\lambda_{i,j}$, we observed better model fits by leaving α_i unconstrained.

More precisely, α_i can best be understood as controlling, for pixel i, the probability that each read is in gene set $J = \{1, 2, \ldots, J\}$, which represents a chosen subset of all genes.

$$P(z_{n,i} \in J) = \sum_{j}^{J} P(z_{n,i} = j \mid \theta_{n,i}, \gamma, \varepsilon_{i,j}) = \sum_{j}^{J} \delta_{i,\theta_{n,i},j} \tag{10}$$

Considering that this sum will be a weighted sum of the cell types present, for simplicity, we will consider this sum for each cell type k. We can compute (for an individual pixel i and cell type k):

$$\sum_j^J \delta_{i,k,j} = \sum_j^J \hat{\mu}_{k,j} e^{\alpha_i + \gamma_j + \varepsilon_{i,j}} \tag{11}$$

Since $\hat{\mu}_{k,j}$ is the average observed expression of gene j in the scRNA-seq reference, we would have $\sum_j^J \hat{\mu}_{k,j} = 1$ if using all genes or $\hat{\mu}_{k,j} \equiv S_k < 1$ if using a subset of genes. Also since γ_j and $\varepsilon_{i,j}$ are independent normal random variables, their sum,

$$\beta_{i,j} = \gamma_j + \varepsilon_{i,j} \sim \text{Normal}(0, \sigma^2), \quad \sigma^2 = \sigma_\varepsilon^2 + \sigma_\gamma^2, \tag{12}$$

is also normally distributed. Therefore:

$$\sum_j^J \delta_{i,k,j} = e^{\alpha_i} \sum_j^J \hat{\mu}_{k,j} e^{\beta_{i,j}} \approx e^{\alpha_i} \sum_j^J \hat{\mu}_{k,j} \mathbb{E}[e^{\beta_{i,j}}] = e^{\alpha_i} S_k e^{\sigma^2/2} \tag{13}$$

Where the above approximation will be with high probability accurate. One can prove that such an approximation is accurate with high probability using the same arguments that appear in the supplemental section "Quantitative bound for deviation of sum of independent random variables". Lastly, we used the expectation of a lognormal distribution.

Now, we can see that assumption that is made about the summed probabilities $\sum_j^J \delta_{i,k,j}$, or, equivalently, the total probability of observing a gene in our set of selected genes $1 \leq j \leq J$. If one were to constrain $\alpha_i = -\sigma^2/2$, this would be equivalent to assuming that $\sum_j^J \delta_{i,k,j} = S_k$, that the total probability of the gene set is the same between the scRNA-seq reference and each spatial transcriptomics pixel. However, this assumption is not generally true, as can be seen by e.g. selecting genes in snRNA-seq and observing the proportion of those genes in scRNA-seq, within the same cell type. We often found that selecting for highly expressed genes in the snRNA-seq dataset tend to be in total less expressed in the target dataset/platform. As a result, we chose not to enforce the assumption that $\sum_j^J \delta_{i,k,j} = S_k$, but rather to allow α_i to be a free parameter that controls the total probability of observing a gene in this gene set. Although we chose this approach, we certainly think it is worth considering other approaches such as (1) constraining $\alpha_i = -\sigma^2/2$, (2) shrinking the estimates of α_i using an empirical Bayes procedure.

Platform Effect Normalization

The platform effects γ_i can be reliably estimated independently from the other parameters by summarizing the spatial transcriptomics data as a single *pseudo-bulk* measurement S_j:

$$S_j \equiv \sum_{i=1}^I Y_{i,j} \sim \text{Poisson}(\sum_{i=1}^I N_i \lambda_{i,j}) \tag{14}$$

where S_j are observed. Next, we calculate the rate parameter of this Poisson distribution:

$$\begin{aligned}\mathbb{E}[S_j|\gamma] = \sum_{i=1}^{I} N_i \lambda_{i,j} &= \sum_{i=1}^{I}\sum_{k=1}^{K} N_i \beta_{k,i} \delta_{i,k,j} = \sum_{k=1}^{K}\sum_{i=1}^{I} N_i \beta_{k,i} \mu_{k,j} e^{\gamma_j + \alpha_i + \varepsilon_{i,j}} \\ &= e^{\gamma_j} \sum_{k=1}^{K} \mu_{k,j} \sum_{i=1}^{I} N_i \beta_{k,i} e^{\alpha_i} e^{\varepsilon_{i,j}} \\ &= I e^{\gamma_j} \bar{N} \sum_{k=1}^{K} \mu_{k,j} B_{k,j} \\ &\approx I e^{\gamma_j} \bar{N} \sum_{k=1}^{K} \mu_{k,j} \beta_k \beta_0\end{aligned} \tag{15}$$

Where we have defined $\beta_0 \equiv \mathbb{E}[\varepsilon_{i,j}] = e^{\sigma_\varepsilon^2/2}$ as a fixed scaling constant (using lognormal expectation), $\beta_k = \frac{1}{I}\sum_{i=1}^{I} \frac{N_i}{\bar{N}} \beta_{i,k} e^{\alpha_i}$ is the average proportion of cell type k, and:

$$\bar{N} = \frac{1}{I}\sum_{i=1}^{I} N_i \;\text{ and }\; B_{k,j} = \frac{1}{I}\sum_{i=1}^{I} \frac{N_i}{\bar{N}} \beta_{k,i} \exp(\alpha_i + \varepsilon_{i,j}). \tag{16}$$

$B_{k,j}$ is a random variable that is approximately proportional to β_k:

$$B_{k,j} \approx \beta_k \beta_0.$$

This follows from the fact that $\mathbb{E}(B_{k,j}) = \beta_k\beta_0$, and $\mathrm{Var}(B_{k,j})$ converges to 0 when I is large, which we show now. Consider k to be fixed. Because each $B_{k,j}$ is independently and identically distributed, if we can show that with high probability, $B_{k,j}$ does not deviate too much from $\mathbb{E}[B_{k,j}]$, then we can justify our approximation. The $e^{\varepsilon_{i,j}}$ are independently and identically lognormally distributed. For each j, $B_{k,j}$ is the sum of independent random variables. If we assume $N_i\alpha_i$ is bounded and that we have many samples I, then Chebyshev's inequality implies that for any $\delta > 0$, as $I \to \infty$,

$$P(|B_{k,j} - \mathbb{E}[B_{k,j}]| > \delta) \to 0 \tag{17}$$

A quantitative bound to (17) for finite I can be derived by Chebyshev's inequality (See "Quantitative bound for deviation of sum of independent random variables"). As such, we are justified in making the approximation that $B_{k,j} \approx \beta_k\beta_0 \equiv \mathbb{E}[B_{k,j}]$ for all k, j.

Consequently, if we define $W_k = \beta_k\beta_0$,

$$S_j|\gamma_j \sim \text{Poisson}\left(I\bar{N}e^{\gamma_j}\sum_{k=1}^{K}\mu_{k,j}W_k\right), \quad \gamma_j \sim \text{Normal}(0, \sigma_\gamma^2). \tag{18}$$

We can estimate W_k by viewing this as an (unconstrained) Poisson-lognormal mixture model. Our procedure to calculate the MLE of W and σ_γ ($\hat{W}$ and $\hat{\sigma_\gamma}$) is the same procedure used to fit RCTD, which is described later (See fitting model (22)). Because the bulk Poisson mean is large for most genes, the Poisson sampling is mostly negligible. Accordingly, with high probability:

$$\bar{S}_j \approx e^{\gamma_i}\sum_{k=1}^{K}\mu_{k,j}W_k \implies \gamma_j \mid \hat{W} \approx \log(\bar{S}_j) - \log\left(\sum_{k=1}^{K}\mu_{k,j}\hat{W}_k\right) \equiv \hat{\gamma}_j, \tag{19}$$

where we have defined $\bar{S}_j = S_j/(I\bar{N})$. This is an approximation, given that there will be some deviation of $\hat{W}$ from the true W. We call this a normalization, because the effect of $\hat{\gamma}_j$ is to re-normalize the cell type means as:

$$\bar{\mu}_{k,j} = \mu_{k,j}e^{\hat{\gamma}_j} \tag{20}$$

Sequential quadratic programming for MLE estimation of RCTD

After estimating platform effects, we use RCTD to independently fit each spot as a weighted sum of cell types. Note that since γ_j has been replaced with a fixed estimate $\hat{\gamma}_j$, there are no shared parameters across different pixels i, except σ_ε. As such, we estimate $\beta_{k,i}$ independently for each pixel, given σ_ε. The parameter σ_ε is estimated by maximum likelihood via stochastic gradient descent (see "Estimating σ"). The parameter α_i effectively allows us to re-scale β_i, so we define $w_{k,i} = \beta_{k,i} e^{\alpha_i}$, which will not be constrained to sum to 1. Next, define:

$$\bar{\lambda}_{i,j}(w_i) = N_i \sum_{k=1}^{K} w_{k,i} \mu_{k,j} e^{\hat{\gamma}_j} = N_i \sum_{k=1}^{K} w_{k,i} \bar{\mu}_{k,j} \tag{21}$$

We will refer to this as the predicted mean of gene j in pixel i. The final model is a Poisson-lognormal mixture model:

$$Y_{i,j} \mid \bar{\lambda}_{i,j} \sim \text{Poisson}(e^{\varepsilon_{i,j}} \bar{\lambda}_{i,j}(w_i)), \quad \varepsilon_{i,j} \sim \text{Normal}(0, \sigma_\varepsilon^2) \tag{22}$$

We estimate $w_i^* \geq 0$ as the solution that maximizes the log-likelihood $\mathcal{L}(w_i)$:

$$\max \mathcal{L}(w_i) = \sum_{j=1}^{J} \log P(Y_{i,j} \mid \bar{\lambda}_{i,j}(w_i)) \quad \text{subject to: } w \geq 0 \tag{23}$$

Since the log-likelihood is non-convex, we implement sequential quadratic programming to optimize quadratic approximations and iterate until convergence - typically less than 15 iterations. Since this is the same model as equation (18) in Platform Effect Normalization, we use the same optimization procedure for both models.

Here, we provide the Sequential quadratic programming (SQP) optimization procedure to optimize the log-likelihood of RCTD (equation (23)). From now on, we consider a fixed pixel i and suppress the notation of i. We can directly compute $P(Y_j \mid \bar{\lambda}_j)$ by integrating over the random effect ε_j:

$$\begin{aligned} P(Y_j \mid \bar{\lambda}_j) &= \int_{-\infty}^{\infty} p_\sigma(z) P(Y_j \mid \lambda_j = \bar{\lambda}_j e^z) \mathrm{dz} \\ &= \int_{-\infty}^{\infty} p_\sigma(z) e^{-\bar{\lambda}_j e^z} \frac{(e^z \bar{\lambda}_j)^{Y_j}}{Y_j!} \mathrm{dz} \\ &= Q_{Y_j}(\bar{\lambda}_i) \end{aligned} \tag{24}$$

Here, p_σ is the probability density function of ε. From now on, for notational convenience, we will use λ_j to denote $\bar{\lambda}_j$. We have defined for $\ell \in \mathbb{Z}^+ \cup \{0\}$:

$$Q_\ell(\lambda) \equiv \int_{-\infty}^{\infty} p_\sigma(z) e^{-\lambda e^z} \frac{(e^z \lambda)^\ell}{\ell!} \mathrm{dz} \tag{25}$$

We will estimate $w^* \geq 0$ as that which maximizes the log-likelihood $\mathcal{L}(w)$:

$$\max_w \mathcal{L}(w) = \sum_{j=1}^{J} \log P(Y_j \mid \lambda_j(w)) = \sum_{j=1}^{J} \log Q_{Y_j}(\lambda_j(w)) \tag{26}$$

Our log likelihood is non-convex. To optimize it, we will apply Sequential Quadratic Programming, an iterative procedure that will be repeated until convergence. Let w_0 be the value of w at a given iteration, and let the gradient of $-\mathcal{L}$ be $b(w)$ and the Hessian of $-\mathcal{L}$ be $A(w)$. Then, we can make the following quadratic Taylor approximation to $\mathcal{L}$:

$$-\mathcal{L}(w) \approx -\mathcal{L}(w_0) + b(w_0)^T (w - w_0) + \frac{1}{2}(w - w_0)^T A(w_0)(w - w_0) \tag{27}$$

If we let $d = w - w_0$, then we get the following optimization problem to minimize our approximation of $-\mathcal{L}(w)$.

$$\begin{aligned} \min_{d} \quad & b(w_0)^T d + \frac{1}{2} d^T A(w_0) d \\ \text{s.t.} \quad & d + w_0 \geq 0 \end{aligned} \tag{28}$$

This is a constrained Quadratic Program (QP). There is one issue we must discuss. This QP will not be well behaved if the Hessian $A(w_0)$ is not positive semi-definite, which can occur due to the non-convexity of $-\mathcal{L}(w)$. As such, we employ an approximation which has been shown to be very effective in non-convex optimization [5]: we set $A(w_0)$ to be the positive semi-definite part of the hessian H of $-\mathcal{L}(w_0)$. Specifically, suppose we have an eigen-decomposition of H as:

$$H = VDV^T \tag{29}$$

Here, D is diagonal matrix of eigenvalues. We obtain the positive semi-definite part of H by taking $D^+ = \max(D, 0)$ and:

$$A = VD^+V^T \tag{30}$$

Effectively, taking the positive semi-definite part of the Hessian will preserve the gradient information and keep the convex part of the quadratic Taylor approximation.

Given a solution of d^*, we set

$$w_0 + \alpha * d^* \to w_0 \tag{31}$$

Here, α is the step size, which we have set to 0.3. We repeat this process until convergence. This gives us a solution w^* minimizing $-\mathcal{L}(w)$. We define convergence as convergence of w_n, the value of w after n SQP iterations. Specifically, we say that RCTD has converged once:

$$\|w_{n+1} - w_n\|_1 = \sum_k |w_{n+1,k} - w_{n,k}| < \epsilon. \tag{32}$$

By default, we use $\epsilon = 0.001$. We tested RCTD (on a randomly chosen subset of $1,000$ cerebellum Slide-seq pixels) with a smaller value of $\epsilon = 10^{-6}$, and we observed that this does not lead to a substantially different value of w. More precisely, the average L1 norm of the difference between the two solutions has mean 0.0022 and maximum 0.0047 (Supplementary Figure 25). We note that, although the quadratic approximation is only locally accurate, because the log-likelihood function is re-approximated at each step, w^* is guaranteed to optimize the function within a local neighborhood, that is, to be a local optimum. Furthermore, the parameter α encourages the algorithm to keep w close to the region where the local quadratic approximation is accurate. If α is too large, then convergence issues will be reached, but with $\alpha = 0.3$, we observed convergence of 100% of pixels in the Slide-seq cerebellum dataset.

In general, we do not have a guarantee that RCTD converges to the global minimum, since our likelihood function is nonconvex and could potentially have multiple local minima. While we cannot mathematically prove that RCTD obtains the global optimum, we obtained the following empirical evidence that RCTD does not get stuck in local minima. In particular, we have initialized the optimization at several randomly chosen initial values, and each one converged to the same local minima. On 50 randomly chosen pixels, we ran RCTD with 50 randomly chosen initial values, and RCTD always converged to the same solution, with maximum L1 norm error $4.0 * 10^{-6}$ (Supplementary Figure 25). The default initial condition is $w_{0,k} = 1/K$ for each cell type.

Returning to the computational aspects of implementing SQP, and recalling that $\lambda_j(w) = Nw^T \bar{\mu}_j$, we will derive an expression for the gradient and hessian of $-\mathcal{L}(w)$:

$$\begin{aligned} b(w) = -\nabla L(w) &= -\sum_{j=1}^{J} \nabla \log Q_{Y_j}(\lambda_j(w)) \\ &= -\sum_{j=1}^{J} \frac{Q'_{Y_j}(\lambda_j(w))}{Q_{Y_j}(\lambda_j(w))} N\bar{\mu}_j \end{aligned} \tag{33}$$

Next, the Hessian:

$$\begin{aligned} A(w) = \text{Hess}(-L(w)) &= -\sum_{j=1}^{J} \nabla(\frac{Q'_{Y_j}(\lambda_j(w))}{Q_{Y_j}(\lambda_j(w))}) \cdot N\bar{\mu}_j \\ &= -\sum_{j=1}^{J} (\frac{Q''_{Y_j}(\lambda_j(w))}{Q_{Y_j}(\lambda_j(w))} - (\frac{Q'_{Y_i}(\lambda_j(w))}{Q_{Y_j}(\lambda_j(w))})^2) \cdot (N\bar{\mu}_j)(N\bar{\mu}_j)^T \end{aligned} \tag{34}$$

Consequently, computation of the gradient and Hessian will depend on computing the first and second derivatives of $Q_\ell(\lambda)$. Notice that:

$$\begin{aligned} Q'_\ell(\lambda) &\equiv \int_{-\infty}^{\infty} p_\sigma(z) \frac{\partial}{\partial \lambda} (e^{-\lambda_i e^z} \frac{(e^z \lambda_i)^\ell}{\ell!}) \text{dz} \\ &= \int_{-\infty}^{\infty} p_\sigma(z) e^{-\lambda_i e^z} \frac{(e^z \lambda_i)^\ell}{\ell!} (-e^z + \frac{\ell}{\lambda}) \text{dz} \\ &= \frac{1}{\lambda}(-(\ell+1)Q_{\ell+1}(\lambda) + \ell Q_\ell(\lambda)) \end{aligned} \tag{35}$$

As such, we are able to express the derivatives of Q_ℓ in terms of Q_ℓ and $Q_{\ell+1}$! Likewise, we can compute:

$$\begin{aligned} Q''_\ell(\lambda) =& \frac{1}{\lambda}(-(\ell+1)Q'_{\ell+1}(\lambda) + \ell Q'_\ell(\lambda)) - \frac{1}{\lambda^2}(-(\ell+1)Q_{\ell+1}(\lambda) + \ell Q_\ell(\lambda)) \\ =& \frac{1}{\lambda^2}((\ell+1)(\ell+2)Q_{\ell+2}(\lambda) - 2\ell(\ell+1)Q_{\ell+1}(\lambda) + \ell(\ell-1)Q_\ell(\lambda)) \end{aligned} \tag{36}$$

Therefore, we have reduced the problems of computing the gradient and Hessian to the problem of computing $Q_\ell(\lambda)$ for all ℓ and λ. For each ℓ, we create a grid of potential λ values and compute $Q_\ell(\lambda)$. We store these results in a matrix, and for new λ values, we linearly interpolate to approximate $Q_\ell(\lambda)$.

Estimating σ

Finally, we address the issue of choosing σ (for both σ_γ and σ_ε) with the following procedure:

1. Initially set $\sigma = 1$.
2. Estimate the weights w for 500 pixels.
3. Compute the MLE σ^* of σ for these 500 samples given these values for w.

