# Fine-scale cellular deconvolution via generalized maximum entropy on canonical correlation features

Jack Kamm
Genentech, Research and Early Development
Department of Computational Biology and Translation
1 DNA Way, South San Francisco, CA, USA 94080

June 7, 2024

**Abstract**

We propose a method for estimating probability distributions over single cells, which we apply to fine-scale cellular deconvolution, which quantifies the composition of external bulk RNAseq samples at high resolution (i.e. at the single-cell or neighborhood level). Our method is based on a computationally-efficient convex optimization problem, and is also an application of the Generalized Cross Entropy method for density estimation. Our method has a much higher resolution than traditional approaches that require computing gene expression profiles at the cell-type level, and also compares favorably to recent high-resolution cellular deconvolution methods, with orders-of-magnitude speedup in computational efficiency. We implement this method in a Python package quipcell, available at `https://github.com/genentech/quipcell`.

*Keywords:* Cellular deconvolution, convex optimization, quadratic programming, single cell, Generalized Cross Entropy, Maximum Entropy, density estimation, $L_2$ minimization, $f$-divergence, $\alpha$-divergence, Linear Discriminant Analysis, Canonical Correlation Analysis

## 1 Introduction

Single-cell RNAseq (scRNAseq) is a powerful tool for understanding heterogeneous biological samples consisting of diverse tissues, cell types, or cell states. A common application of scRNAseq data is to quantify differences in cell composition between samples or conditions. Two variants of this task are *differential abundance* analysis and *cellular deconvolution* analysis (also known as *bulk deconvolution*). In differential abundance analysis, we seek to compare cell composition between samples within the same single-cell dataset (Amezquita et al. (2023),

Chapter 6). By contrast, in bulk deconvolution analysis, we use single-cell data to estimate the composition of external bulk RNAseq samples (Maden et al., 2023; Nguyen et al., 2024).

One straightforward approach to estimating sample composition, in both differential abundance and bulk deconvolution, is to bin cells into discrete cell types or clusters, and then to quantify the abundance of these discrete groups across samples. However, this coarse-grained approach can miss fine-scale differences in cell type composition, such as from continuous cell differentiation processes.

In recent years, several high-resolution methods have been developed for differential abundance. Typically, these methods involve building a nearest-neighbor graph of the single cells, and then quantifying the abundance of samples along the graph. In particular, the methods MELD (Burkhardt et al., 2021) and CNA (Reshef et al., 2022) use a random walk on the nearest neighbor graph to represent each sample as a probability distribution over cells, while miloR (Dann et al., 2022) breaks up the graph into overlapping neighborhoods and measures the sample frequency and spatial correlation of the neighborhoods. DA-seq (Zhao et al., 2021) is another high-resolution differential abundance method that makes use of nearest-neighbor information, by fitting logistic regression models in neighborhoods around each cell to distinguish between samples.

By contrast, there are relatively few bulk deconvolution methods that work at single cell resolution. Instead, most deconvolution methods aggregate single cell data into expression profiles of discrete cell types, and then model the bulk samples as mixtures of these discrete profiles (Altboum et al., 2014; Newman et al., 2019; Wang et al., 2019; Jew et al., 2020; Dong et al., 2021; Erdmann-Pham et al., 2021). One notable exception is CPM (Frishberg et al., 2019), which uses Support Vector Regression to model the bulk samples as a linear combination of reference single cells. However, CPM was designed to distinguish continuous cell states within cell types, and may not work well on datasets containing very diverse cell types. More recent high-resolution deconvolution methods include ConDecon (Aubin et al., 2023), which learns a map from rank-correlation scores to a probability distribution over single cells; and MeDuSa (Song et al., 2023), which deconvolves bulk samples along a linear continuous pseudotime trajectory using a mixed effects model. In addition, spatial deconvolution methods, such as Tangram (Biancalani et al., 2021), bear some similarities to high-resolution bulk deconvolution, but operate in a different regime with distinct challenges (many voxels with few cells per voxel) as well as advantages (additional spatial information, well-matched single-cell and spatial samples).

We propose a novel method for bulk deconvolution, quipcell, that is a convex optimization problem and a Generalized Cross Entropy method (Botev and Kroese, 2011). Quipcell represents each sample as a probability distribution over some reference single-cell dataset, similar to the MELD differential abundance method (Burkhardt et al., 2021), or the ConDecon bulk deconvolution method (Aubin et al., 2023). However, quipcell fits this distribution using an alternative approach that only relies on solving an efficient convex optimization problem, instead of constructing nearest neighbor graphs or fitting predictive machine learning models.

As a Generalized Cross Entropy method for density estimation (Botev and Kroese, 2011), quipcell involves matching distributional moments while minimizing an $f$-divergence (Csiszár, 1964) from the uniform distribution. By default, quipcell uses the $\alpha$-divergence with $\alpha = 2$ (also known as Pearson's $\chi^2$-divergence), which has particularly favorable properties and is a Quadratic Program (QP). However, quipcell also allows using other

values for $\alpha$, including the special case of $\alpha = 1$ which is equivalent to the Maximum Entropy method.

A key aspect of this density estimation procedure is the embedding space used to represent the single cells. Quipcell requires this embedding to be a linear transformation of the original single cell data. We show that an effective approach is to use Canonical Correlation Analysis (CCA) (Hotelling, 1936) to find directions in gene expression space that are highly correlated with celltype-relevant features. In the case that the single cell data is annotated with discrete celltype labels, we perform CCA between the gene expression matrix and the indicator matrix of the celltype labels; this is also equivalent to the multiclass Linear Discriminant Analysis (LDA) (Rao, 1948). In the case where the single cell data does not have discrete celltype labels, but is instead annotated with a continuous "pseudotime", we perform CCA between the gene expression matrix and a cubic B-spline basis (De Boor, 1978) to extract the embedding space.