We note two differences in the estimation of σ_γ for the Platform Effect Normalization: there is only 1 bulk sample, so this is used, rather than 500 pixel, and more iterations over α_n are used below. The MLE for σ is found using stochastic gradient descent. We select σ to maximize the likelihood across all spots (with I samples):

$$\mathcal{L}(\sigma, w) = \sum_{i=1}^{I} \sum_{j=1}^{J} \log P(Y_{i,j} \mid \lambda_{i,j}) = \sum_{i=1}^{I} \sum_{j=1}^{J} \log Q_{Y_{i,j}}(\lambda_{i,j}) = \sum_{i,j} \log Q_{Y_{i,j}}(\lambda_{i,j}) \tag{37}$$

We alternatively maximizing $\mathcal{L}$ with respect to each of σ and w, as described above. This is guaranteed to converge to a local minimum. In practice, the choice of σ does not affect the maximum likelihood weights w too much, so we typically see convergence of σ in a couple iterations. The stochastic gradient descent algorithm proceeds as follows:

1. Set $\sigma = \sigma_0$ initial value.
2. Set $\alpha_n = \frac{.0001}{n}$, and repeat the following for $1 \leq n \leq N_{\text{iter}}$:

3. For each $1 \leq i \leq I, \quad 1 \leq j \leq J$, update:

$$\sigma \leftarrow \sigma + \alpha_n \frac{\partial}{\partial \sigma} \log Q_{Y_{i,j}}(\lambda_{i,j}) \tag{38}$$

The choice of α_n is to guarantee convergence to the global minimum since $\sum_n^\infty \alpha_n = \infty$, but $\sum_n^\infty \alpha_n^2 < \infty$. Traces of σ throughout this procedure imply convergence (Supplementary Figure 27). A typical choice for N_{iter} is 10. One remaining detail for this procedure is the calculation of $\frac{\partial}{\partial \sigma} \log Q_{Y_{i,j}}(\lambda_{i,j})$:

$$\begin{aligned} \frac{\partial}{\partial \sigma} \log Q_{Y_{i,j}}(\lambda_{i,j}) &= \frac{1}{Q_{Y_{i,j}}(\lambda_{i,j})} \frac{\partial}{\partial \sigma} Q_{Y_{i,j}}(\lambda_{i,j}) \\ &= \frac{1}{Q_{Y_{i,j}}(\lambda_{i,j})} \int_{-\infty}^{\infty} \frac{\partial p_\sigma(z)}{\partial \sigma} e^{-\lambda_{i,j} e^z} \frac{(e^z \lambda_{i,j})^{Y_{i,j}}}{(Y_{i,j})!} \mathrm{dz} \end{aligned} \tag{39}$$

Finally, we must calculate $\frac{\partial p_\sigma(z)}{\partial \sigma}$. We define:

$$p_\sigma(z) = \begin{cases} \frac{C}{\sqrt{2\pi}\sigma} e^{-\frac{z^2}{2\sigma^2}}, & |z| \leq 3\sigma \\ \frac{Ca}{\sigma(z/\sigma - c)^2}, & |z| > 3\sigma \end{cases} \tag{40}$$

Here, p_σ is a Normal Distribution, modified to be heavy tailed. The intuition here is that we want to allow for the (inevitable) possibility of outliers. Inverse square tails are almost as heavy tailed as possible for probability distributions. We choose c and a so that p_σ is continuously differentiable at the boundary $|z| = 3\sigma$. This results in $c = 7/3$ and $a = \frac{4}{9} e^{-9/2}/\sqrt{2\pi}$. C is a normalizing constant which is chosen to make p_σ integrate to 1. Note that these choices of constants hold for all σ. It follows that:

$$\frac{\partial p_\sigma(z)}{\partial \sigma} = \frac{-1}{\sigma} p_\sigma(z) + \begin{cases} \frac{z^2}{\sigma^3} p_\sigma(z) & |z| \leq 3\sigma \\ \frac{-2/\sigma^2}{z/\sigma - c} p_\sigma(z) & |z| > 3\sigma \end{cases} \tag{41}$$

Quantitative bound for deviation of sum of independent random variables

Here, we provide a quantitative bound for equation (17). Recall:

$$B_{k,j} = \frac{1}{I} \sum_{i=1}^{I} \frac{N_i}{\bar{N}} \beta_{k,i} \exp(\alpha_i + \varepsilon_{i,j}) \tag{42}$$

Recall we have independently lognormally distributed $e^{\varepsilon_{i,j}}$. For each j, $B_{k,j}$ is the sum of independent random variables. If we assume $N_i \alpha_i$ is bounded and that we have many samples I, then Chebyshev's inequality implies that for any $\delta > 0$:

$$\begin{aligned} P(|B_{k,j} - \mathbb{E}[B_{k,j}]| > \delta) &\leq \frac{\mathrm{Var}(B_{k,j})}{\delta^2} \\ &= \frac{1}{(I\bar{N}\delta)^2} \sum_{i=1}^{I} (N_i \beta_{k,i} e^{\alpha_i})^2 \mathrm{Var}(e^{\varepsilon_{i,j}}) \\ &\leq \frac{\max_{1 \leq i \leq I} (N_i e^{\alpha_i})^2 (e^{\sigma_\varepsilon^2}(e^{\sigma_\varepsilon^2} - 1))}{I\delta^2 \bar{N}^2}, \end{aligned} \tag{43}$$

where we have used the variance of a lognormal distribution. Therefore, with high probability, $\beta_{k,j}$ will not deviate far from its mean for large I.

Expected cell type-specific gene expression

Once β has been estimated, we probabilistically assign reads to cell types. This is accomplished by our model of sampling gene expression: for each pixel $1 \leq i \leq I$ and for each read $1 \leq n \leq N_i$, we select cell type $1 \leq \theta_{n,i} \leq K$ and gene $1 \leq z_{n,i} \leq J$ as:

$$P(\theta_{n,i} = k \mid \beta) = \beta_{k,i}, \quad P(z_{n,i} = j \mid \theta_{n,i}, \gamma, \varepsilon) = \delta_{i,\theta_{n,i},j} \tag{44}$$

Here, we define:

$$\log(\delta_{i,k,j}) = \alpha_i + \log(\hat{\mu}_{k,j}) + \gamma_j + \varepsilon_{i,j} \tag{45}$$

We can use Bayes' Theorem to calculate the probability that each read n of gene i came from cell type j in sample k. We assume that the reads are conditionally independent given $\beta, \gamma, \varepsilon$:

$$\begin{aligned} P(\theta_{n,i} = k \mid \beta, z_{n,i} = j, \gamma, \varepsilon) &\propto P(\theta_{n,i} = k, z_{n,i} = j \mid \beta, \gamma, \varepsilon) \\ &= P(\theta_{n,i} = k \mid \beta) P(z_{n,i} = j \mid \beta, \theta_{n,i} = k, \gamma, \varepsilon) \\ &= \beta_{k,i} \delta_{i,k,j} \end{aligned} \tag{46}$$

This implies that:

$$P(\theta_{n,i} = k \mid \beta, z_{n,i} = j, \gamma, \varepsilon) = \frac{\beta_{k,i}\delta_{i,k,j}}{\sum_{k'=1}^{K} \beta_{k',i}\delta_{i,k',j}} = \frac{\beta_{k,i}\hat{\mu}_{k,j}}{\sum_{k'=1}^{K} \beta_{k',i}\hat{\mu}_{k',j}} \tag{47}$$

This conditional probability does not depend on γ, ε, since the ratio (across k) of $\delta_{i,k,j}$ is fixed. This implies (by tower law):

$$P(\theta_{n,i} = k \mid \beta, z_{n,i} = j) = \mathbb{E}_{\gamma,\varepsilon}[P(\theta_{n,i} = k \mid \beta, z_{n,i} = j, \gamma, \varepsilon)] = \frac{\beta_{k,i}\hat{\mu}_{k,j}}{\sum_{k'=1}^{K} \beta_{k',i}\hat{\mu}_{k',j}} \tag{48}$$

Therefore, we can use Equation (48) to calculate the probability that each UMI came from each cell type given $\beta_{k,i}$. Intuitively, this probability is a competition between different cell types and is proportional to (1) the proportion of the cell type on a spot and (2) the probability of observing the gene in the cell type for a single UMI. Consequentially, conditional on the observed gene counts, we can calculate in pixel i the expectation of gene j originating from each cell type k, $Y_{i,k,j}$, as proportional to the observed gene counts $Y_{i,j}$:

$$\mathbb{E}[Y_{i,k,j} \mid \beta, Y_{i,j}] = \mathbb{E}[\sum_{i=1}^{I} \mathbb{I}[\theta_{n,i} = k, z_{n,i} = j] \mid \beta, Y_{i,k}] = \frac{Y_{i,j}\beta_{k,i}\hat{\mu}_{k,j}}{\sum_{k'=1}^{K} \beta_{k',i}\hat{\mu}_{k',j}} \tag{49}$$

Cell type identification by model selection

Although we show that RCTD's default choice of $\delta = 10$ leads to high accuracy across several datasets, we offer the following guidance to the user in choosing a value of δ. In general, the user may plot the results of RCTD using multiple values of δ and observe the qualitative effect on accuracy. In particular, if $\delta = 10$ yields accurate results, but predicts few pixels confidently, decreasing δ may be considered. If $\delta = 10$ yields inaccurate results, the user can consider increasing δ. Accuracy can be heuristically measured using prior spatial knowledge or comparison of cell type predictions to marker genes. Because it is in general difficult to measure classification accuracy without ground truth cell type labels, we do not have a quantitative procedure for selecting the optimal δ for achieving a target classification accuracy.

Modes of RCTD

RCTD with doublet mode constrains each pixel to at most two cell types. RCTD can also be run without constraining the number of cell types per pixel. RCTD with doublet mode has been extended to *multi mode*, which predicts more than two cell types by proceed by greedy search: at each iteration, the next cell type is added that will best improve the model fit until diminishing returns are reached. Confidence is determined for each cell type by testing if there is an alternative cell type that will yield a similar model fit.

To quantitatively define *multi mode* for cell types $c_1, c_2, \ldots, c_p \in \{1, 2, \ldots, K\}$, let $\mathcal{L}(c_1, c_2, \ldots, c_p)$ be the log-likelihood of the model fit with these cell types on a particular pixel. After selecting p cell types $(c_1, c_2, \ldots, c_p)$, in order to choose an additional cell type by greedy search, we select the next cell type maximizing log-likelihood:

$$c_{p+1} \equiv \text{argmax}_c \left(\mathcal{L}(c_1, c_2, \ldots, c_p, c)\right). \tag{50}$$

Because we expect many pixels to contain few cell types, we then used a penalized approach similar to AIC [6] to decide the model $\mathcal{M}_p$ containing the best number of cell types p. $\mathcal{M}_p$ is defined as the model that cell types $c_1, c_2, \ldots, c_p$ are present on the pixel. We select p as maximizing,

$$\text{AIC}(\mathcal{M}_p) \equiv \mathcal{L}(\mathcal{M}_p) - Vp = \mathcal{L}(c_1, c_2, \ldots, c_p) - Vp,$$

with p the number of parameters (cell types) and V a penalty weight. In the results presented here, we selected $V = 25$ based on simulation studies.

After selecting cell types, final cell type weights are estimated. To predict, for each Slide-seq pixel, the number of cell types present (Supplementary Figure 5), we ran RCTD on multi mode on the cerebellum Slide-seq dataset.

Differing cell types between spatial data and reference

To measure RCTD's performance on pixels containing cell types other than those appearing in the reference, we focused on cross-platform prediction of single cells. We executed this by using the same 12 cell types in the scRNA-seq dataset, but we removed several cell types from the reference training data. Recall that for the cerebellum dataset, cell classes are defined as {MLI1, MLI2}, {Polydenrocytes, Oligodendrocytes}, {Bergmann, Astrocytes}, and {Fibroblast, Endothelial}. First, we removed one cell type from each class: MLI2, Polydendrocytes, Bergmann, and Fibroblast. We observed that RCTD most often classified each missing cell type as the present cell type within the class (Supplementary Figure 11). Next, we tested RCTD by removing two cell types from each class, and measuring RCTD's performance on predicting those cell types. This was done for each of {MLI1, MLI2}, {Polydenrocytes, Oligodendrocytes}, {Bergmann, Astrocytes}, and {Fibroblast, Endothelial}.

Supplementary Table 1: Estimated Parameters

Target Dataset	σ_γ	σ_ε
snRNA-seq Cerebellum	0	0.77
scRNA-seq Cerebellum	1.28	0.88
Slide-seq Cerebellum from snRNA-seq	0.94	0.95
Slide-seq Hippocampus	0.51	0.80
Visium Hippocampus	0.70	0.45
Slide-seq Cerebellum from scRNA-seq	0.61	0.76
Slide-seq Somatosensory Cortex (from Smart-seq2)	1.36	0.52

Supplementary Experimental Methods

Animal Handling

All procedures involving animals at the Broad Institute were conducted in accordance with the US National Institutes of Health Guide for the Care and Use of Laboratory Animals under protocol number 0120-09-16.

Transcardial Perfusion

C57BL/6J mice were anesthetized by administration of isoflurane in a gas chamber flowing 3% isoflurane for 1 minute. Anesthesia was confirmed by checking for a negative tail pinch response. Animals were moved to a dissection tray and anesthesia was prolonged via a nose cone flowing 3% isoflurane for the duration of the procedure. Transcardial perfusions were performed with ice cold pH 7.4 HEPES buffer containing 110 mM NaCl, 10 mM HEPES, 25 mM glucose, 75 mM sucrose, 7.5 mM MgCl2, and 2.5 mM KCl to remove blood from brain and other organs sampled. The appropriate organs were removed and frozen for 3 minutes in liquid nitrogen vapor and moved to -80C for long term storage.

Tissue Handling

Fresh frozen tissue was warmed to -20 C in a cryostat (Leica CM3050S) for 20 minutes prior to handling. Tissue was then mounted onto a cutting block with OCT and sliced at a 5° cutting angle at 10 µm thickness. Pucks were then placed on the cutting stage and tissue was maneuvered onto the pucks. The tissue was then melted onto the puck by moving the puck off the stage and placing a finger on the bottom side of the glass. The puck was then removed from the cryostat and placed into a 1.5 mL eppendorf tube. The sample library was then prepared as below. The remaining tissue was re-deposited at -80 C and stored for processing at a later date.

Puck preparation and sequencing

Pucks were prepared as described recently using barcoded beads synthesized in-house on an Akta Oligopilot 10 according to the updated Slide-seqV2 protocol [2]. Pucks were sequenced using a monobase-encoding sequencing-by-ligation approach also described in the updated protocol. We used slide-seq tools for alignment and processing of Slide-seq data.

Pucks were generated using one of two separate bead batches with the oligo sequences listed below:

Batch 1:

5'-
TTT_PC_GCCGGTAATACGACTCACTATAGGGCTACACGACGCTCTTCCGATCTJJJJJJJJTCTTCAGCGTTCCCGAGAJ
JJJJJJTCNNNNNNNNT25

Batch 2:

5'-
TTT_PC_GCCGGTAATACGACTCACTATAGGGCTACACGACGCTCTTCCGATCTJJJJJJJJTCTTCAGCGTTCCCGAGAJ
JJJJJNNNNNNNVVT30

"PC" designates a photocleavable linker; "J" represents bases generated by split-pool barcoding, such that every oligo on a given bead has the same J bases; "N" represents bases generated by mixing, so every oligo on a given bead has different N bases; and "TX" represents a sequence of X thymidines. "V" represents bases which may contain A, C, G but not T.

Slide-seqV2 library preparation

RNA Hybridization:
Pucks in 1.5 mL tubes were immersed in 200 µL of hybridization buffer (6x SSC with 2 U/µL Lucigen NxGen RNAse inhibitor) for 15 minutes at room temperature to allow for binding of the RNA to the oligos on the beads.

First Strand Synthesis
Subsequently, first strand synthesis was performed by incubating the pucks in RT solution for 30 minutes at room temperature followed by 1.5 hours at 52 °C.

RT solution:
115 µL H2O
40 µL Maxima 5x RT Buffer (Thermofisher, EP0751**)**
20 µL 10 mM dNTPs (NEB N0477L)
5 µL RNase Inhibitor (Lucigen 30281)
10 µL 50 µM Template Switch Oligo **(**Qiagen #339414YCO0076714)
10 µL Maxima H- RTase (Thermofisher, EP0751)

Tissue Digestion:
200 µL of 2x tissue digestion buffer was then added directly to the RT solution and the mixture was incubated at 37 °C for 30 minutes.

2x tissue digestion buffer:
200 mM Tris-Cl pH 8
400 mM NaCl
4% SDS
10 mM EDTA
32 U/mL Proteinase K (NEB P8107S)

Second Strand Synthesis:
The solution was then pipetted up and down vigorously to remove beads from the surface, and the glass substrate was removed from the tube using forceps and discarded. 200 µL of Wash Buffer was then added to the 400 µL of tissue clearing and RT solution mix and the tube was then centrifuged for 2 minutes at 3000 RCF. The supernatant was then removed from the bead pellet, the beads were resuspended in 200 µL of Wash Buffer, and were centrifuged again. This was repeated a total of three

times. The supernatant was then removed from the pellet. The beads were then resuspended in 200 µL of ExoI mix and incubated at 37 °C for 50 minutes.

Wash Buffer:
10 mM Tris pH 8.0
1 mM EDTA
0.01% Tween-20

ExoI mix:
170 µL H20
20 µL ExoI buffer
10 µL ExoI (NEB M0568)

After ExoI treatment the beads were centrifuged for 2 minutes at 3000 RCF. The supernatant was then removed from the bead pellet, the beads were resuspended in 200 µL of Wash Buffer, and were centrifuged again. This was repeated a total of three times. The supernatant was then removed from the pellet. The pellet was then resuspended in 200 µL of 0.1 N NaOH and incubated for 5 minutes at room temperature. To quench the reaction, 200 µL of Wash Buffer was added and beads were centrifuged for 2 minutes at 3000 RCF. The supernatant was then removed from the bead pellet, the beads were resuspended in 200 µL of Wash Buffer, and were centrifuged again. This was repeated a total of three times. Second Strand Synthesis was then performed on the beads by incubating the pellet in 200 µL of Second Strand Mix at 37 °C for 1 hour.

Second Strand Synthesis mix:
133 µL H2O
40 µL Maxima 5x RT Buffer
20 µL 10 mM dNTPs
2 µL 1 mM dN-SMRT oligo
5 µL Klenow Enzyme (NEB M0210)

After Second Strand Synthesis, 200 µL of Wash Buffer was added and the beads were centrifuged for 2 minutes at 3000 RCF. The supernatant was then removed from the bead pellet, the beads were resuspended in 200 µL of Wash Buffer, and were centrifuged again. This was repeated a total of three times.