For our application (Section 3), we consider three datasets. First, we apply quipcell to the Human Lung Cell Atlas dataset (Sikkema et al., 2023), which is a highly heterogeneous reference set consisting of many studies and technologies. We show that our method has good accuracy and runtime properties on this heterogeneous reference, especially compared to existing high-resolution bulk deconvolution methods. Next, we apply quipcell to the problem of pseudotime deconvolution on datasets of stem cells (Chu et al., 2016) and monocytes (Oetjen et al., 2018), and show comparable performance to the pseudotime deconvolution method MeDUSa (Song et al., 2023).

Quipcell is implemented as a Python package, available at `https://github.com/genentech/quipcell`.

# 2 Methods

Let the single-cell reference set $\mathbf{X} \in \mathbb{R}^{n \times p}$ be a matrix of $n$ cells by $p$ features. For example, $\mathbf{X}$ could be normalized scRNAseq gene counts, in which case $p$ would equal the number of genes.

Let $\boldsymbol{\mu}_s \in \mathbb{R}^p$ be the bulk expression of the sample $s$ on these $p$ features. We aim to represent $s$ as a probability distribution $\mathbb{P}_s$ over $\mathbb{R}^p$ such that $\boldsymbol{\mu}_s = \mathbb{E}_{\mathbf{x} \sim \mathbb{P}_s}[\mathbf{x}]$. Intuitively, $\mathbb{P}_s$ represents a probability distribution over single cells, and the bulk expression levels $\boldsymbol{\mu}_s$ are measured as an average over the single cells in $s$. However, note that $\mathbb{P}_s(\mathbf{x})$ is *cell-size biased*, because larger cells contribute more reads to the bulk expression $\boldsymbol{\mu}_s$. In particular, $\mathbb{P}_s$ represents the probability that a random *read* from $s$ is sampled from a cell at $\mathbf{x}$, rather than the probability that a random *cell* from $s$ is sampled at $\mathbf{x}$. In Section 2.3, we discuss how to correct the cell-size bias and convert the read-level probabilities to cell-level probabilities.

We approximate $\mathbb{P}_s$ by a distribution $\hat{\mathbb{P}}_s$ with support on the $n$ single cells (row vectors) of the reference set $\mathbf{X}$. We let $\hat{\mathbf{w}}_s = (\hat{w}_{s,c})_{1 \leq c \leq n} \in \mathbb{R}^n$ denote the corresponding vector of probability weights $\hat{w}_{s,c} = \hat{\mathbb{P}}_s(\mathbf{x}_{c,:})$, where $\mathbf{x}_{c,:}$ denotes the $c$-th row vector (cell) in $\mathbf{X}$. Intuitively, we interpret $\hat{w}_{s,c}$ as the probability that a random read from $s$ originates from a small neighborhood around cell $c$. Furthermore, we use $\hat{w}_{s,c}$ as importance sampling weights to approximate expectations over $\mathbb{P}_s$, i.e. $\mathbb{E}_{\mathbf{x} \sim \mathbb{P}_s}[f(\mathbf{x})] \approx \frac{1}{n} \sum_c \hat{w}_{s,c} f(\mathbf{x}_{c,:})$.

We fit $\hat{\mathbf{w}}_s \in \mathbb{R}^n$ by solving the optimization problem (for any fixed $\epsilon \geq 0$ and fixed $\alpha$):

$$\begin{aligned} \hat{\mathbf{w}}_s = \ & \arg\min_{\mathbf{w}} && D_\alpha(\mathbf{w}||\text{Uniform}) \\ & \text{s.t.} && \mathbf{w}^T\mathbf{X} - \boldsymbol{\mu}_s \preceq \epsilon, \\ & && \mathbf{w}^T\mathbf{X} - \boldsymbol{\mu}_s \succeq -\epsilon, \\ & && \mathbf{w} \succeq 0, \\ & && \|\mathbf{w}\|_1 = 1. \end{aligned} \tag{1}$$

where $D_\alpha(\mathbf{w}||\text{Uniform})$ denotes the $\alpha$-divergence of $\mathbf{w}$ from the discrete uniform distribution, and is defined as:

$$D_\alpha(\mathbf{w}||\text{Uniform}) = \begin{cases} \sum_{c=1}^n \frac{(nw_c)^\alpha - 1}{n\alpha(\alpha-1)}, & \alpha \neq 0, \alpha \neq 1 \\ -\log(n) - \sum_{c=1}^n w_c \log(w_c), & \alpha = 1 \\ -n\log(n) - \sum_{c=1}^n \log(w_c), & \alpha = 0. \end{cases}$$

Note that when $\alpha = 1$, this is equivalent to the KL-Divergence and also the negative entropy $-H(\mathbf{w})$, $D_1(\mathbf{w}||\text{Uniform}) = D_{KL}(\mathbf{w}||\text{Uniform}) = -H(\mathbf{w}) + H(\text{Uniform})$, while for $\alpha = 0$ it is the reverse-KL-divergence $D_0(\mathbf{w}||\text{Uniform}) = D_{KL}(\text{Uniform}||\mathbf{w})$.

Two notable special cases are the KL-divergence ($\alpha = 1$), in which case (1) is the Maximum Entropy method, and Pearson's $\chi^2$-divergence ($\alpha = 2$), in which case (1) is a Quadratic Program (QP) and can be solved especially efficiently. Botev and Kroese (2011) particularly favor using $\alpha = 2$, and point out several advantages over other $f$-divergences (including the KL-divergence), such as computational efficiency, robustness to outliers, and the fact that minimizing this objective also guarantees an upper bound on both the KL-divergence and the $L_1$ (Total Variation) distance.