Library Amplification:
200 µL of water was then added to the bead pellet and the beads were centrifuged for 2 minutes at 3000 RCF. The supernatant was then removed from the bead pellet and the beads were resuspended in 50 µL of library PCR mix and moved into a 200 µL PCR strip tube. PCR was then performed as outlined below:

Library PCR mix:
22 µL H2O
25 µL of Terra Direct PCR mix Buffer (Takara Biosciences 639270)
1 µL of Terra Polymerase (Takara Biosciences 639270)
1 µL of 100 µM Truseq PCR primer (IDT)
1 µL of 100 µM SMART PCR primer (IDT)

PCR program:
95 °C 3 minutes

4 cycles of:
98 °C 20 seconds
65 °C 45 seconds
72 °C 3 minutes

9 cycles of:
98 °C 20 seconds
67 °C 20 seconds
72 °C 3 minutes

Then:
72 °C 5 minutes
Hold at 4 °C

PCR cleanup and Nextera Tagmentation:

Samples were cleaned with Ampure XP (Beckman Coulter A63880) beads in accordance with manufacturer's instructions at a 0.6x bead/sample ratio (30 µL of beads to 50 µL of sample) and resuspended in 50 µL of water. The cleanup procedure was repeated, this time resuspending in a final volume of 10 µL. 1 µL of the library was quantified on an Agilent Bioanalyzer High sensitivity DNA chip (Agilent 5067-4626). Then, 600 pg of cDNA was taken from the PCR product and prepared into Illumina sequencing libraries through tagmentation using the Nextera XT kit (Illumina FC-131-1096). Tagmentation was performed according to manufacturer's instructions and the library was amplified with primers Truseq5 and N700 series barcoded index primers. The PCR program was as follows:

PCR program:
72 °C for 3 minutes
95 °C for 30 seconds

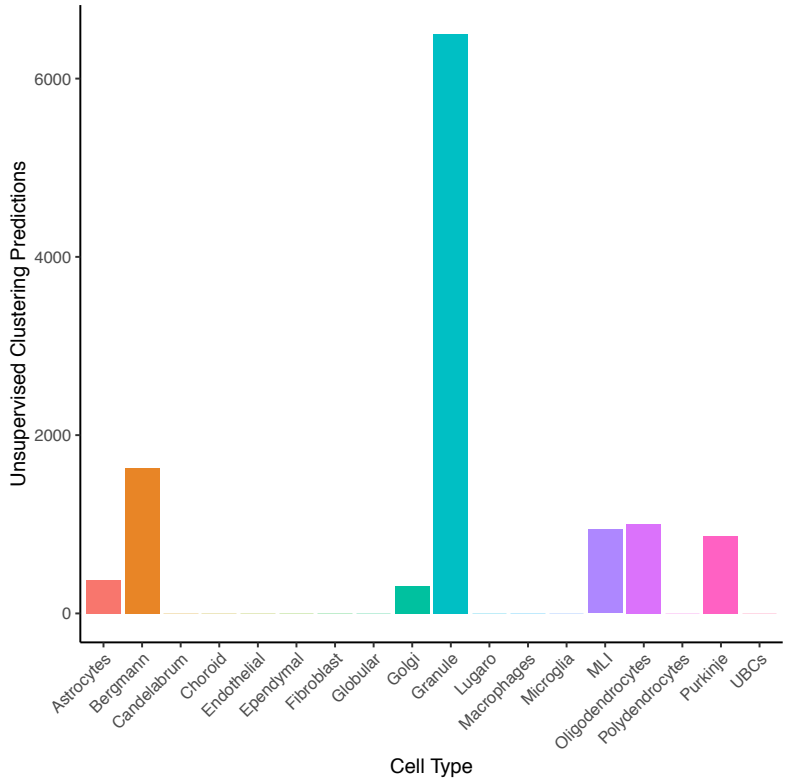
12 cycles of:
95 °C for 10 seconds
55 °C for 30 seconds
72 °C for 30 seconds

72 °C for 5 minutes
Hold at 4 °C

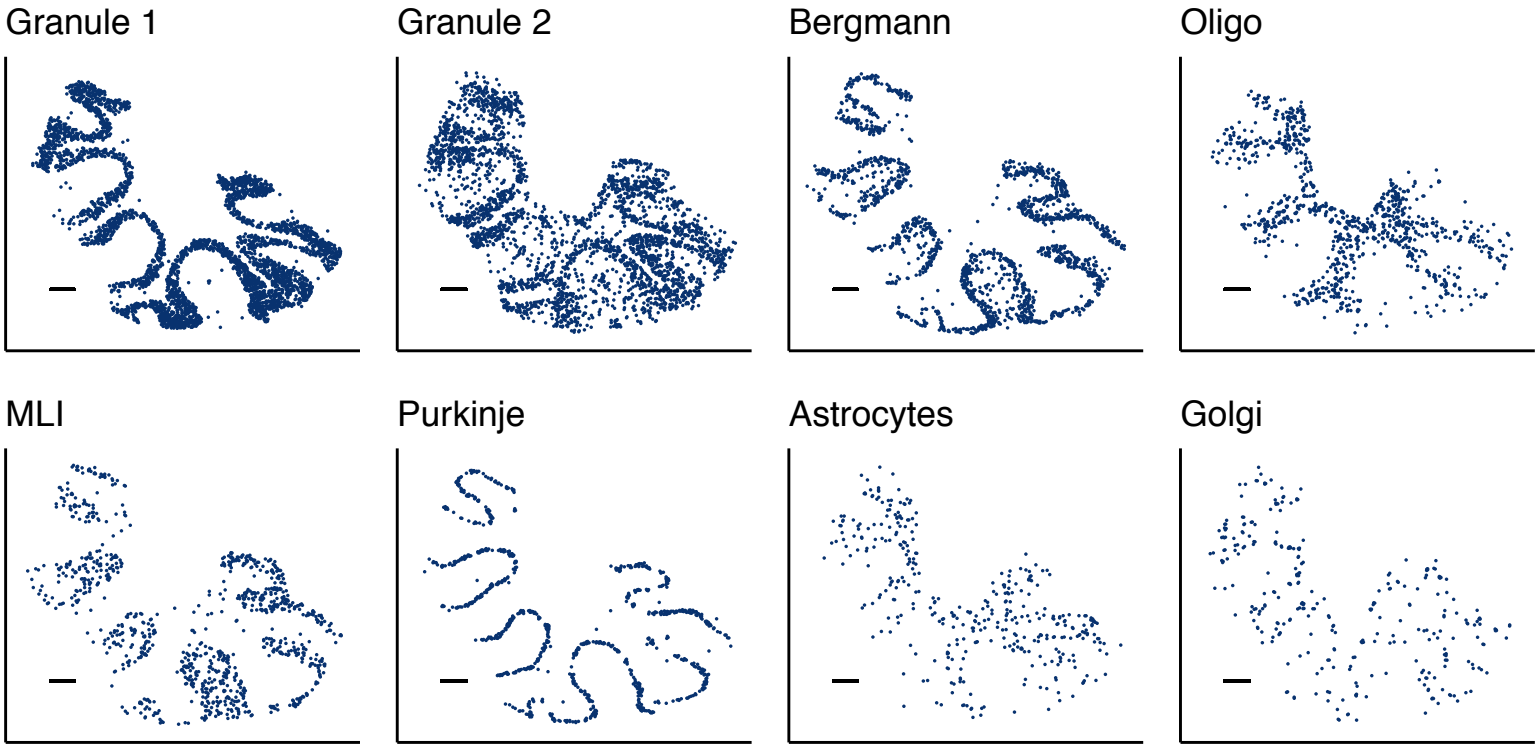
Samples were cleaned with Ampure XP (Beckman Coulter A63880) beads in accordance with manufacturer's instructions at a 0.6x bead/sample ratio (30 µL of beads to 50 µL of sample) and resuspended in 10 µL of water. 1 µL of the library was quantified on an Agilent Bioanalyzer High sensitivity DNA chip (Agilent 5067-4626). Finally, the library concentration was normalized to 4 nM for sequencing. Samples were sequenced on the Illumina NovaSeq S2 flowcell 100 cycle kit with 12 samples per run (6 samples per lane) with the read structure 44 bases Read 1, 8 bases i7 index read, 50 bases Read 2. Each puck received approximately 200-400 million reads, corresponding to 3,000-5,000 reads per bead.

Supplementary Figures

a)


Unsupervised Clustering Predictions
6000
4000
2000
0
Astrocytes
Bergmann
Candelabrum
Choroid
Endothelial
Ependymal
Fibroblast
Globular
Golgi
Granule
Lugaro
Macrophages
Microglia
MLI
Oligodendrocytes
Polydendrocytes
Purkinje
UBCs
Cell Type

b)

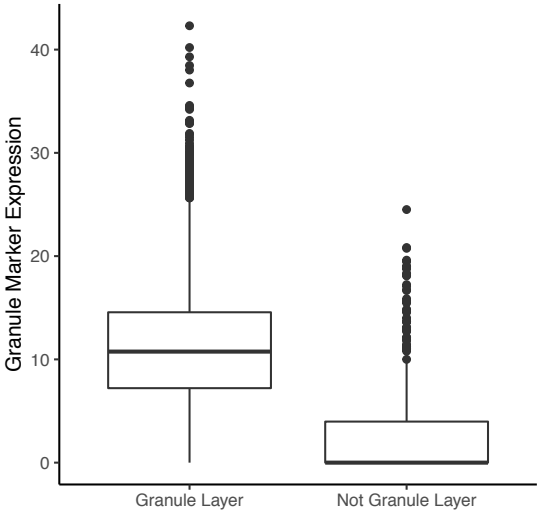

Granule 1
Granule 2
Bergmann
Oligo
MLI
Purkinje
Astrocytes
Golgi

Supplementary Figure 1: Predicted spatial localization of cell types by unsupervised clustering in Slide-seq cerebellum.

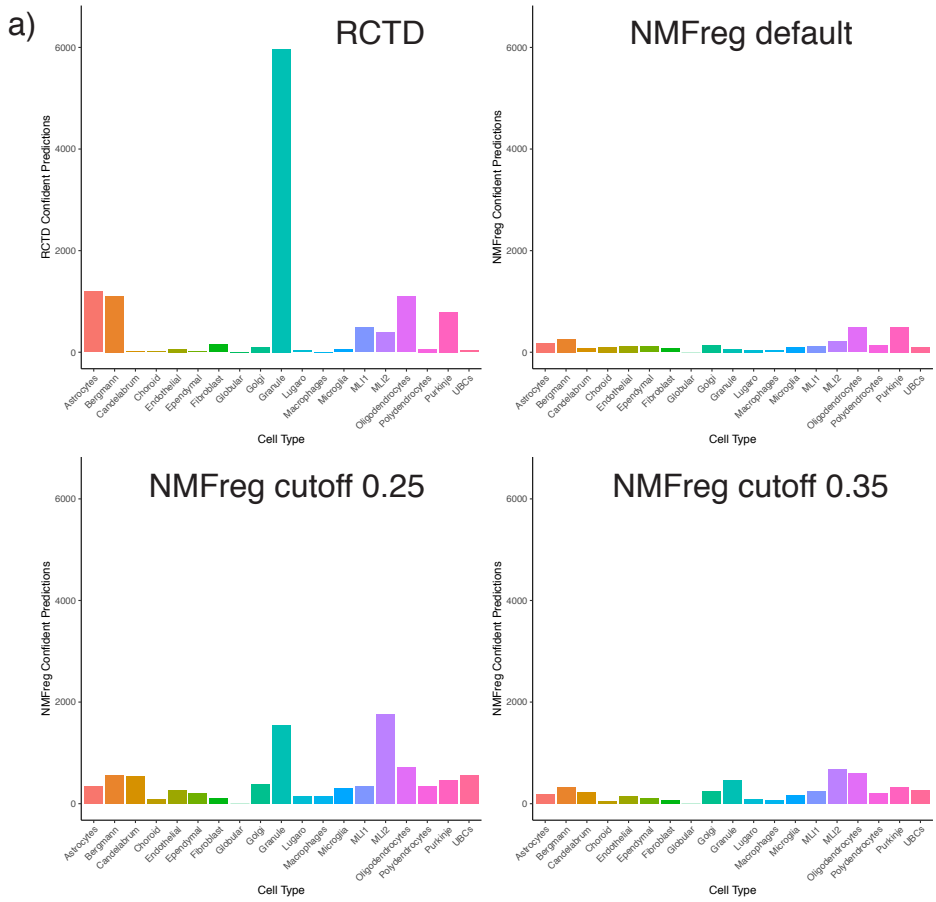
a) Barplot of counts for predicted pixels assigned to each cell type. Calls obtained unsupervised clustering. Note that there are two granule clusters, and only 1 cluster for both MLI1 and MLI2.

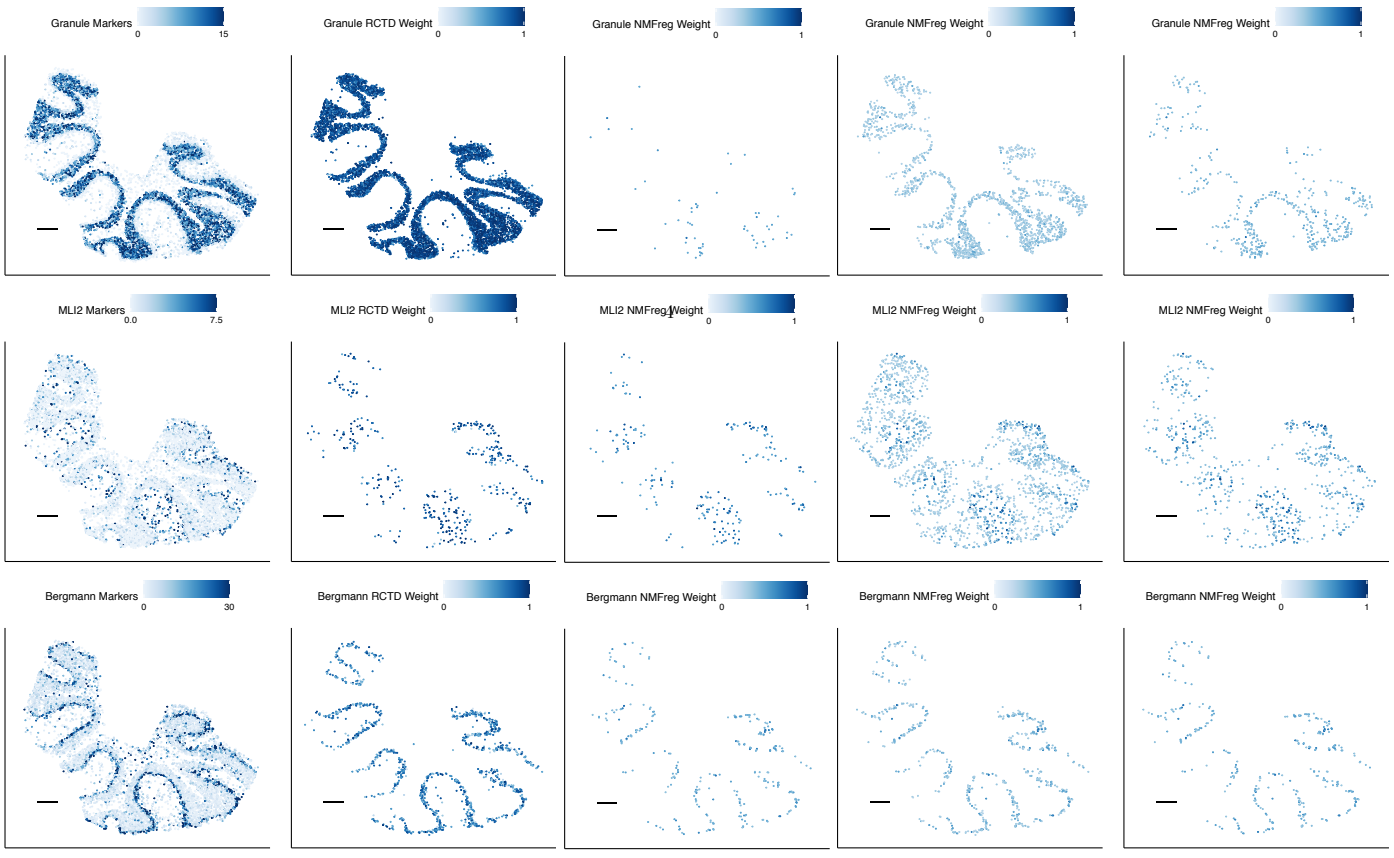
b) Predicted spatial locations of each cell type by unsupervised clustering.

All scale bars 250 microns.


Granule Marker Expression
40
30
20
10
0
Granule Layer
Not Granule Layer

Supplementary Figure 2: Expression of Granule Markers within Unsupervised Clustering Classifications
Boxplot of granule marker expression within pixels classified as granule by unsupervised clustering. Pixels were split into pixels within and outside the granule layer.


a)
RCTD
NMFreg default
NMFreg cutoff 0.25
NMFreg cutoff 0.35
RCTD Confident Predictions
NMFreg Confident Predictions
Cell Type
Astrocytes
Bergmann
Candelabrum
Choroid
Endothelial
Ependymal
Fibroblast
Globular
Golgi
Granule
Lugaro
Macrophages
Microglia
MLI1
MLI2
Oligodendrocytes
Polydendrocytes
Purkinje
UBCs

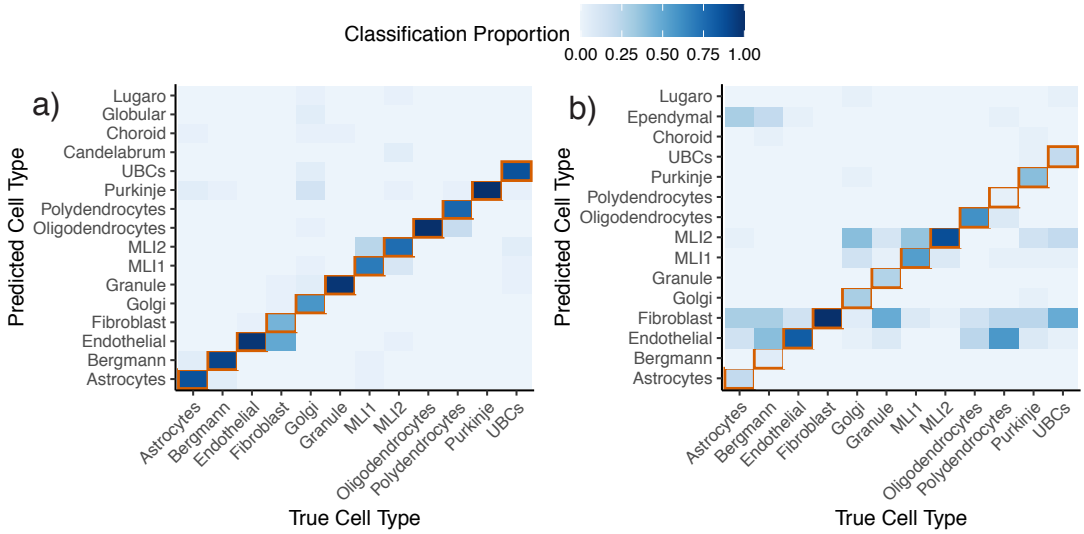

b)
Markers
RCTD
NMFreg default
NMFreg cutoff 0.25
NMFreg cutoff 0.35
Granule Markers
Granule RCTD Weight
Granule NMFreg Weight
MLI2 Markers
MLI2 RCTD Weight
MLI2 NMFreg Weight
Bergmann Markers
Bergmann RCTD Weight
Bergmann NMFreg Weight

Supplementary Figure 3: Comparison of NMFreg and RCTD.

a) Barplot of counts for confidently predicted pixels assigned to each cell type. Calls obtained by RCTD (top left) or NMFreg (other three panels). NMFreg confidence was determined either by default (top right), or by a constant proportion cutoff of 0.25 (bottom left) or 0.35 (bottom right).

b) Confidently predicted spatial localization of cell types by RCTD and NMFreg for granule, molecular layer interneurons 2 (MLI2), and Bergmann. Left: expression (counts per 500) (represented by color) of marker genes. Middle: predicted spatial locations of a cell type by RCTD, with color representing predicted cell type proportion. Column 2: predicted spatial locations of a cell type by NMFreg, with color representing predicted cell type proportion. NMFreg underpredicted granule cells in the granular layer and instead incorrectly overpredicted MLI2 in this layer [8]. NMFreg confidence was determined either by default (column 3), or by a constant proportion cutoff of 0.25 (column 4) or 0.35 (column 5).

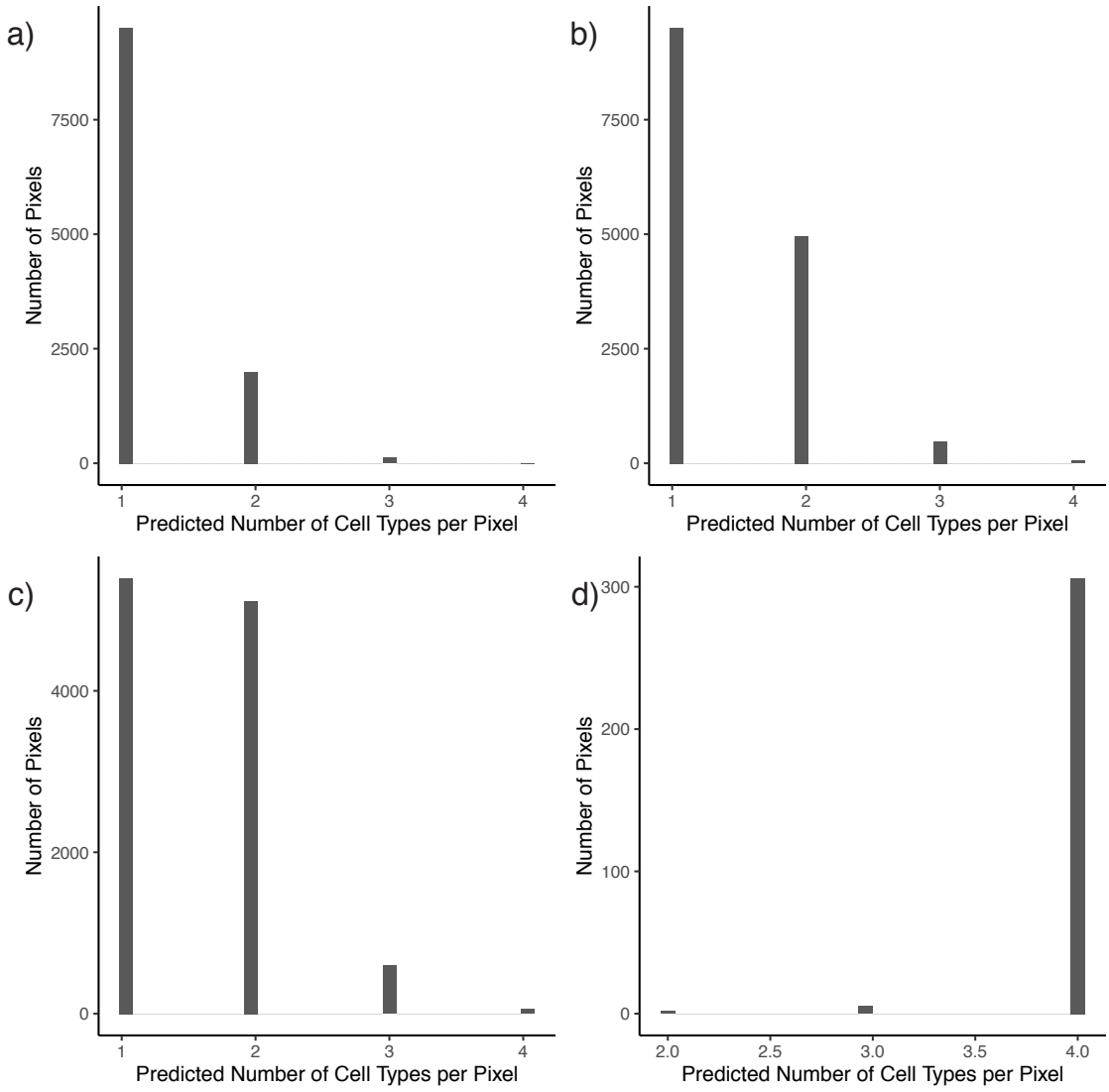
All scale bars 250 microns.


Classification Proportion
0.00 0.25 0.50 0.75 1.00
a)
b)
Predicted Cell Type
True Cell Type
Lugaro
Globular
Choroid
Candelabrum
UBCs
Purkinje
Polydendrocytes
Oligodendrocytes
MLI2
MLI1
Granule
Golgi
Fibroblast
Endothelial
Bergmann
Astrocytes
Ependymal

Supplementary Figure 4: Performance of DWLS on cell type classification.
All panels: confusion matrix of DWLS cell type predictions, as compared to ground truth labels.

a) DWLS was trained and tested on the single-nucleus cerebellum reference.

b) DWLS was trained on single-nucleus and tested on single-cell.


a)
b)
c)
d)
Number of Pixels
Predicted Number of Cell Types per Pixel
0
2500
5000
7500
2000
4000
100
200
300
1
2
3
4
2.0
2.5
3.0
3.5
4.0

Supplementary Figure 5: Number of cell types per Slide-seq cerebellum pixel.

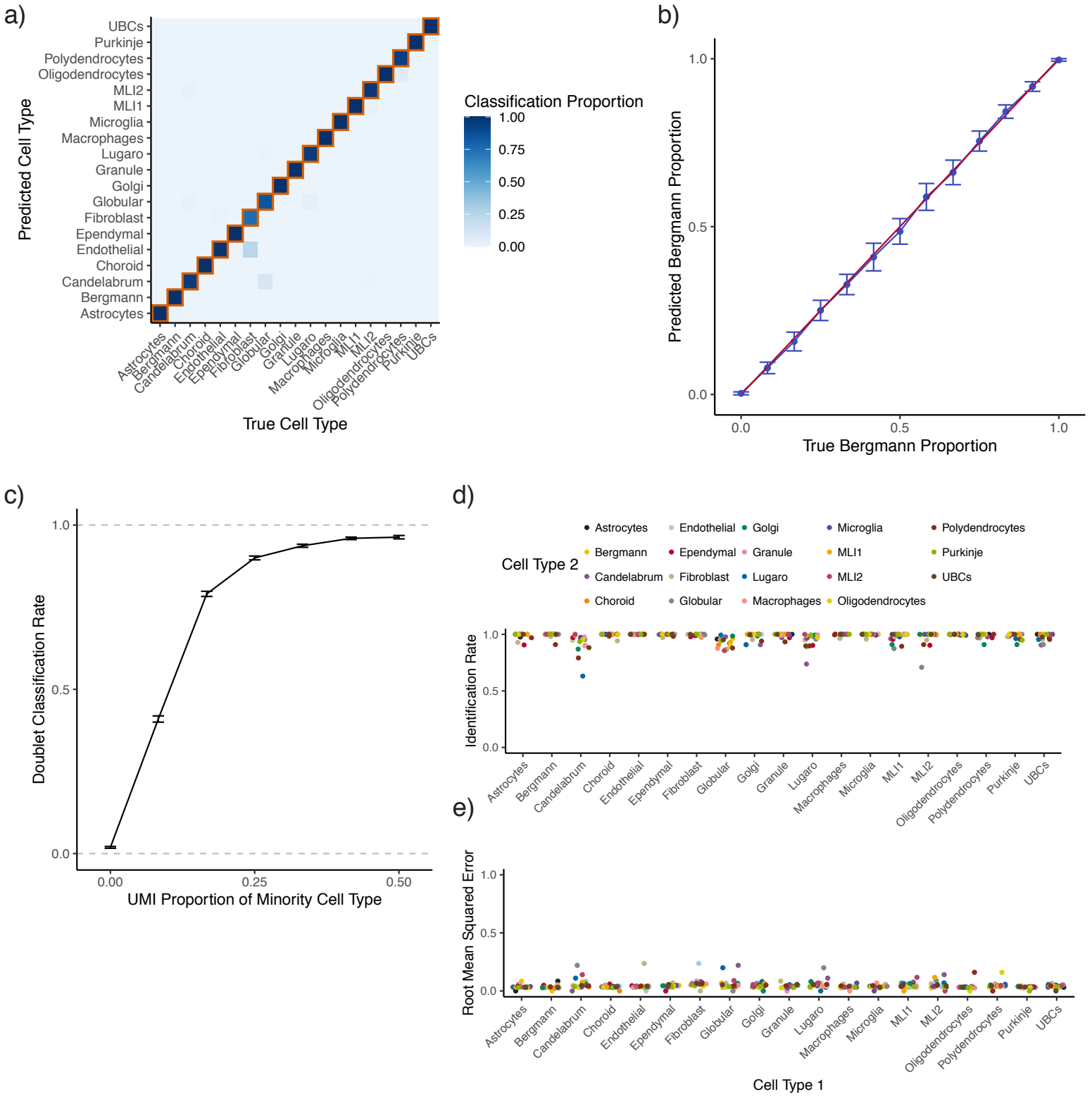
Histogram of number of cell types per spatial transcriptomic pixel, estimated by RCTD with multi mode.

a) Slide-seq cerebellum. In total, 1.1% of pixels were predicted to contain more than two cell types, consistent with previous estimates of Slide-seq data [9].

b) Slide-seq hippocampus. In total, 3.6% of pixels were predicted to contain more than two cell types.

c) Slide-seq somatosensory cortex. In total, 5.8% of pixels were predicted to contain more than two cell types.

d) Visium hippocampus. In total, 99.3% of pixels were predicted to contain more than two cell types.


a)
Predicted Cell Type
True Cell Type
Classification Proportion
1.00
0.75
0.50
0.25
0.00
UBCs
Purkinje
Polydendrocytes
Oligodendrocytes
MLI2
MLI1
Microglia
Macrophages
Lugaro
Granule
Golgi
Globular
Fibroblast
Ependymal
Endothelial
Choroid
Candelabrum
Bergmann
Astrocytes
b)
Predicted Bergmann Proportion
True Bergmann Proportion
1.0
0.5
0.0
c)
Doublet Classification Rate
UMI Proportion of Minority Cell Type
0.00
0.25
0.50
d)
Cell Type 2
Identification Rate
e)
Root Mean Squared Error
Cell Type 1

Supplementary Figure 6: Robust Cell Type Decomposition (RCTD) accurately decomposes mixtures of cells.

All: RCTD was trained on the single-nucleus RNA-seq cerebellum dataset and tested on a the same dataset.

a) Confusion matrix for RCTD on within-reference cell type assignment for single cells.

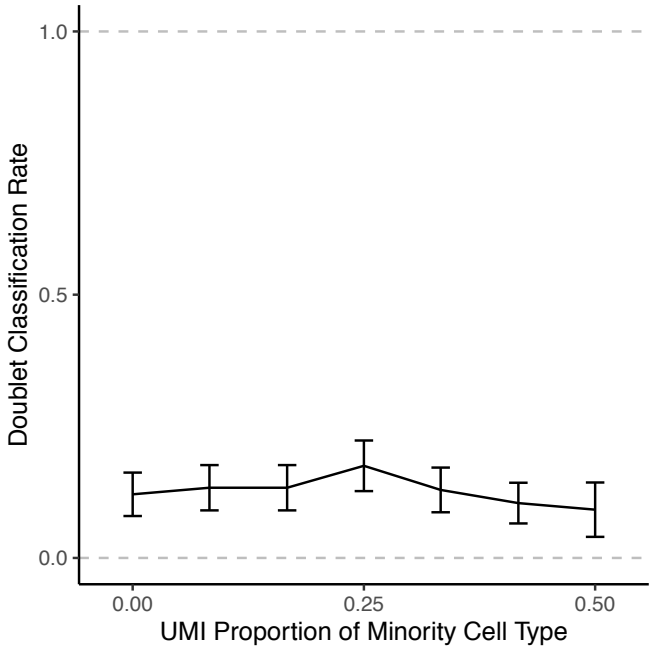
b) Rate of doublet classification of simulated mixtures of single cells, with 95% confidence intervals. The x-axis represents the proportion of UMIs sourced from the minority cell type, ranging from 0% (true singlet) to 50% (full doublet) (n between 5130 and 10260 simulations per condition).

c) Predicted Bergmann proportion as a function of true Bergmann proportion on simulated Bergmann-Purkinje doublets. The red line is the identity line, and the blue line is the average and standard deviation ($n = 30$ simulations per condition) of prediction.

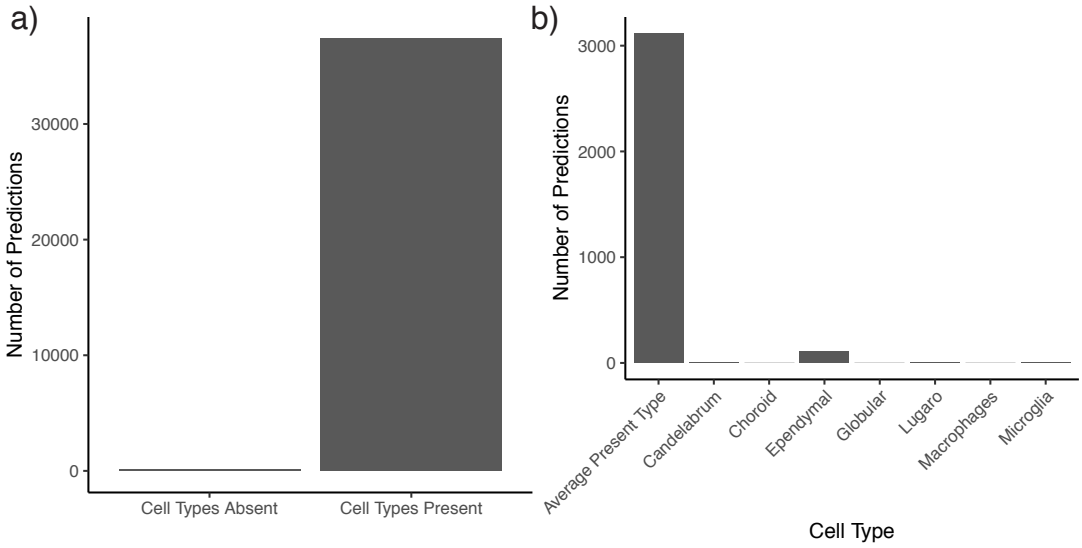
d) Metrics for performance of RCTD on simulated doublets ($n = 390$ simulations per cell type pair). Column represents cell type 1, and color represents cell type 2. Top: RCTD assigns doublets to two cell types. Identification rate is the percentage of confident calls correctly identifying cell class 1.

e) Bottom: RMSE of predicted vs true cell type proportion (as in (c)).

All scale bars 250 microns.


1.0
0.5
0.0
Doublet Classification Rate
0.00
0.25
0.50
UMI Proportion of Minority Cell Type

Supplementary Figure 7: Doublet classification of pixels with two transcriptionally similar cell types. Rate of doublet classification by RCTD on simulated mixtures of single cells, with 95% confidence intervals. The x-axis represents the true proportion of UMIs sampled from the minority cell type, ranging from 0% (true singlet) to 50% (equal proportion doublet). RCTD was trained on the single-nucleus RNA-seq cerebellum dataset and tested on a dataset of simulated mixtures of single cells from a single-cell RNA-seq cerebellum dataset. We restricted the calculations to pixels containing two cell types of the same cell type class, defined as {Astrocytes, Bergmann}, {Endothelial, Fibroblast}, {MLI1, MLI2}, and {Oligodendrocytes, Polydendrocytes}.

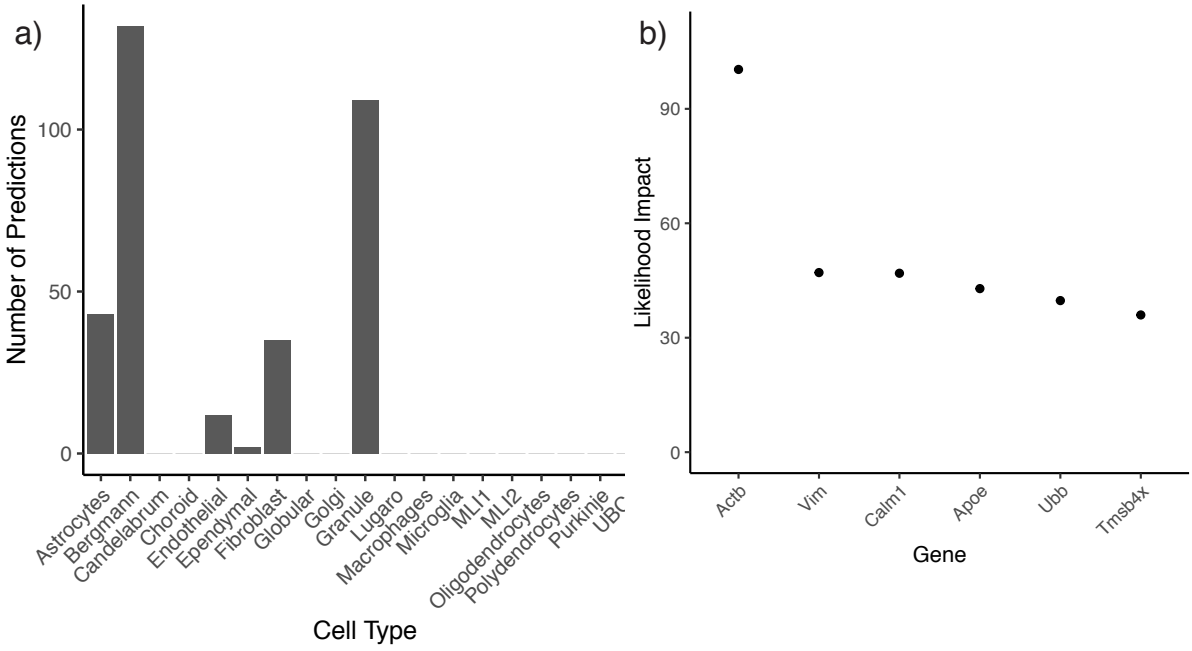

a)
Number of Predictions
30000
20000
10000
0
Cell Types Absent
Cell Types Present
b)
Number of Predictions
3000
2000
1000
0
Average Present Type
Candelabrum
Choroid
Ependymal
Globular
Lugaro
Macrophages
Microglia
Cell Type

Supplementary Figure 8: Performance of RCTD on Cell Types that do not appear in the Spatial Data

a) Barplot of confident cell type prediction. "Cell Types Absent" represent cell types appearing in the reference but not the spatial data, whereas "Cell Types Present" represent cell types appearing in the spatial data and reference.

b) The same as (a), but broken down by cell type not appearing in the spatial data. For comparison "Average Present Type" is the average number of predictions of a cell type appearing in the spatial data.

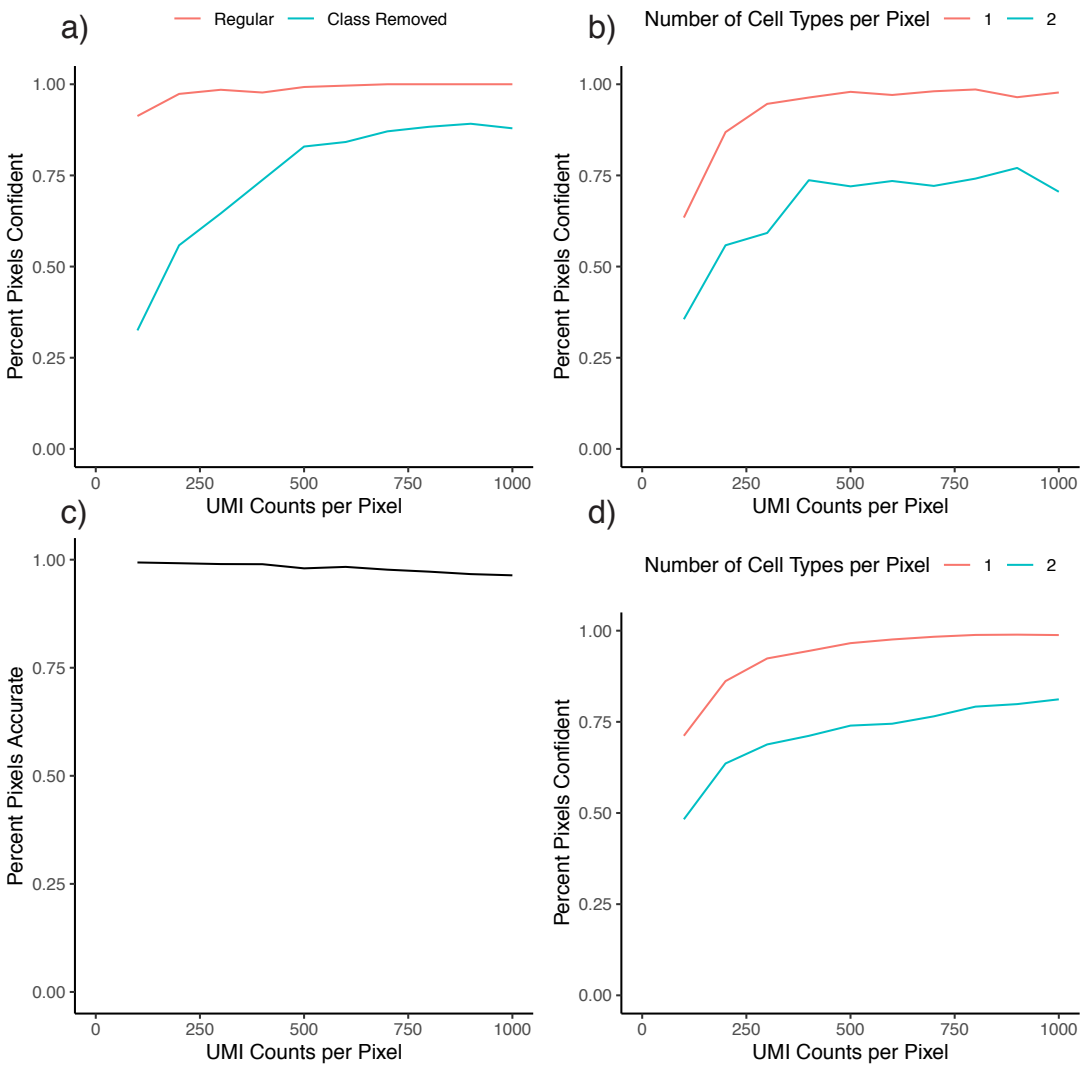
All analysis occurs on the simulated cerebellum doublets dataset (trained on snRNA-seq, tested on scRNA-seq).


a)
Number of Predictions
100
50
0
Astrocytes
Bergmann
Candelabrum
Choroid
Endothelial
Ependymal
Fibroblast
Globular
Golgi
Granule
Lugaro
Macrophages
Microglia
MLI1
MLI2
Oligodendrocytes
Polydendrocytes
Purkinje
UBC
Cell Type
b)
Likelihood Impact
90
60
30
0
Actb
Vim
Calm1
Apoe
Ubb
Tmsb4x
Gene

Supplementary Figure 9: Misclassification errors of RCTD on true Bergmann-Granule doublet pixels. All panels: RCTD was trained on single-nucleus cerebellum and tested on single-cell simulated doublets. Analysis constrained to doublets containing a Bergmann cell and a granule cell.

a) Histogram of cell type predictions made by RCTD.