For the default values of the parameters $\alpha, \epsilon$ in (1), quipcell uses $\alpha = 2$ (following Botev and Kroese (2011)) and $\epsilon = 0$ (requiring exact equality for the monent constraint). We use these default settings for the examples in the remainder of this manuscript unless stated otherwise. In this case, (1) can be rewritten as the quadratic program

$$\hat{\mathbf{w}}_s = \arg\min_{\mathbf{w}} \quad \|\mathbf{w}\|_2^2 \tag{2a}$$

$$\text{s.t.} \quad \boldsymbol{\mu}_s = \mathbf{w}^T\mathbf{X}, \tag{2b}$$

$$\mathbf{w} \succeq 0, \tag{2c}$$

$$\|\mathbf{w}\|_1 = 1. \tag{2d}$$

## 2.1 Solution via convex optimization

$D_\alpha(\mathbf{w}||\text{Uniform})$ is convex in $\mathbf{w}$, so (1) can be efficiently solved via convex optimization, and model diagnostics can also be obtained by inspection of the Lagrangian dual solution.

Quipcell uses CVXPY (Diamond and Boyd, 2016) to solve (1) and fit $\hat{\mathbf{w}}_s$. The default case $\alpha = 2$ is a

Quadratic Program (QP) and can be solved particularly efficiently; CVXPY uses the OSQP solver (Stellato et al., 2020) by default in this case. In addition, the case $\alpha = 1$ (Maximum Entropy) can be solved efficiently and directly via its Lagrangian dual (Jaynes (1957); see also Section 5.5.5 of Boyd and Vandenberghe (2004)). For this case, quipcell uses scipy (Virtanen et al., 2020) and JAX (Bradbury et al., 2018) to run L-BFGS-B (Byrd et al., 1995) on the dual. For general $\alpha \notin \{1, 2\}$, quipcell uses CVXPY with the ECOS solver (Domahidi et al., 2013) by default; however, quipcell also allows users to specify alternative solvers and options to CVXPY for optimization.

Any solution to (1) is guaranteed to be unique, due to the strict convexity of $D_\alpha(\mathbf{w}||\text{Uniform})$. The strict convexity can be seen by noting that the first derivative is strictly increasing along any direction $\mathbf{z}$: defining $f(u) = \sum_c \frac{(w_c+uz_c)^\alpha}{\alpha(\alpha-1)}$, its second derivative is $f''(u) = \sum_c z_c^2(w_c + uz_c)^{\alpha-2}$, which is strictly positive on $\{u : \mathbf{w} + u\mathbf{z} \succeq 0\}$ Lebesgue almost everywhere.

The stability of the solution can be analyzed via the Lagrangian dual of (1). Locally, the dual solution $\lambda^*$ gives the gradient of the optimum with respect to the constraints (see Section 5.6.3 of Boyd and Vandenberghe (2004)). In particular, if a dual value corresponding to the $i$th coordinate of $\boldsymbol{\mu}_s$ is large, then small perturbations of the $i$th feature may cause large changes in the result. quipcell provides functionality for users to inspect the dual solution $\lambda^*$ to aid interpretation.

## 2.2 Linear transformations and canonical correlation features

In the prior exposition, we assumed that $\mathbf{X}$ and $\boldsymbol{\mu}_s$ represented normalized counts of $p$ genes. In practice $p$ can be large, and it may be desirable to use a lower dimensional representation to improve speed and prevent overfitting.

By linearity of expectation, we may replace $\mathbf{X}$ and $\boldsymbol{\mu}_s$ with any linear transformation $\mathbf{X}A^T$ and $A\boldsymbol{\mu}_s$, because $A\boldsymbol{\mu}_s = A\mathbb{E}_{\mathbf{x}\sim\mathbb{P}_s}[\mathbf{x}] = \mathbb{E}_{\mathbf{x}\sim\mathbb{P}_s}[A\mathbf{x}]$.

In our applications below, we set $A = A_2A_1$, where $A_1$ are PCA gene-loadings, so that $\mathbf{X}A_1^T$ gives the loadings of each cell along the principal components, and $A_2$ are loadings of a CCA between the PCA cell-loadings $\mathbf{X}A_1^T$ and some informative set of cell annotations $B \in \mathbb{R}^{n\times k}$. In our first application (Section 3.1), we set $B$ to be the indicator (one-hot) matrix of the celltype annotations from the Human Lung Cell Atlas. Note that the CCA of an indicator matrix is equivalent to the multiclass Linear Discriminant Analysis (Bach and Jordan, 2005). In our second example (Section 3.2), we set $B$ to be a cubic B-spline representation of the cell's pseudotime coordinate.

Note that quipcell only works on linear transformations of the normalized gene counts. In particular, the gene counts should **not** be log-transformed after normalization.

## 2.3 Correcting cell-size bias to estimate cell-level probabilities

As noted earlier, the weights $\hat{w}_{s,c}$ represent the probability of sampling a random read from $s$ at cell $c$. To see this, consider the case where $s$ is comprised of a subset of cells from $\mathbf{X}$, i.e. $s \subseteq \{1, , \ldots, n\}$. Let $\mathbf{Y}$ be the raw single cell gene counts, and $n_c = \sum_j Y_{c,j}$ the size factor for cell $c$. Let $X_{c,j} = \frac{1}{n_c}Y_{c,j}$ be the normalized gene counts. Then we have that $\boldsymbol{\mu}_s = \frac{1}{N_s}\sum_c \mathbb{I}_{c\in s}Y_c$, where $N_s = \sum_{c\in s} n_c$. If we then consider the optimization problem (1)

with $\boldsymbol{\mu}_s$ and $\mathbf{X}$, a feasible point is $w_{s,c} = \frac{n_c}{N_s}\mathbb{I}_{c\in s}$, which is the probability that a random read of $s$ comes from cell $c$. Therefore we interpret the inferred weights $\hat{w}_{s,c}$ as read-level rather than cell-level probabilities. The same argument holds if we replace $\mathbf{Y}, \mathbf{X}, \boldsymbol{\mu}_s$ with any linear transformation $\mathbf{Y}A^T, \mathbf{X}A^T, A\boldsymbol{\mu}_s$.