b) The most frequent misclassification in (a) was Fibroblast. The genes with the largest impact on likelihood for these pixels, favoring Fibroblast over granule. "Likelihood Impact" is defined as the difference in the log-likelihood for the gene between the Fibroblast and granule models.


a)
Regular
Class Removed
Percent Pixels Confident
1.00
0.75
0.50
0.25
0.00
0
250
500
750
1000
UMI Counts per Pixel
b)
Number of Cell Types per Pixel
1
2
Percent Pixels Confident
UMI Counts per Pixel
c)
Percent Pixels Accurate
UMI Counts per Pixel
d)
Number of Cell Types per Pixel
1
2
Percent Pixels Confident
UMI Counts per Pixel

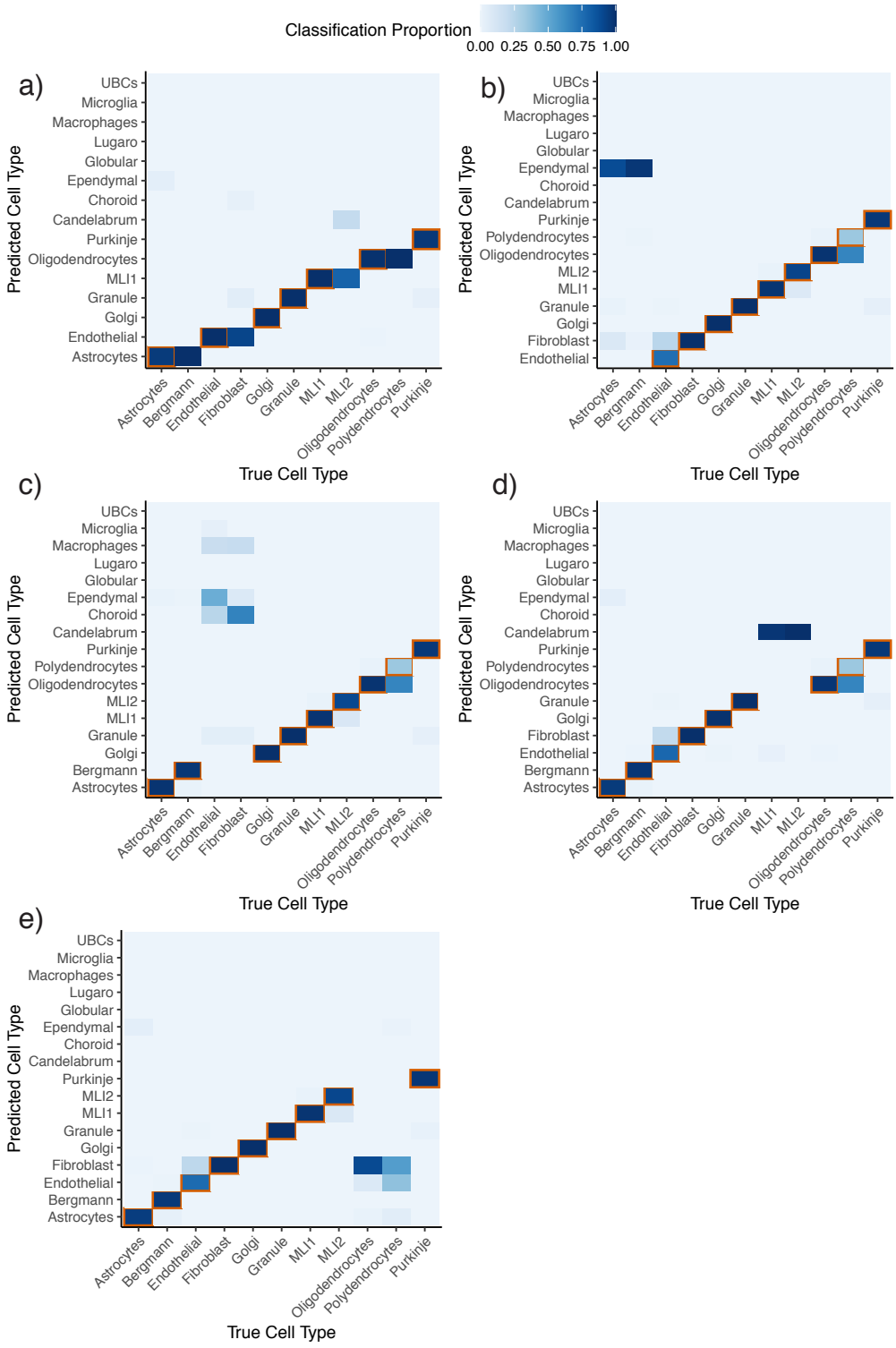
Supplementary Figure 10: Performance of RCTD as a function of number of UMIs.
For simulated datasets, pixels were downsampled for each UMI condition.

a) Percentage of single-cells predicted confidently as a function of UMI. Performance measured on cross-platform single-cell dataset (trained on single-nucleus, tested on single-cell). For "class-removed", the testing cell class was removed from the reference. Performance is averaged over {Astrocytes, Bergmann, Endothelial, Fibroblast, MLI1, MLI2, Oligodendrocytes, Polydendrocytes}.

b) Percentage of pixels predicted confidently as a function of UMI in the cerebellum Slide-seq dataset. Pixels are binned with binwidth 100 UMI. Percentage of cell types with ≥ 1 confident cell type and percentage of doublets with 2 confident cell types is shown.

c) Cell class prediction accuracy as a function of UMI per pixel. Performance measured on simulated doublets dataset (trained on single-nucleus, tested on single-cell). Restricted to confident predictions.

d) Percentage of pixels predicted confidently as a function of UMI. Performance measured on simulated doublets dataset (trained on single-nucleus, tested on single-cell). Percentage of cell types with ≥ 1 confident cell type and percentage of doublets with 2 confident cell types is shown.


Classification Proportion
0.00 0.25 0.50 0.75 1.00
a)
b)
c)
d)
e)
Predicted Cell Type
True Cell Type
UBCs
Microglia
Macrophages
Lugaro
Globular
Ependymal
Choroid
Candelabrum
Purkinje
Polydendrocytes
Oligodendrocytes
MLI2
MLI1
Granule
Golgi
Fibroblast
Endothelial
Bergmann
Astrocytes

Supplementary Figure 11: Performance of RCTD on Cell Types that do not appear in the Reference All panels: confusion matrix for RCTD on cross-reference cell type assignment for single cells. Recall, cell classes are defined as {Astrocytes, Bergmann}, {Endothelial, Fibroblast}, {MLI1, MLI2}, and {Oligodendrocytes, Polydendrocytes}. RCTD is trained on the single-nucleus cerbellum dataset, with cell types removed as follows:

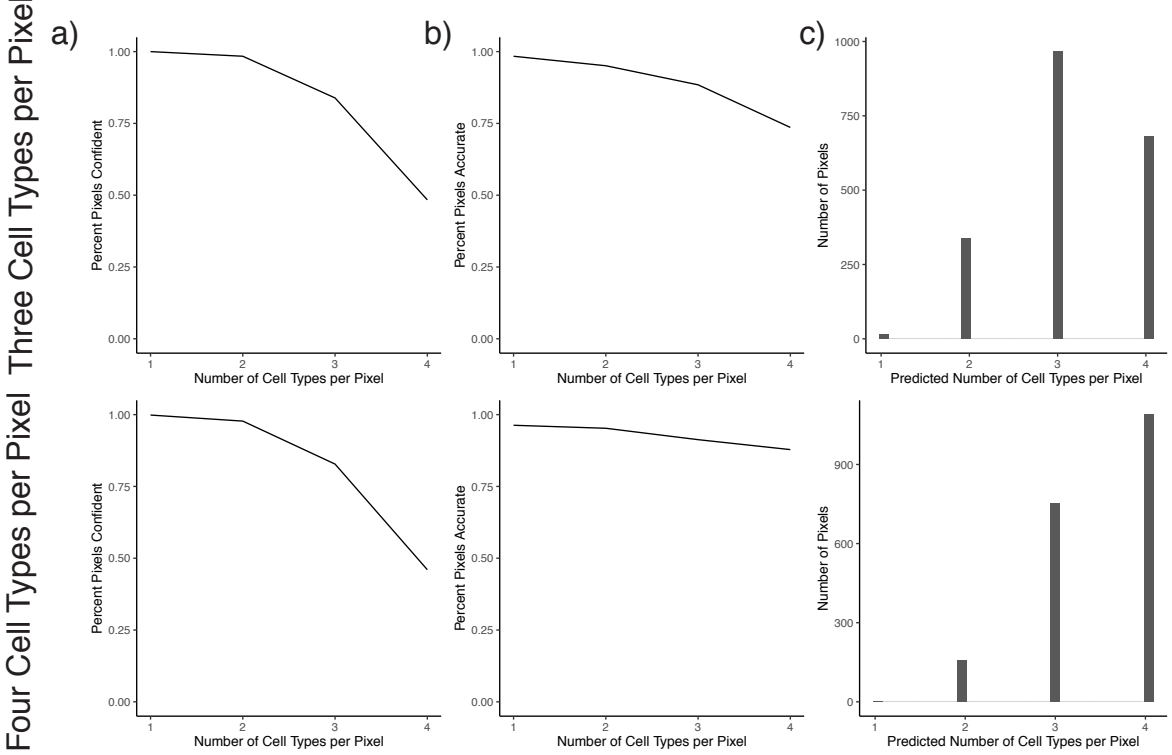
a) Cell types removed (one from each class): {Bergmann, Fibroblast, MLI2, Polydendrocytes}.

b) Cell types removed (two from one class): {Astrocytes, Bergmann}.

c) Cell types removed (two from one class): {Endothelial, Fibroblast}.

d) Cell types removed (two from one class): {MLI1, MLI2}.

e) Cell types removed (two from one class): {Oligodendrocytes, Polydendrocytes}.

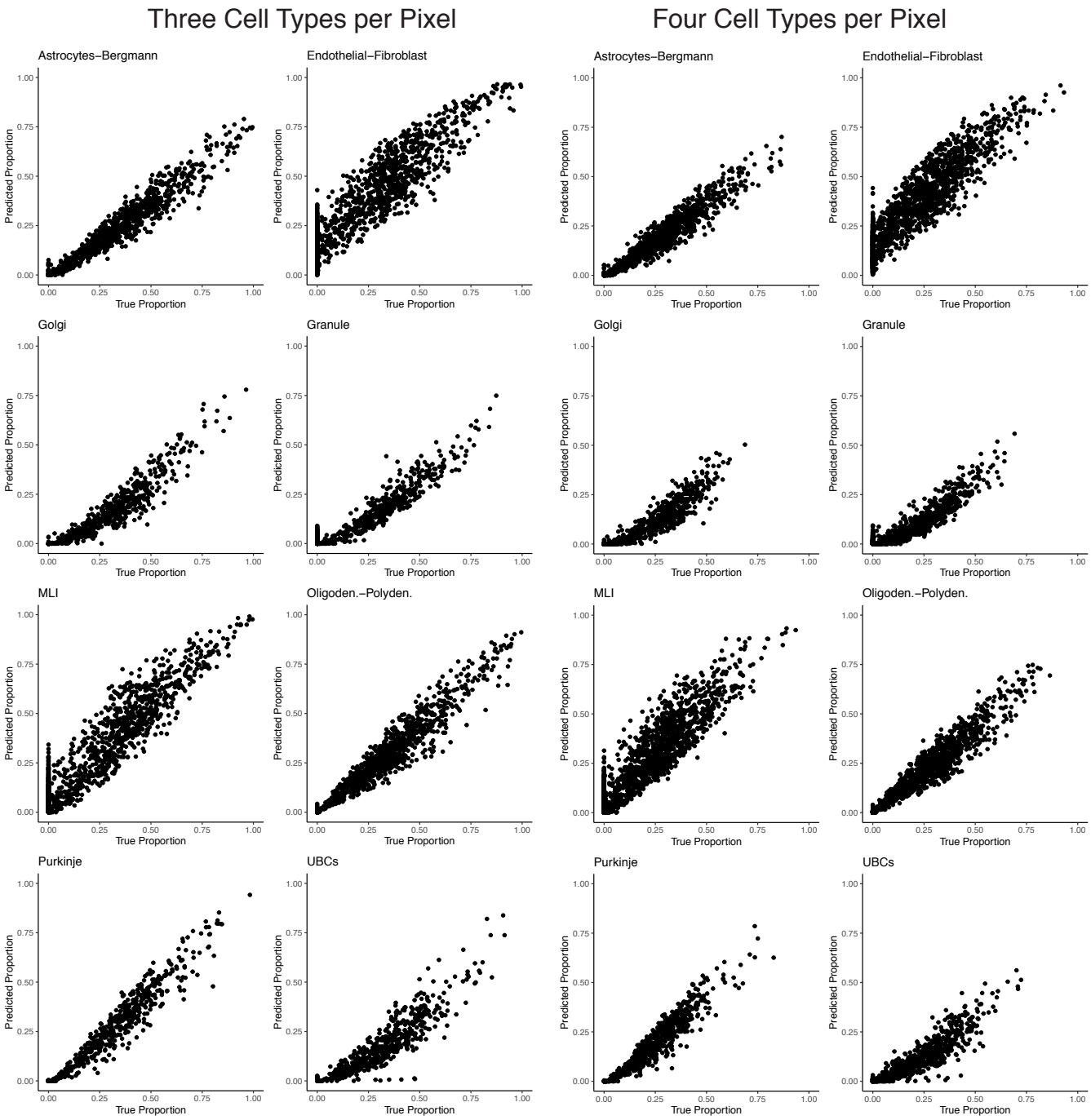

Four Cell Types per Pixel Three Cell Types per Pixel
a)
b)
c)
Percent Pixels Confident
Percent Pixels Accurate
Number of Pixels
Number of Cell Types per Pixel
Predicted Number of Cell Types per Pixel
0.00
0.25
0.50
0.75
1.00
0
250
500
750
1000
300
600
900
1
2
3
4

Supplementary Figure 12: Performance of RCTD on simulated Visium dataset.
All panels: RCTD was trained on single-nucleus cerebellum and tested on single-cell. Simulated Visium dataset consisted of pixels comprised of multiple cells totaling $10,000$ UMIs. Each pixel in the dataset contained either three cell types (top row) or four cell types (bottom row). RCTD was run on multi mode.

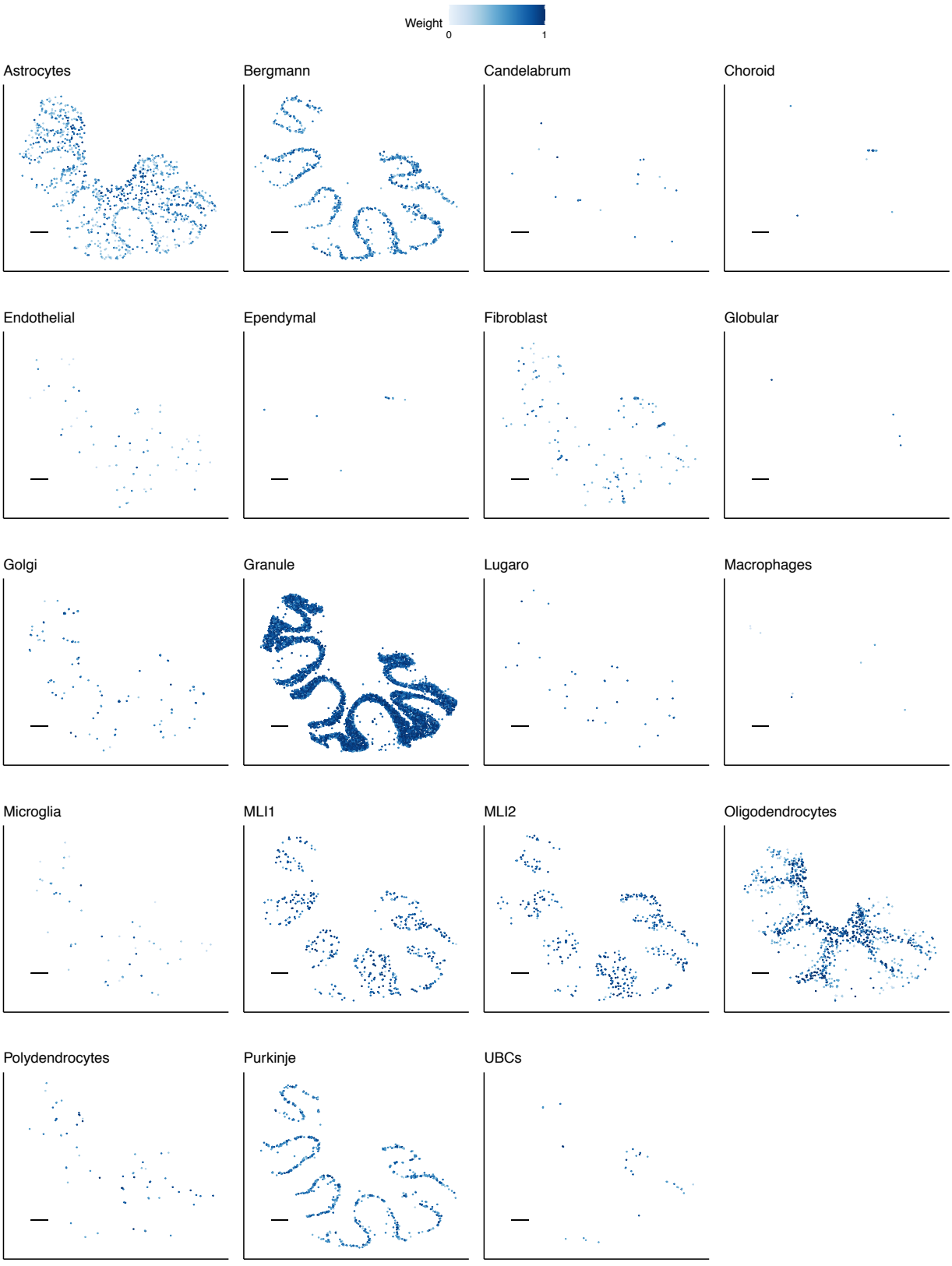
a) For each number of cell types, the proportion of pixels confidently predicting at least as many cell types. Calculated over pixels predicted to contain at least as many cell types.

b) RCTD cell class prediction accuracy as a function of number of cell types confidently predicted per pixel. We observed that RCTD with multi mode achieved high accuracy on cell class prediction. Overall accuracy was 87.7% for the three cell type dataset and 91.0% for the four cell type dataset.

c) Histogram of total cell types predicted per pixel.

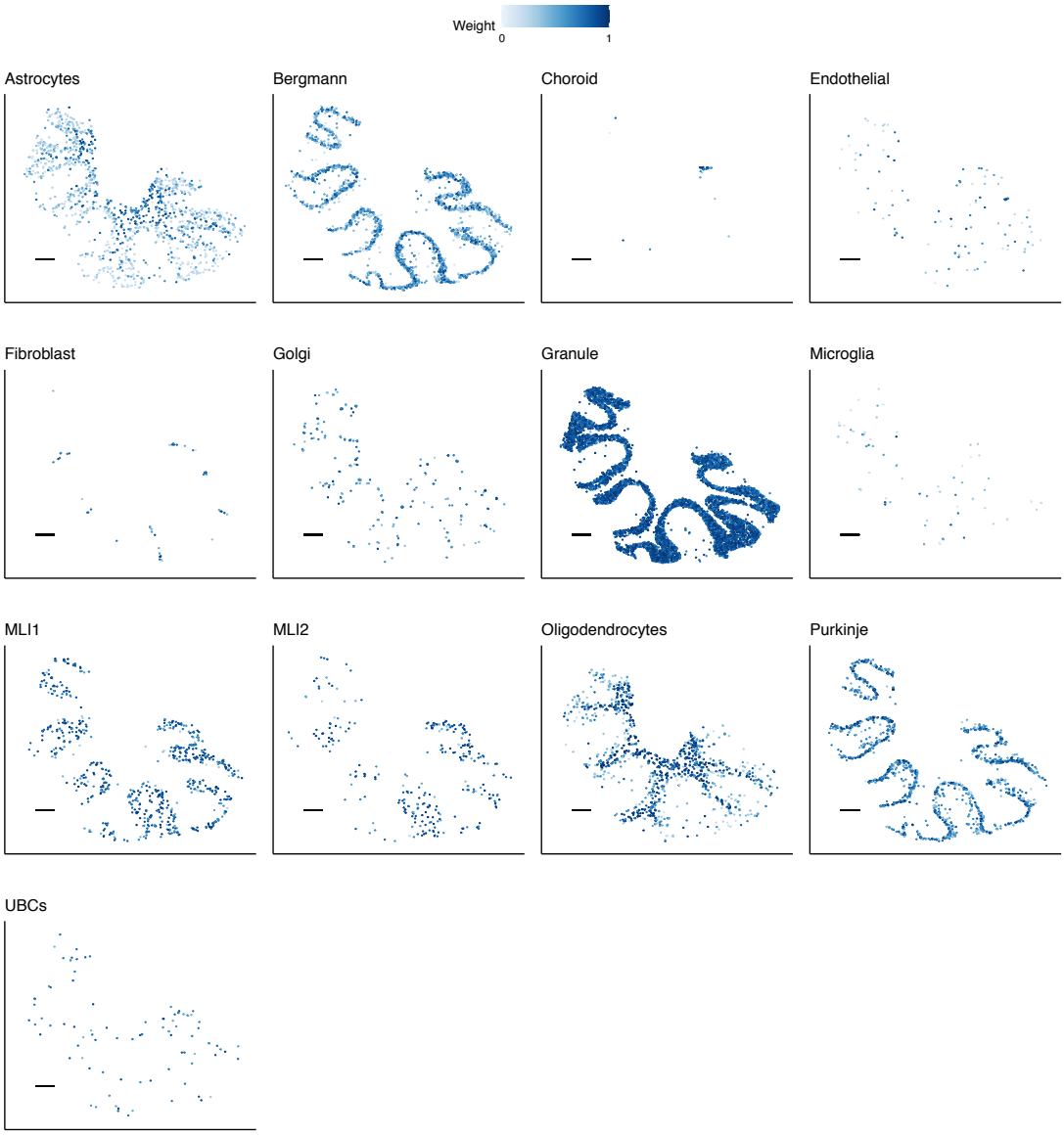

Three Cell Types per Pixel
Four Cell Types per Pixel
Astrocytes−Bergmann
Endothelial−Fibroblast
Golgi
Granule
MLI
Oligoden.−Polyden.
Purkinje
UBCs
Predicted Proportion
True Proportion
0.00
0.25
0.50
0.75
1.00

Supplementary Figure 13: RCTD predictions on simulated Visium dataset.
Scatterplot of true vs. predicted cell class proportions by RCTD. RCTD was trained on single-nucleus cerebellum and tested on single-cell. Simulated Visium dataset consisted of pixels comprised of multiple cells totaling $10,000$ UMIs. Each pixel in the dataset contained either three cell types (left column) or four cell types (right column). RCTD was run without constraining cell types.


Weight
0
1
Astrocytes
Bergmann
Candelabrum
Choroid
Endothelial
Ependymal
Fibroblast
Globular
Golgi
Granule
Lugaro
Macrophages
Microglia
MLI1
MLI2
Oligodendrocytes
Polydendrocytes
Purkinje
UBCs

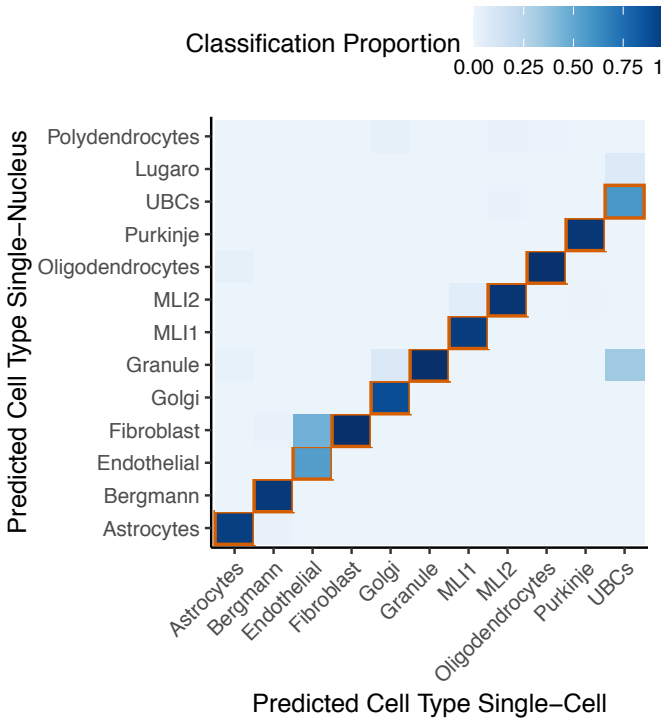
Supplementary Figure 14: Predicted spatial localization of cell types by RCTD in Slide-seq cerebellum.

Predicted spatial locations of each cell type, with color representing predicted cell type proportion. All scale bars are 250 microns. For this analysis, RCTD was trained on the single-nucleus reference.


Weight
0
1
Astrocytes
Bergmann
Choroid
Endothelial
Fibroblast
Golgi
Granule
Microglia
MLI1
MLI2
Oligodendrocytes
Purkinje
UBCs

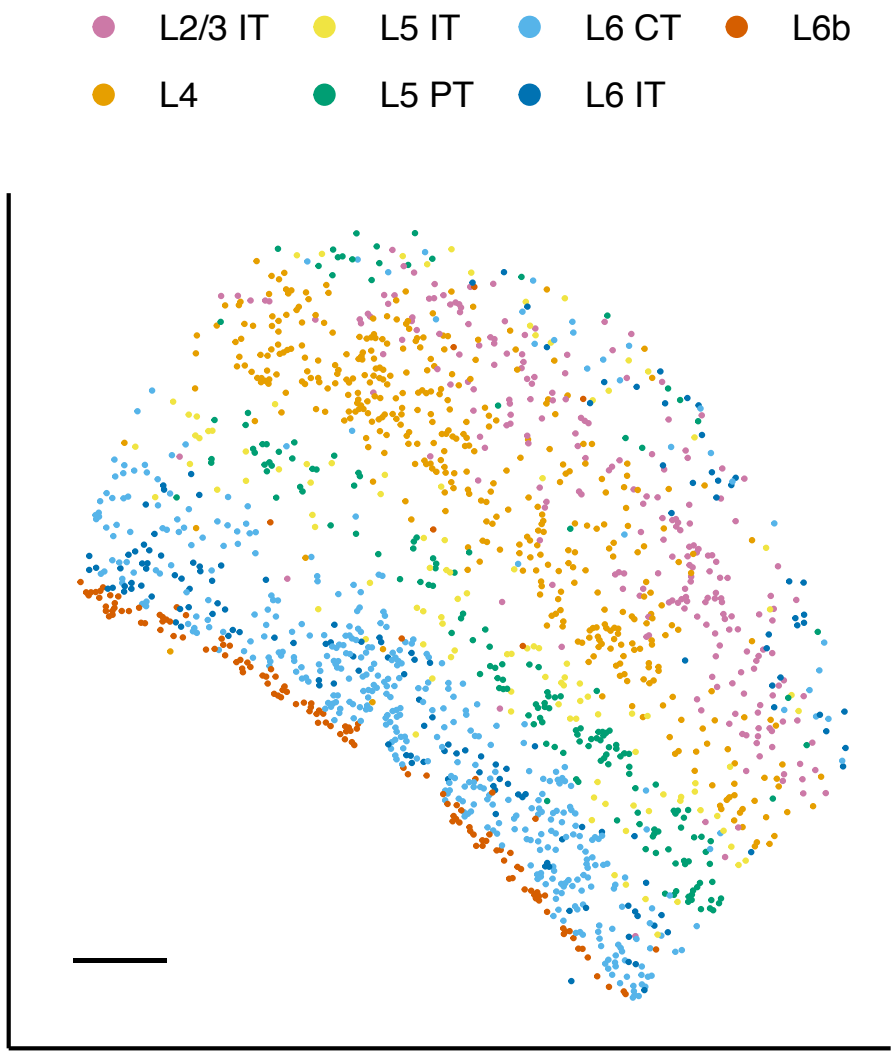
Supplementary Figure 15: Predicted spatial localization of cell types by RCTD in Slide-seq cerebellum, trained on single-cell reference.

Predicted spatial locations of each cell type, with color representing predicted cell type proportion. All scale bars are 250 microns. For this analysis, RCTD was trained on the single-cell reference.

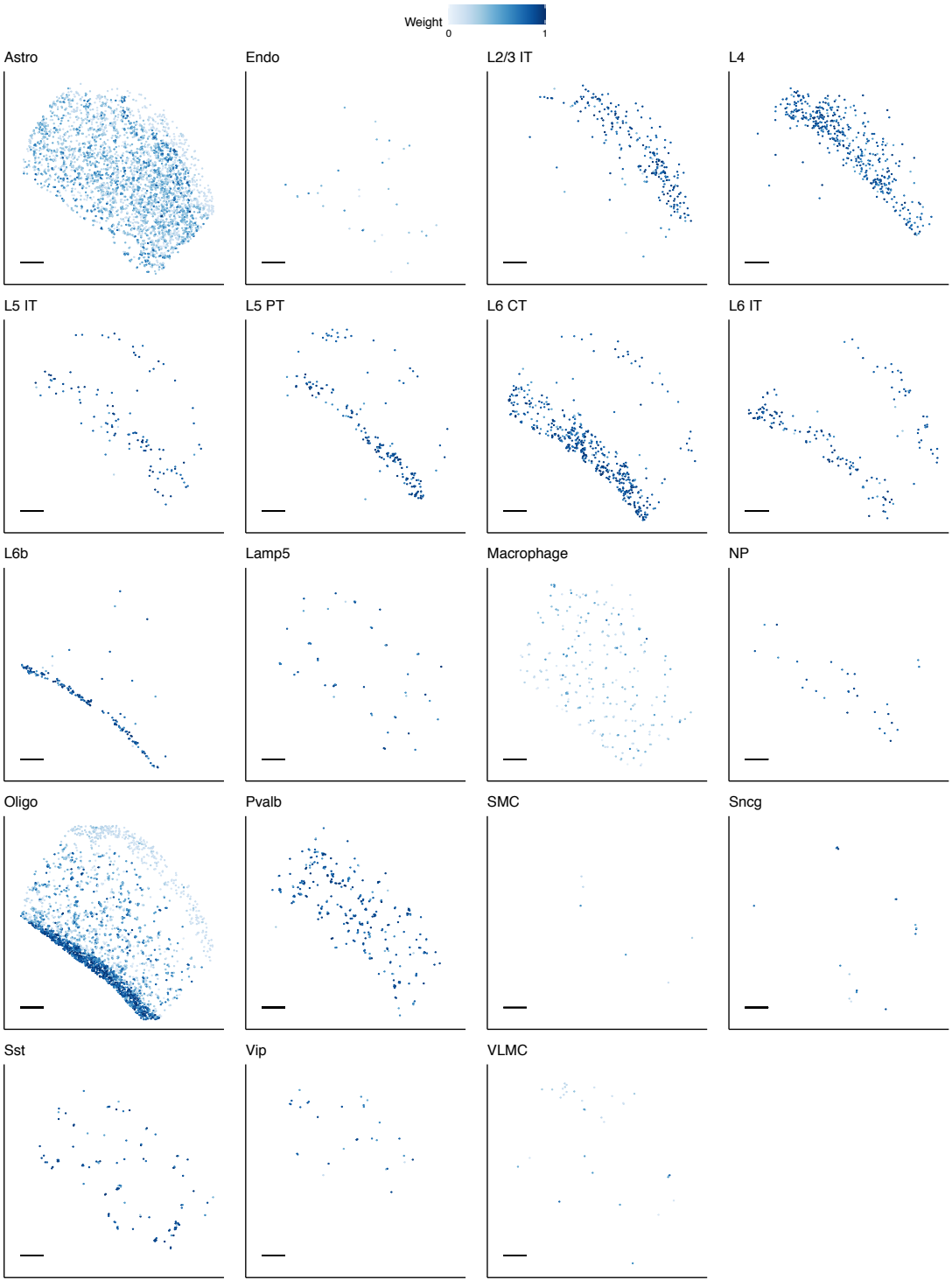

Classification Proportion
0.00 0.25 0.50 0.75 1
Predicted Cell Type Single−Nucleus
Polydendrocytes
Lugaro
UBCs
Purkinje
Oligodendrocytes
MLI2
MLI1
Granule
Golgi
Fibroblast
Endothelial
Bergmann
Astrocytes
Astrocytes
Bergmann
Endothelial
Fibroblast
Golgi
Granule
MLI1
MLI2
Oligodendrocytes
Purkinje
UBCs
Predicted Cell Type Single−Cell

Supplementary Figure 16: Agreement of RCTD predictions across multiple training datasets.

Confusion matrix of cell type classifications on Slide-seq cerbellum, comparing RCTD predictions when trained on the single-cell and single-nucleus cerebellum dataset. Overall, 95.7% of confident cell type predictions are in agreement.

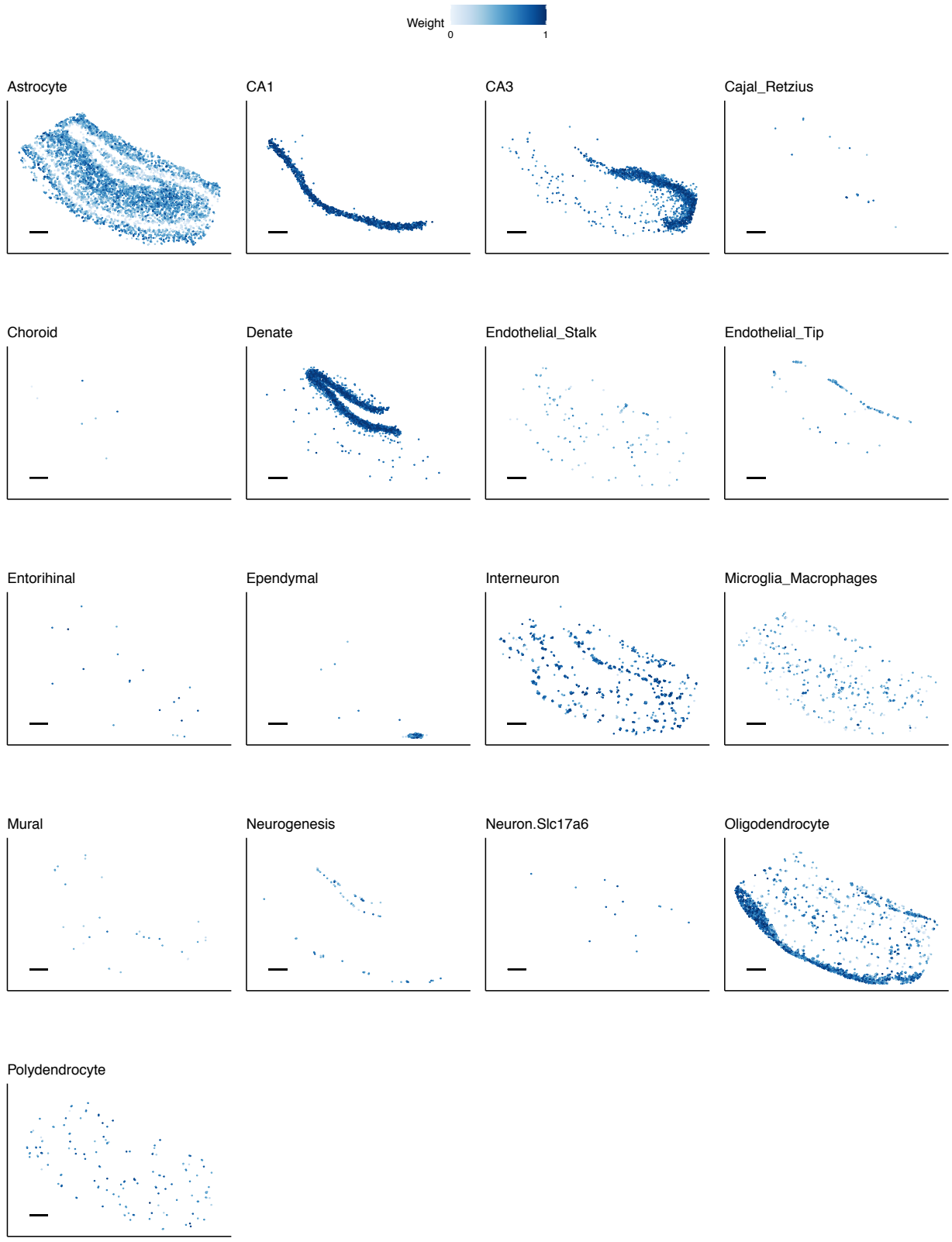

L2/3 IT
L5 IT
L6 CT
L6b
L4
L5 PT
L6 IT

Supplementary Figure 17: RCTD's spatial map of cell type assignments in the somatosensory cortex. Only cortical neuron cell types are shown. Scale bar 250 microns. We abbreviate intratelencephalic as IT, pyramidal tract as PT, and corticothalamic as CT.


Weight
0
1
Astro
Endo
L2/3 IT
L4
L5 IT
L5 PT
L6 CT
L6 IT
L6b
Lamp5
Macrophage
NP
Oligo
Pvalb
SMC
Sncg
Sst
Vip
VLMC

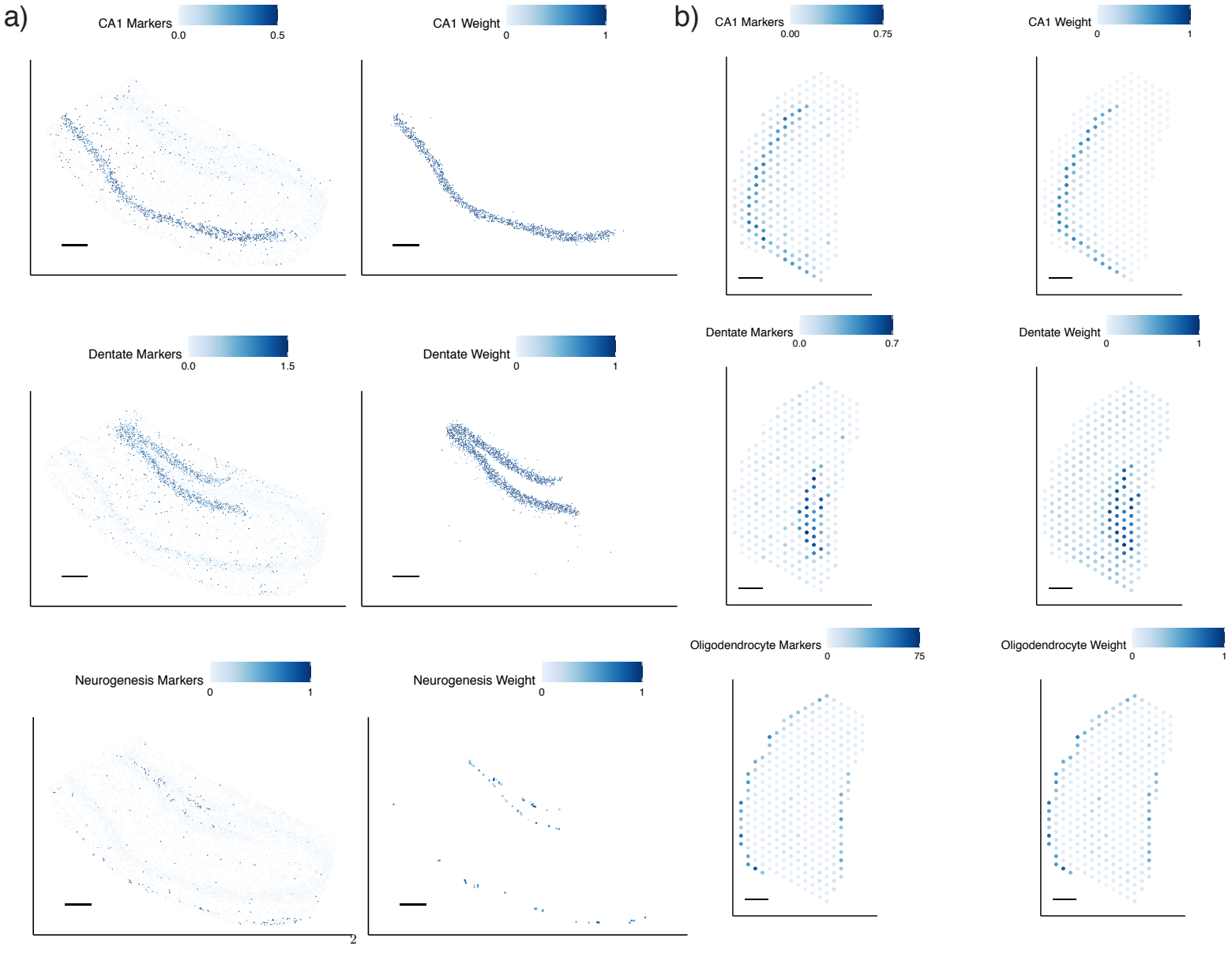
Supplementary Figure 18: Predicted spatial localization of cell types by RCTD in Slide-seq somatosensory cortex.