Since the weights $\hat{w}_{s,c}$ correspond to the probability of sampling reads rather than cells, they should only be used to compare relative abundances *between* samples, and cannot directly compare cell-type abundances *within* a single sample.

To enable within-sample comparisons, we can also obtain cell-level weights $\tilde{w}_{s,c}$ by renormalizing the read-level weights $\hat{w}_{s,c}$. In particular, if we have cell-specific size factors $m_c$, we can convert the *read*-level abundances $\hat{w}_{s,c}$ to *cell*-level abundances $\tilde{w}_{s,c}$ by

$$\tilde{w}_{s,c} \propto \frac{1}{m_c}\hat{w}_{s,c} \tag{3}$$

where the size factor $m_c$ is proportional to the number of UMIs coming from cell $c$.

Naively, we could simply set $m_c$ to be the number of UMIs from cell $c$; however, this can be problematic for heterogeneous reference sets, such as the Human Lung Cell Atlas (HLCA). For example, Figure A.2 shows that the different studies within the HLCA have very different sequencing depths and cell-type distributions.

We therefore use a Poisson GLM to estimate cell-specific size factors $m_c$ that account for different sequencing depths between batches or samples. In particular, we estimate the number of UMIs $n_c$ as a function of the cell's sample $S_c$ and its low-dimensional representation $XA^T$:

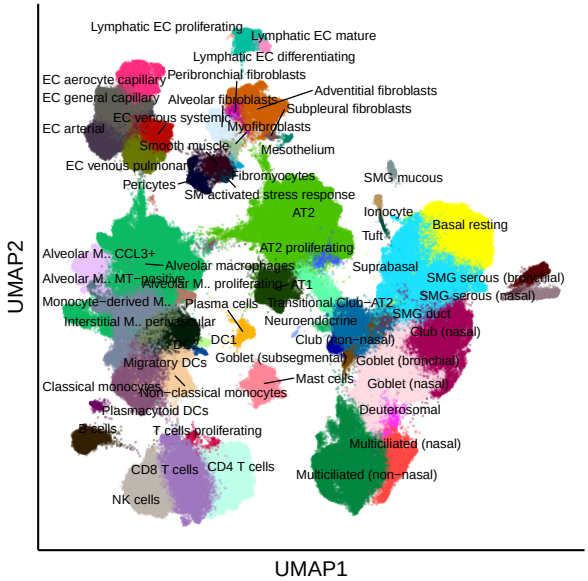
$$n_c \sim \text{Poisson}\left(\exp\left(\alpha_{S_c} + XA^T\beta\right)\right)$$

where we estimate the sample-specific normalization terms $\alpha_S$ and the feature coefficients $\beta$. We then set the cell-specific size factor $m_c$, after controlling for the sample sequencing depth, as

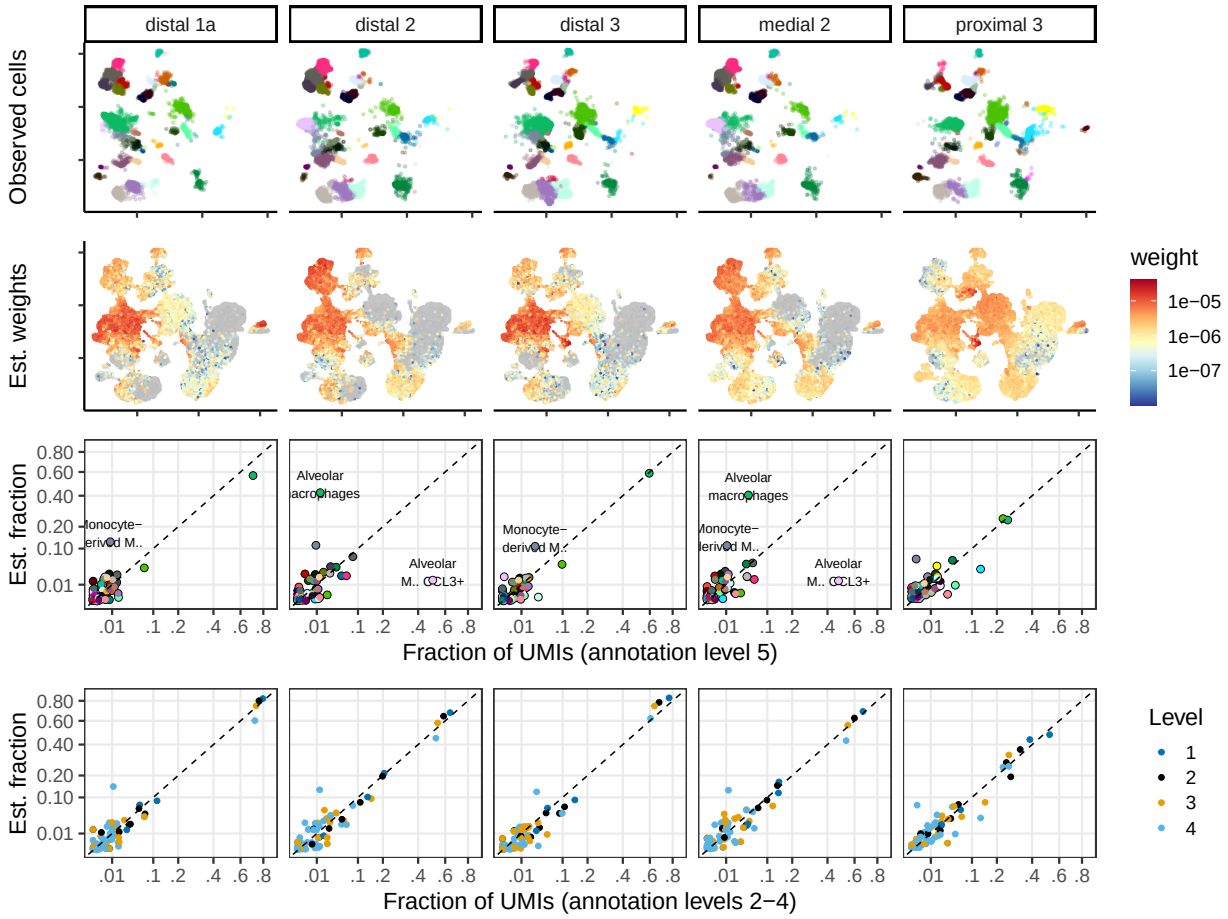
$$m_c = \exp\left(XA^T\beta\right).$$

We then use these cell-specific size factors $m_c$ to convert between read-level and cell-level abundances with equation (3).

Note that the renormalized cell-level weights may be less accurate than the read-level weights – see Figure 2 for a comparison, and Maden et al. (2023) for a discussion of challenges on converting read-level to cell-level abundances. See also Maden et al. (2024) for a recent method to correct cell-specific size factors in traditional (coarse-grained) bulk deconvolution.




(a) UMAP of the Human Lung Cell Atlas with the finest resolution (level 5) of celltype annotations.




(b) Deconvolution of pseudobulk samples from held-out validation study Travaglini et al. (2020).

Figure 1: Results from deconvolving the held-out validation study Travaglini et al. (2020) on the rest of the Human Lung Cell Atlas (Sikkema et al., 2023). (1a) UMAP and cell type labels downloaded from the original publication. (1b) Top row shows the cells from each of the 5 held-out samples. 2nd row shows estimated weights $\hat{w}_{s,c}$ for each sample; weights below 1e-8 were set to 0 and are colored gray. 3rd row shows the fraction of UMIs coming from each cell type (annotation level 5), vs the estimated fraction based on the weights $\hat{w}_{s,c}$. The x and y axes are square-root-scaled. Note quipcell has trouble distinguishing some macrophage subtypes, especially the "Alveolar macrophages" and "Alveolar Macrophage CCL3+" subtypes (see Appendix A.2). 4th row shows the accuracy of bulk deconvolution results at coarser resolutions of cell-type annotation (annotation levels 2-4).

# 3 Results

## 3.1 Application to Human Lung Cell Atlas

We applied quipcell to using the Human Lung Cell Atlas (HLCA) (Sikkema et al., 2023) as the reference single cell data set. It originally comprised 580K cells from 166 samples (107 donors) across 11 studies, but we held out the `Krasnow_2020` study (Travaglini et al., 2020) from the dataset as a validation set for bulk deconvolution, resulting in a reference set with 520K cells. For the cell-feature matrix $\mathbf{X}$, we used the loadings from the top 15 components of a multiclass Linear Discriminant Analysis (LDA), using the cell types from Sikkema et al. (2023) (shown in Figure 1a) as the class labels. The held-out `Krasnow_2020` study (Travaglini et al., 2020) consisted of 5 samples, from which we generated pseudobulk samples to benchmark bulk deconvolution with a known ground truth. See Appendix A.1 for more details about processing steps, and Figure 1a for a UMAP of the HLCA dataset.

We show the results of the deconvolution in Figure 1b. Overall, the estimated abundances are reasonable, though we note that the quipcell estimates become more accurate (closer to the diagonal) as the resolution becomes coarser (with annotation level 1 being the coarsest level of annotation, and annotation level 5 being the finest; see also Figure A.3). In particular, the method struggles to distinguish some macrophage subpopulations at the finest annotation resolutions, most notably "Alveolar macrophage CCL3+" vs. other Alveolar macrophages, and to a lesser extent "Interstitial macrophages" (more specifically "Monocyte-derived macrophages") from other macrophages. Part of the trouble stems from the fact that "Alveolar macrophage CCL3+" cells mainly appear in the validation set, and are relatively rare in the reference set (Table A.2), which causes none of the top 15 LDA components to separate these populations well (Figure A.5).