Predicted spatial locations of each cell type, with color representing predicted cell type proportion. All scale bars are 250 microns. We abbreviate intratelencephalic as IT, pyramidal tract as PT, and corticothalamic as CT.


Weight
0
1
Astrocyte
CA1
CA3
Cajal_Retzius
Choroid
Denate
Endothelial_Stalk
Endothelial_Tip
Entorihinal
Ependymal
Interneuron
Microglia_Macrophages
Mural
Neurogenesis
Neuron.Slc17a6
Oligodendrocyte
Polydendrocyte

Supplementary Figure 19: Predicted spatial localization of cell types by RCTD in Slide-seq hippocampus.

Predicted spatial locations of each cell type, with color representing predicted cell type proportion. All scale bars are 250 microns.

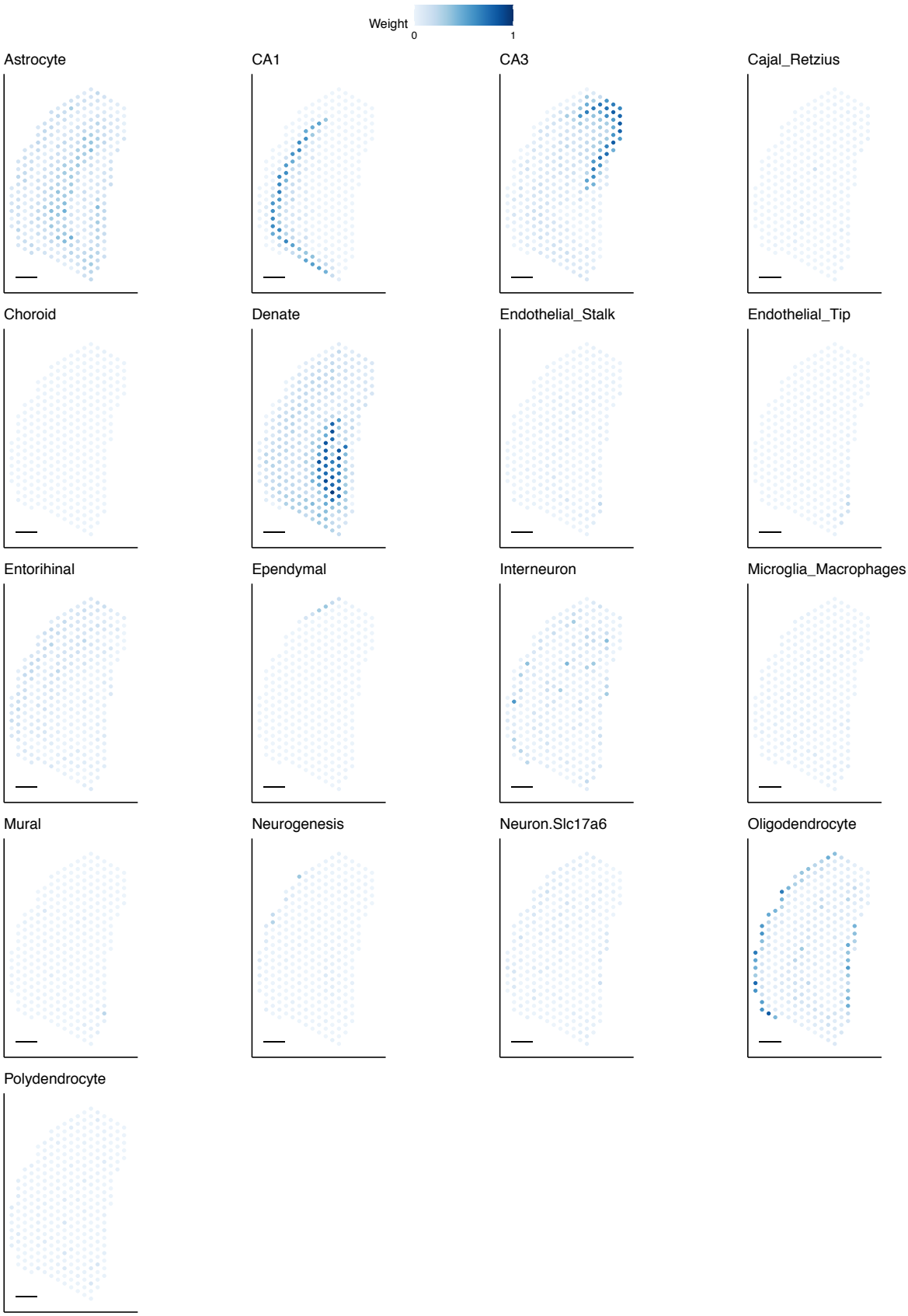

a)
CA1 Markers
0.0
0.5
CA1 Weight
0
1
Dentate Markers
0.0
1.5
Dentate Weight
0
1
Neurogenesis Markers
0
1
Neurogenesis Weight
0
1
2
b)
CA1 Markers
0.00
0.75
CA1 Weight
0
1
Dentate Markers
0.0
0.7
Dentate Weight
0
1
Oligodendrocyte Markers
0
75
Oligodendrocyte Weight
0
1

Supplementary Figure 20: Comparison of marker expression and predicted spatial localization of cell types by RCTD in Slide-seq and Visium hippocampus

a) Slide-seq hippocampus. Left: expression (counts per 500) (represented by color) of marker genes of each cell type. Right: predicted spatial locations of a cell type by RCTD, with color representing predicted cell type proportion.

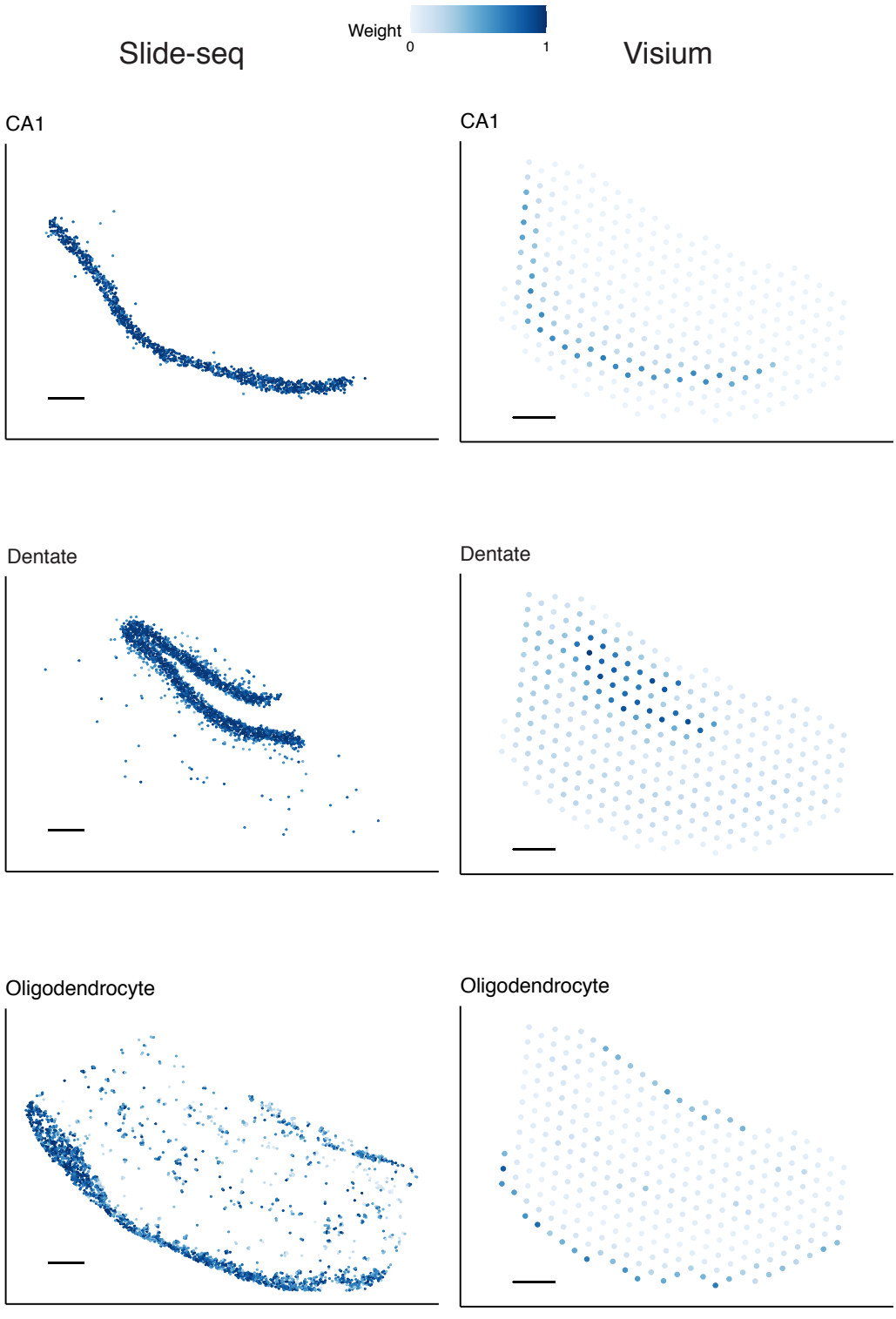
b) Visium hippocampus. Left: expression (counts per 500) (represented by color) of marker genes of each cell type. Right: predicted spatial locations of a cell type by RCTD, with color representing predicted cell type proportion.

All scale bars 250 microns.


Weight
0
1
Astrocyte
CA1
CA3
Cajal_Retzius
Choroid
Denate
Endothelial_Stalk
Endothelial_Tip
Entorihinal
Ependymal
Interneuron
Microglia_Macrophages
Mural
Neurogenesis
Neuron.Slc17a6
Oligodendrocyte
Polydendrocyte

Supplementary Figure 21: Predicted spatial localization of cell types by RCTD in Visium hippocampus.

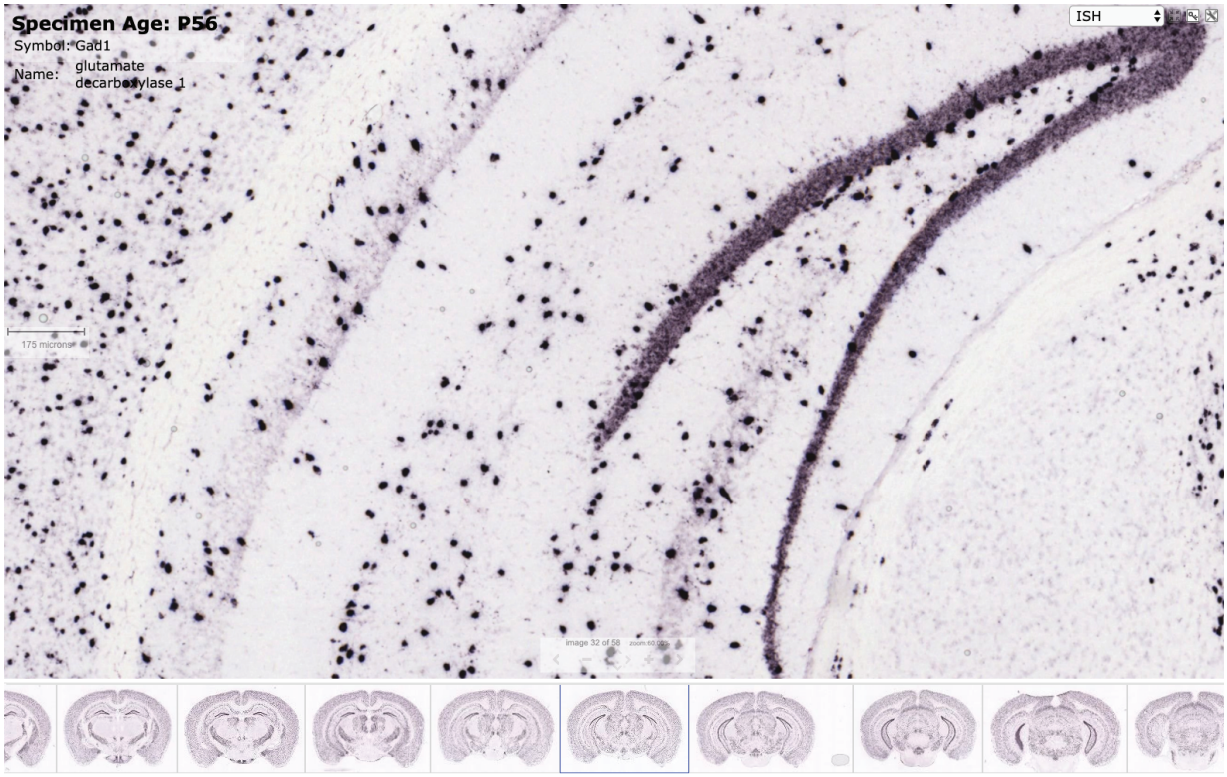
Predicted spatial locations of each cell type, with color representing predicted cell type proportion. pixels were not constrained in the number of cell types present. All scale bars are 250 microns.


Weight
0
1
Slide-seq
Visium
CA1
CA1
Dentate
Dentate
Oligodendrocyte
Oligodendrocyte

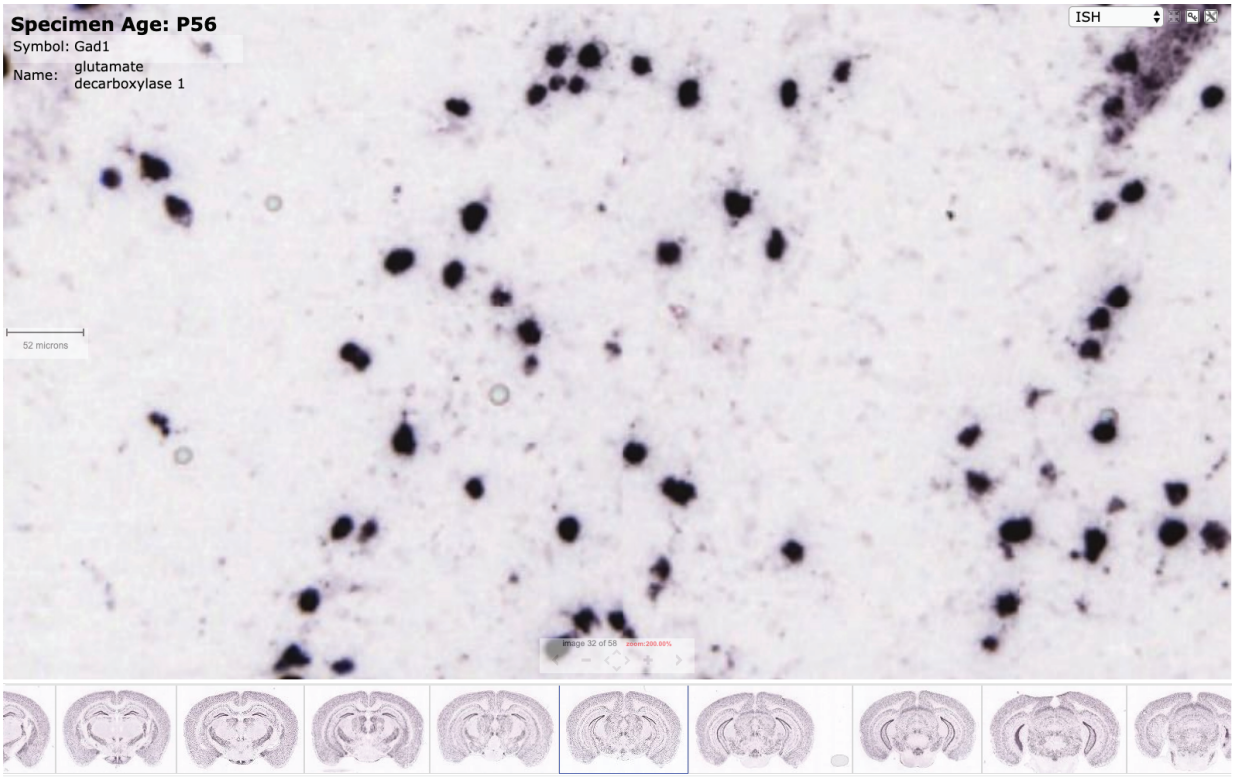
Supplementary Figure 22: Comparison of spatial localization of cell types by RCTD in Slide-seq and Visium hippocampus.

Predicted spatial locations of each cell type on Slide-seq (left) or Visium (right), with color representing predicted cell type proportion. All scale bars are 250 microns.

a)


Specimen Age: P56
Symbol: Gad1
Name: glutamate decarboxylase 1
ISH
175 microns

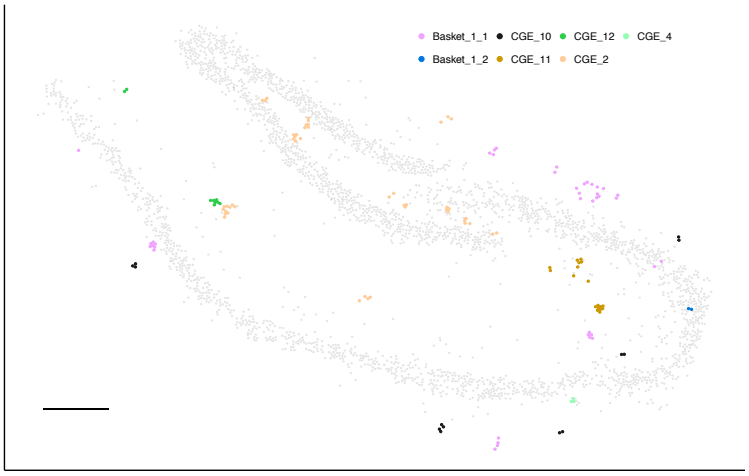
b)

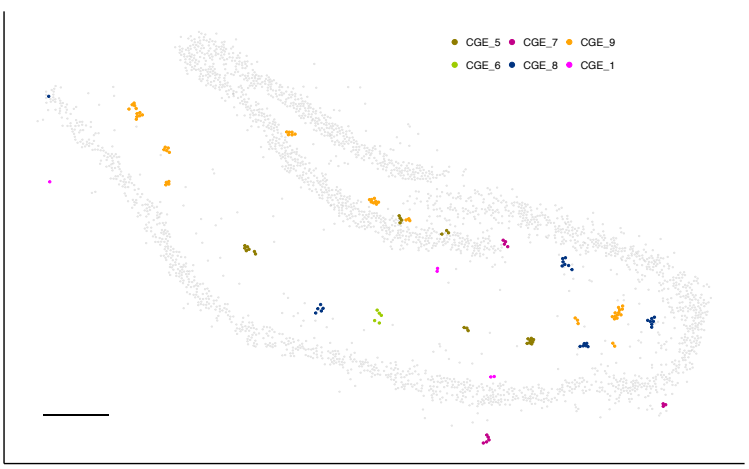

Specimen Age: P56
Symbol: Gad1
Name: glutamate decarboxylase 1
ISH
52 microns

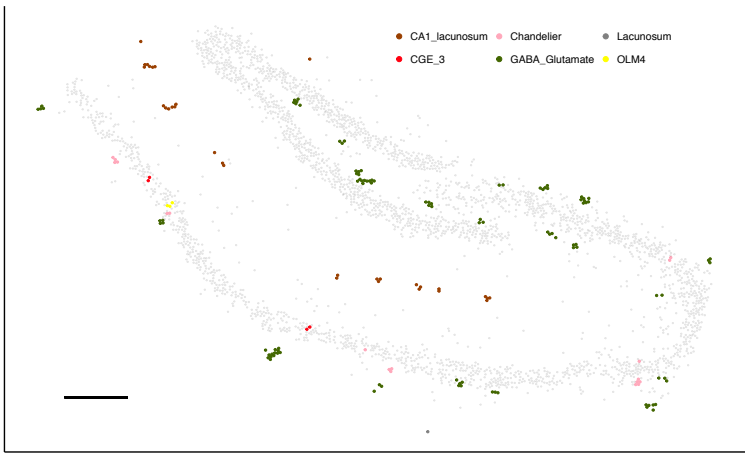
Supplementary Figure 23: In Situ Hybridization (ISH) of *Gad1*, an interneuron marker gene, from the Allen Brain Atlas [10].