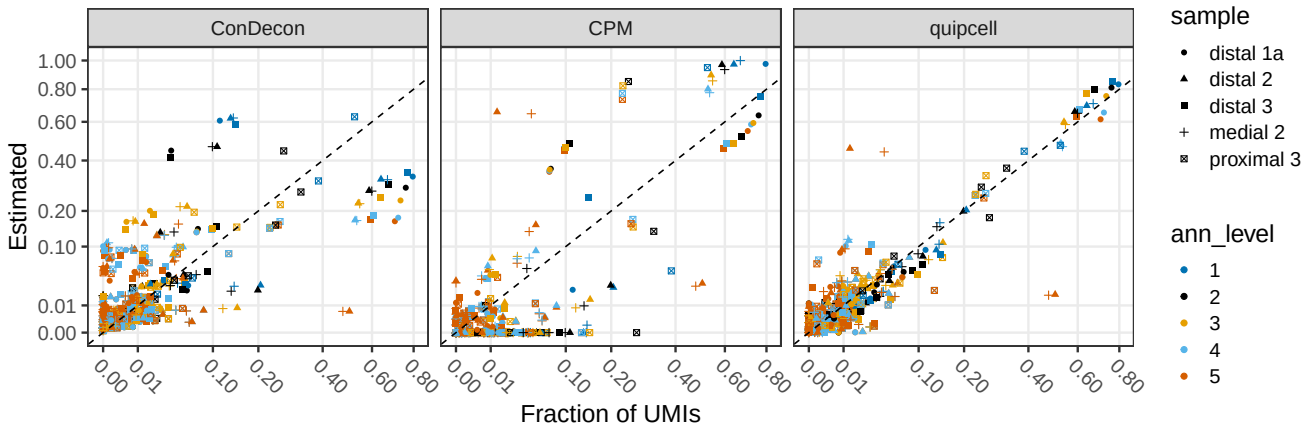
In Appendix A.3, we illustrate how examining the dual variables can be used to inspect model diagnostics in this example, and in Appendix A.4, we compare how the results change when using different $\alpha$-divergences.

### 3.1.1 Comparison with existing fine-resolution deconvolution methods

We compare our method with two existing fine-resolution deconvolution methods: CPM (Frishberg et al., 2019) and ConDecon (Aubin et al., 2023). We plot the accuracy of these methods in Figure 2, which shows our method to more accurately estimate abundances in general – in particular, notice that quipcell has points closer to the diagonal. Note that we ran all methods on a random $\frac{1}{8}$ of the reference set for this comparison, so that all methods could finish running with a reasonable amount of time and memory.

CPM (Frishberg et al., 2019) solves a similar QP as quipcell, but uses the gene counts directly (instead of a feature space like LDA), and also lacks the constraints (2c), (2d). This means the CPM-inferred weights don't sum to 1, and generally include negative weights. To compare cell-type proportion estimates, we summed the CPM weights by cell-type, set the negative cell-types to 0, and renormalized the positive cell-types to sum to 1. Note that setting celltypes with negative probability to 0 may be the culprit for the bottom horizontal stripe in the CPM panels of Figure 2. Note also that CPM does make use of a lower-dimensional embedding of the reference data, but only to subsample the reference single-cells according to a grid, and this embedding is

(a) True vs Estimated abundance by number of UMIs




(b) True vs Estimated abundance by number of cells

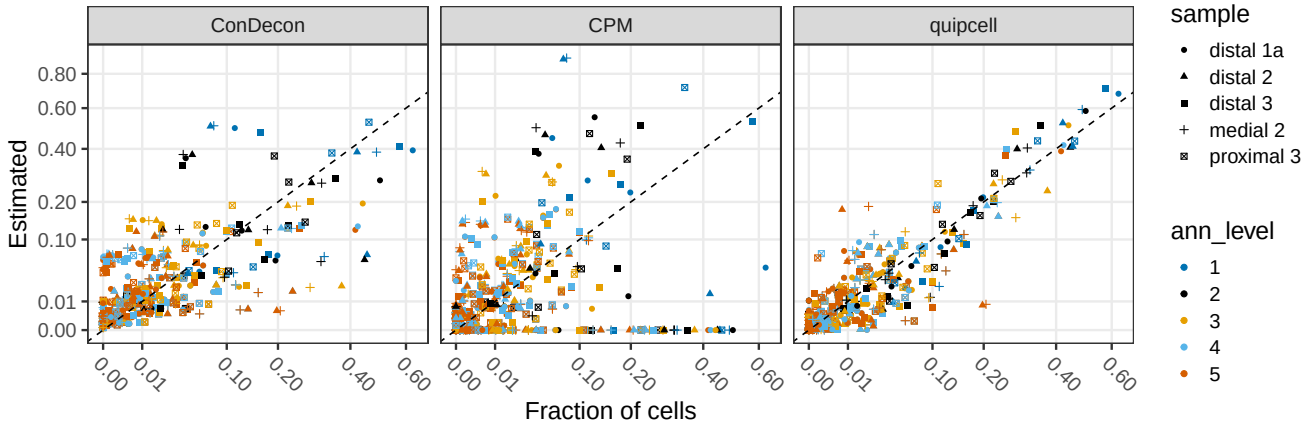



Figure 2: Comparison of our density estimation method with CPM (Frishberg et al., 2019) and ConDecon (Aubin et al., 2023). For each sample in the held-out validation study (Travaglini et al., 2020), we compare the estimated fraction of UMIs (top) or cells (bottom) coming from each celltype across several annotation levels (higher level means finer resolution). $x$ and $y$ axes are transformed by a square-root scaling.

required to be 1- or 2-dimensional. We used the original UMAP from the HLCA (Figure 1a) for this 2-dimensional embedding when running CPM.

ConDecon (Aubin et al., 2023) is a more recent high-resolution deconvolution method, that learns a map from the rank correlations between bulk and single-cell samples to a probability distribution on single cells. The method is trained on random resamplings of the single-cell training set. It also requires an embedding space of the single-cell data; we used the same 15-dimensional LDA embedding for this, as in quipcell. We ran ConDecon with default parameters.

Note that the QP of quipcell and CPM estimate UMI-level distributions, while ConDecon estimates cell-level distributions. However, the cell-level and UMI-level representations can be converted between each other by using cell-specific size factors (Section 2.3). We use the same cell size factors, estimated by the Poisson GLM of Section 2.3, for all 3 methods. In Figure 2, we show the accuracy of the three methods on both UMI-level and cell-level abundances.

| Method | Runtime | Memory usage |
|---|---|---|
| quipcell | 7 seconds | $< 16$ GB |
| CPM (Frishberg et al., 2019) | 55406 seconds (15.4 hours) | $< 16$ GB |
| ConDecon (Aubin et al., 2023) | 6654 seconds (1.85 hours) | 120-220 GB |

Table 1: Comparison of runtime and memory usage to perform deconvolution on the 5 validation samples. Memory usage is approximate: quipcell and CPM were run in jobs that were allocated 16 GB memory, but in fact use much less memory. ConDecon failed when run in a job allocated 120 GB memory, but succeeded when the memory limit was increased to 220 GB. Note that all methods were run on the same random $\frac{1}{8}$ subset of the reference for this comparison.

In terms of runtime, our QP-based method was more performant than either CPM or ConDecon, and ran in about 7 seconds for 5 bulk samples on a job allocated 16GB RAM (about the same as a laptop). By contrast, CPM took 15 hours to run, likely due to its lack of sparse matrix support. ConDecon took about 110 minutes to run, and also required significant memory usage (between 120-220GB RAM).

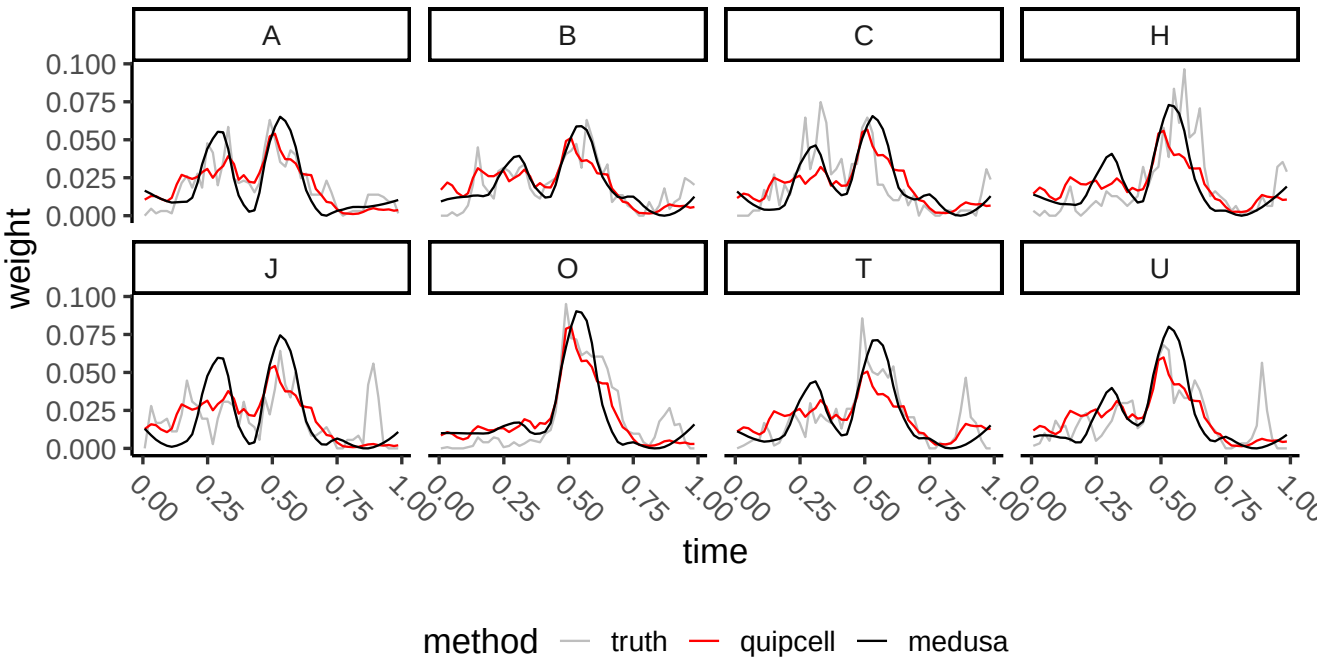
## 3.2 Application to pseudotime deconvolution

We next applied quipcell to the problem of deconvolving a bulk sample along a 1-dimensional pseudotime. We compared quipcell against the recent pseudotime deconvolution method MeDuSA (Song et al., 2023), on two datasets featured in the MeDuSA tutorial vignettes: a dataset of human stem cells (Chu et al., 2016), and a dataset of monocytes from human bone marrow (Oetjen et al., 2018). Both datasets contained matched single-cell and bulk samples, and we applied deconvolution to the bulk RNAseq samples, taking the matched single-cell samples as the ground truth.

In both cases, we used the pseudotime annotations provided with the MeDuSA vignette, and ran MeDuSA with the same settings as in the vignette (`markerGene=NULL`, `span=0.35`, `resolution=50`, `smooth=TRUE`, `fractional=TRUE`). To run quipcell, we used canonical correlation features derived from a CCA between the gene-expression matrix and a cubic B-spline basis of the pseudotime coordinate (computed with the R `fda` package, Ramsay and Silverman (2005)), and report the cell-size corrected weights (Section 2.3). In both cases, we used a basis of 16 splines and selected the top 3 CCA components (using more CCA components resulted in overly sparse weights).

Figure 3 shows the deconvolution results on the monocyte dataset, while Figure 4 shows the deconvolution results on the stem cell dataset. To compare quipcell and MeDuSA, we used the pseudotime bins output by MeDuSA, and computed the total quipcell weights within each bin, as well as the fraction of ground truth cells. We computed the summed absolute ($L_1$) and squared ($L_2$) errors of the predicted fractions against the ground truth fractions for both methods. In the monocyte dataset (Figure 3), quipcell generally had lower errors than MeDuSA, while in the stem cell dataset (Figure 4) the results were mixed, with MeDuSA being more accurate on some samples, and less accurate on others.