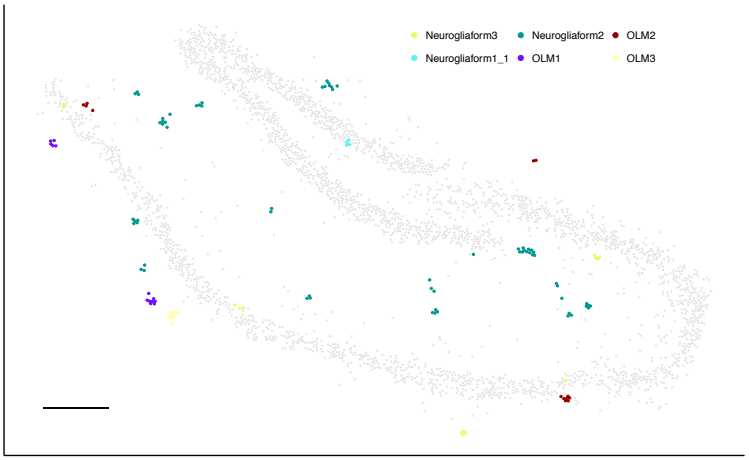
a) Scale bar 175 microns.

b) Scale bar 52 microns.

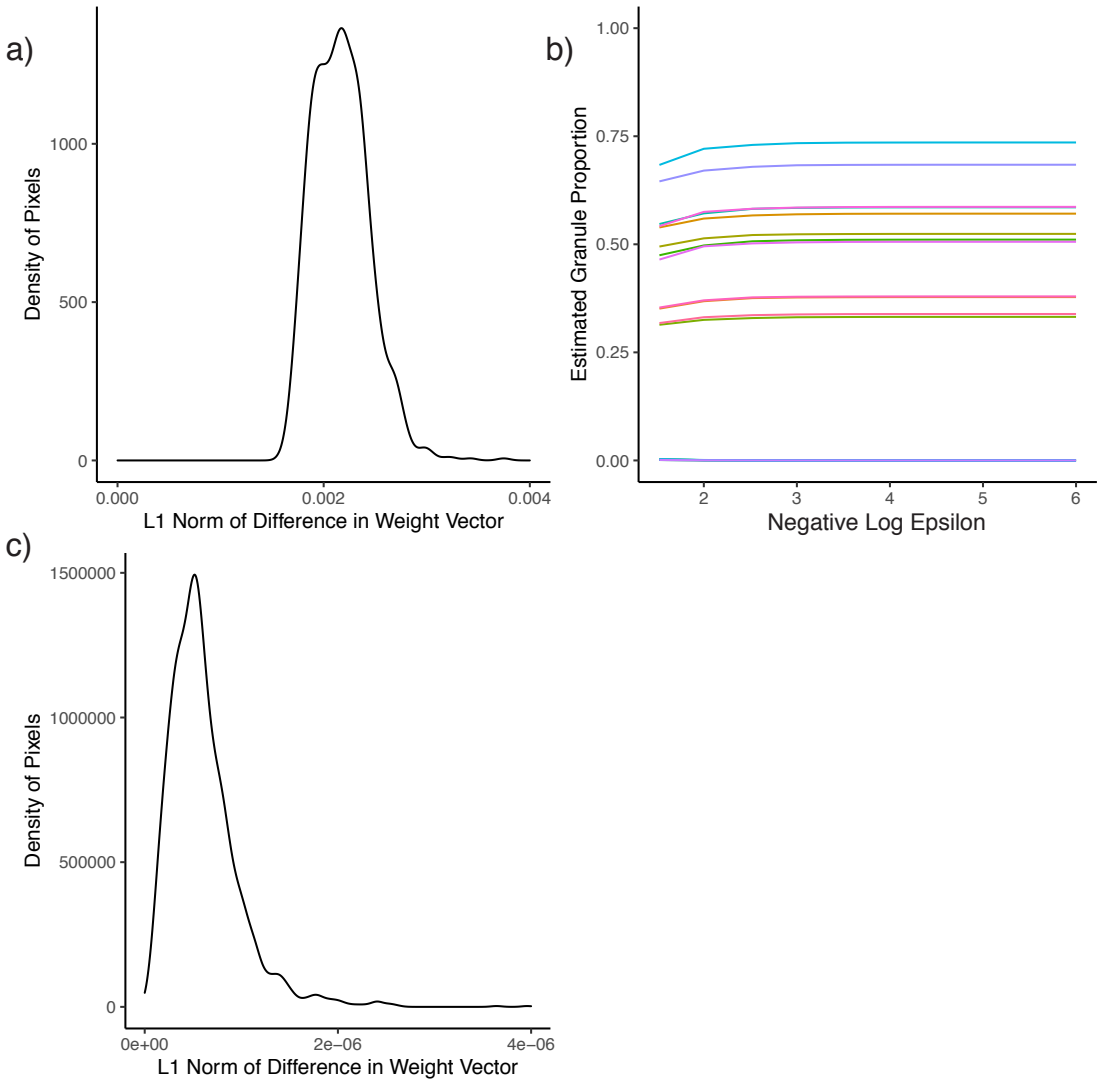

Basket_1_1
CGE_10
CGE_12
CGE_4
Basket_1_2
CGE_11
CGE_2


CGE_5
CGE_7
CGE_9
CGE_6
CGE_8
CGE_1


CA1_lacunosum
Chandelier
Lacunosum
CGE_3
GABA_Glutamate
OLM4


Neurogliaform3
Neurogliaform2
OLM2
Neurogliaform1_1
OLM1
OLM3

Supplementary Figure 24: RCTD's confident assignment of spatial clusters to 27 interneuron subtypes (25/27 subtypes assigned) in the Slide-seq hippocampus. Split into four panels in order to clearly display individual cell types. All scale bars 250 microns. Grey circles represent location of CA1, CA3, and dentate gyrus excitatory neurons for reference.


a)
Density of Pixels
1000
500
0
0.000
0.002
0.004
L1 Norm of Difference in Weight Vector
b)
Estimated Granule Proportion
1.00
0.75
0.50
0.25
0.00
2
3
4
5
6
Negative Log Epsilon
c)
Density of Pixels
1500000
1000000
500000
0
0e+00
2e−06
4e−06
L1 Norm of Difference in Weight Vector

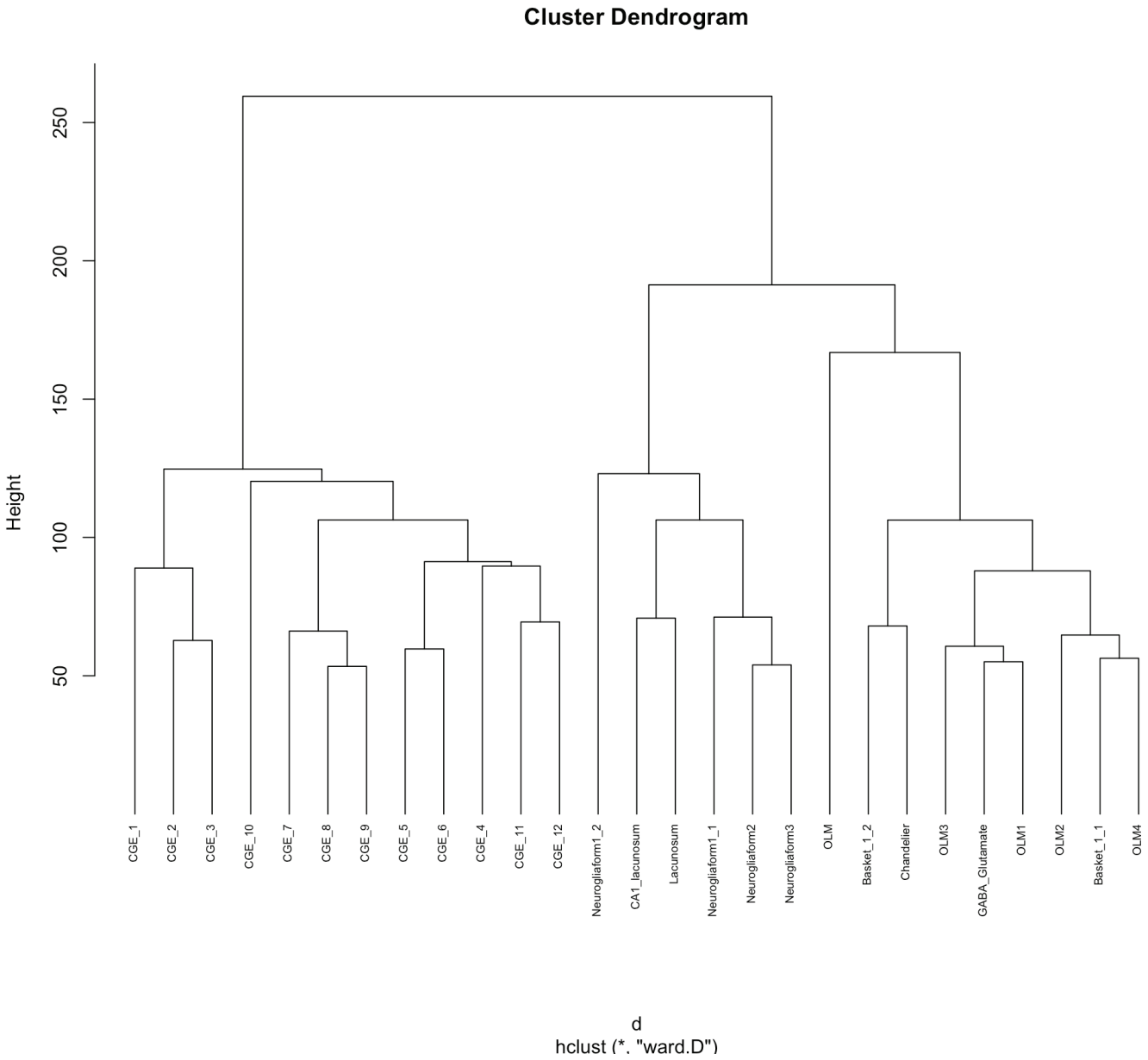
Supplementary Figure 25: Convergence Properties of RCTD.

a) Density plot (over $1,000$ randomly selected pixels) of L1 norm of difference in weight vector between RCTD with convergence parameter $\varepsilon = 10^{-3}$ and $\varepsilon = 10^{-6}$.

b) For several randomly chosen pixels, plot of granule proportion over several values of ε (represented in negative $\log_{10}$). As ε becomes small, the estimated proportion does not change very much.

c) Density plot (over 50 randomly chosen initial values for each of 50 randomly selected pixels) of L1 norm of difference in weight vector between RCTD initialized to a uniform random chosen initial value and to the default initial value. We ran RCTD with convergence parameter $\varepsilon = 10^{-6}$.

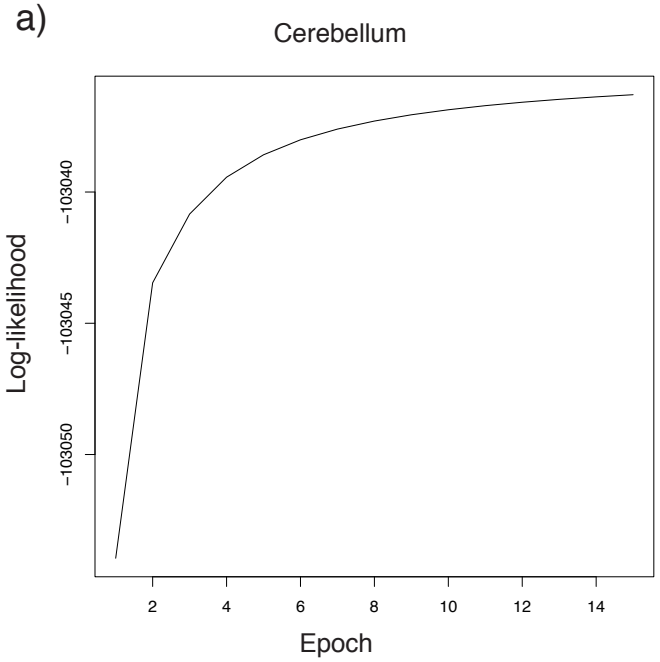
All analysis occurs on the Slide-seq cerebellum.

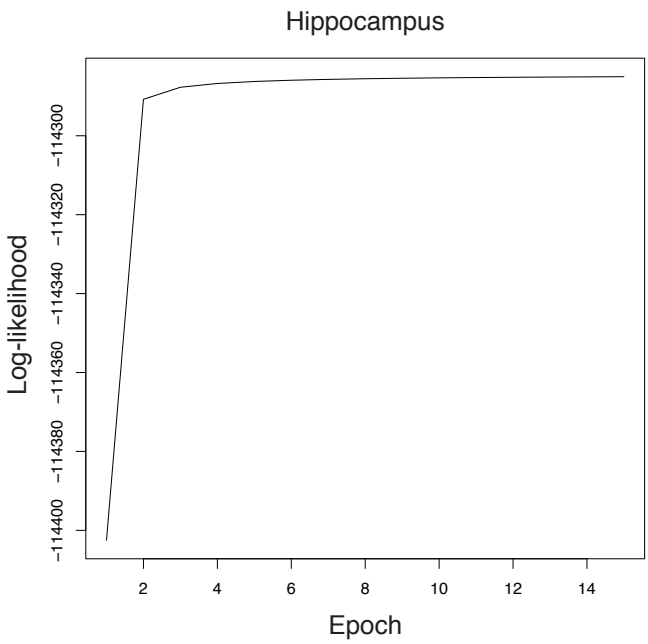

Cluster Dendrogram
Height
250
200
150
100
50
CGE_1
CGE_2
CGE_3
CGE_10
CGE_7
CGE_8
CGE_9
CGE_5
CGE_6
CGE_4
CGE_11
CGE_12
Neurogliaform1_2
CA1_lacunosum
Lacunosum
Neurogliaform1_1
Neurogliaform2
Neurogliaform3
OLM
Basket_1_2
Chandelier
OLM3
GABA_Glutamate
OLM1
OLM2
Basket_1_1
OLM4
d
hclust (*, "ward.D")

Supplementary Figure 26: Dendrogram of interneuron subtypes.

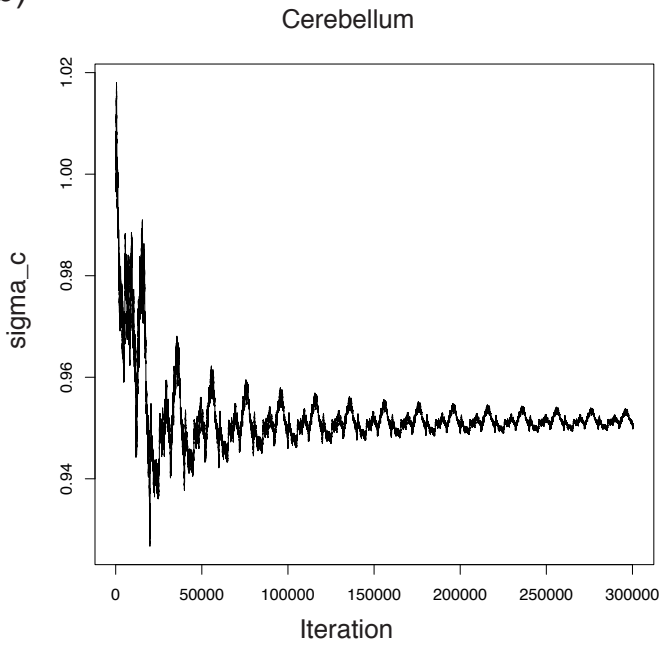
Dendrogram of 27 interneuron subtypes that appeared in the scRNA-seq hippocampus dataset, hierarchically clustered with Ward clustering.

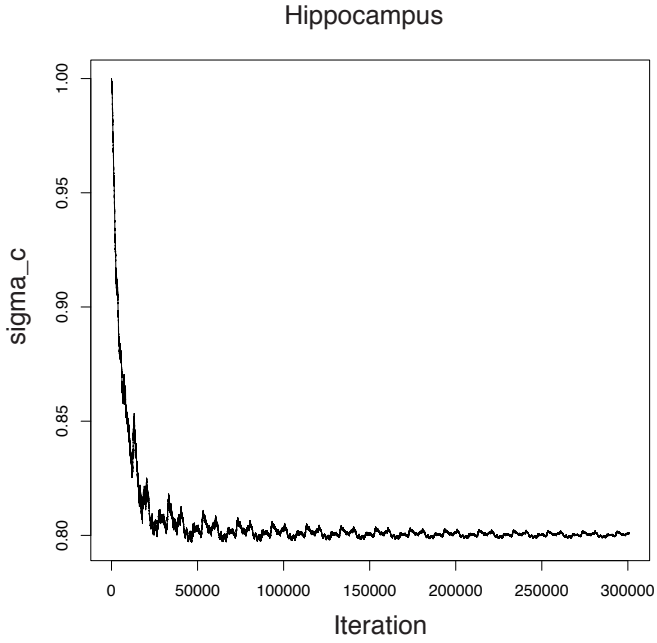
a)


Cerebellum
Log-likelihood
−103040
−103045
−103050
2
4
6
8
10
12
14
Epoch


Hippocampus
Log-likelihood
−114300
−114320
−114340
−114360
−114380
−114400
2
4
6
8
10
12
14
Epoch

b)


Cerebellum
sigma_c
1.02
1.00
0.98
0.96
0.94
0
50000
100000
150000
200000
250000
300000
Iteration


Hippocampus
sigma_c
1.00
0.95
0.90
0.85
0.80
0
50000
100000
150000
200000
250000
300000
Iteration

Supplementary Figure 27: Trace plots showing convergence of σ_ε hyperparameter during optimization in RCTD.

a) Value of log-likelihood at each stochastic gradient descent (SGD) epoch optimizing σ_ε. Left: Slide-seq cerebellum. Right: Slide-seq hippocampus.

b) Value of σ_ε at each stochastic gradient descent (SGD) iteration optimizing σ_ε. Left: Slide-seq cerebellum. Right: Slide-seq hippocampus.

References

[1] Blei, D. M., Ng, A. Y. & Jordan, M. I. Latent dirichlet allocation. *Journal of machine Learning research* **3**, 993–1022 (2003).

[2] Efron, B. Multiparameter Exponential Families, notes for Stats 305B: Methods for Applied Statistics I: Exponential Families in Theory and Practice, Stanford University (2019).

[3] Agresti, A. *Categorical data analysis, third edition*, vol. 482 (John Wiley & Sons, 2012).

[4] Townes, F. W., Hicks, S. C., Aryee, M. J. & Irizarry, R. A. Feature selection and dimension reduction for single-cell rna-seq based on a multinomial model. *Genome biology* **20**, 1–16 (2019).

[5] Duchi, J. Sequential Convex Programming, notes for EE364b: Convex Optimization II, Stanford University (2018).

[6] Sakamoto, Y., Ishiguro, M. & Kitagawa, G. Akaike information criterion statistics. *Dordrecht, The Netherlands: D. Reidel* **81** (1986).

[7] Stickels, R. R. *et al.* Sensitive spatial genome wide expression profiling at cellular resolution. *bioRxiv* (2020). https://www.biorxiv.org/content/early/2020/03/14/2020.03.12.989806.full.pdf.

[8] Brown, A. M. *et al.* Molecular layer interneurons shape the spike activity of cerebellar purkinje cells. *Scientific reports* **9**, 1–19 (2019).

[9] Rodriques, S. G. *et al.* Slide-seq: A scalable technology for measuring genome-wide expression at high spatial resolution. *Science* **363**, 1463–1467 (2019).

[10] Sunkin, S. M. *et al.* Allen brain atlas: an integrated spatio-temporal portal for exploring the central nervous system. *Nucleic acids research* **41**, D996–D1008 (2012).