(a) True vs inferred weights along pseudotime.




(b) $L_1$ and $L_2$ errors summed across pseudotime bins.

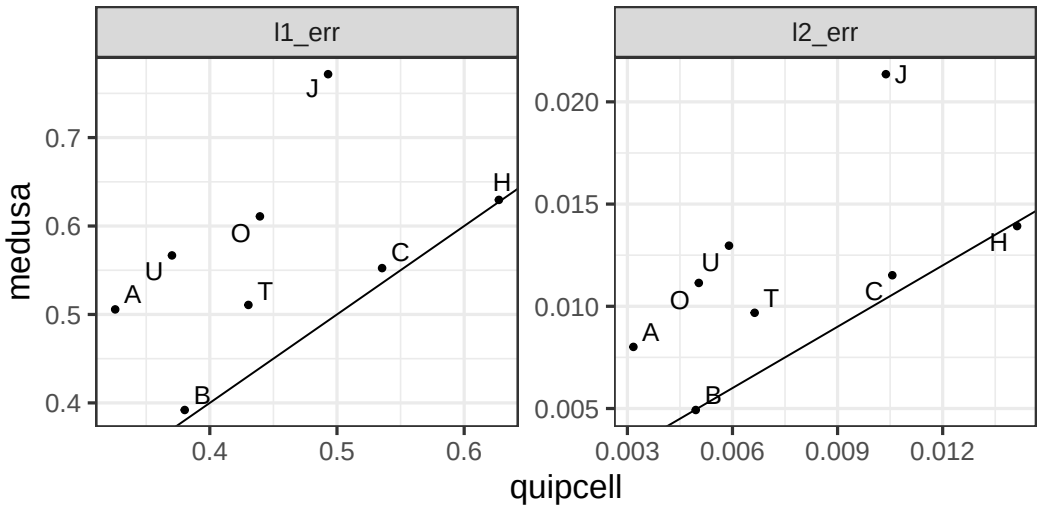


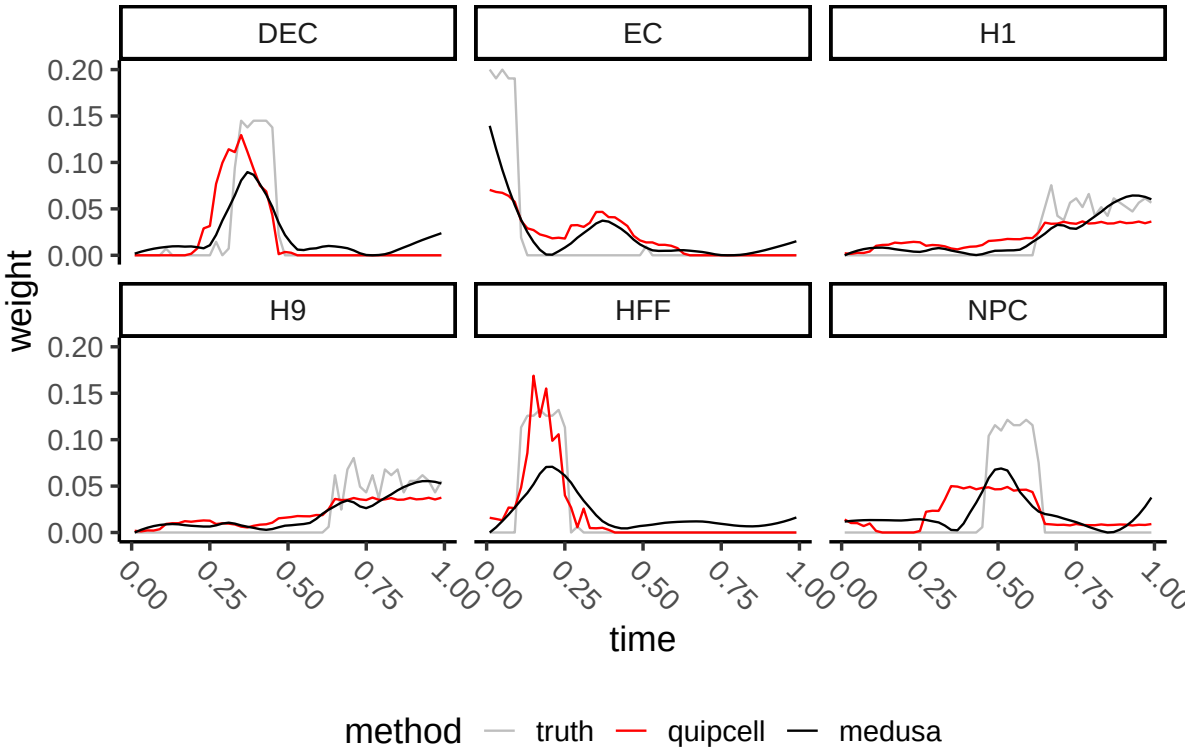

Figure 3: Comparison of quipcell and MeDuSA on the 8 bulk samples from the monocyte dataset.

# 4 Conclusions

We present quipcell, a distribution-fitting method for single-cell data based on moment-matching and convex optimization, which can be applied for bulk deconvolution analyses. When combined with Canonical Correlation Analysis (or its special case, Linear Discriminant Analysis), our method performs well on a range of datasets, including highly heterogeneous single cell atlases, and compares favorably to other high-resolution bulk deconvolution methods in both computational efficiency and predictive accuracy.

One direction for future work would be to apply the generalized entropy framework to the problem of

(a) True vs inferred weights along pseudotime.




(b) $L_1$ and $L_2$ errors summed across pseudotime bins.

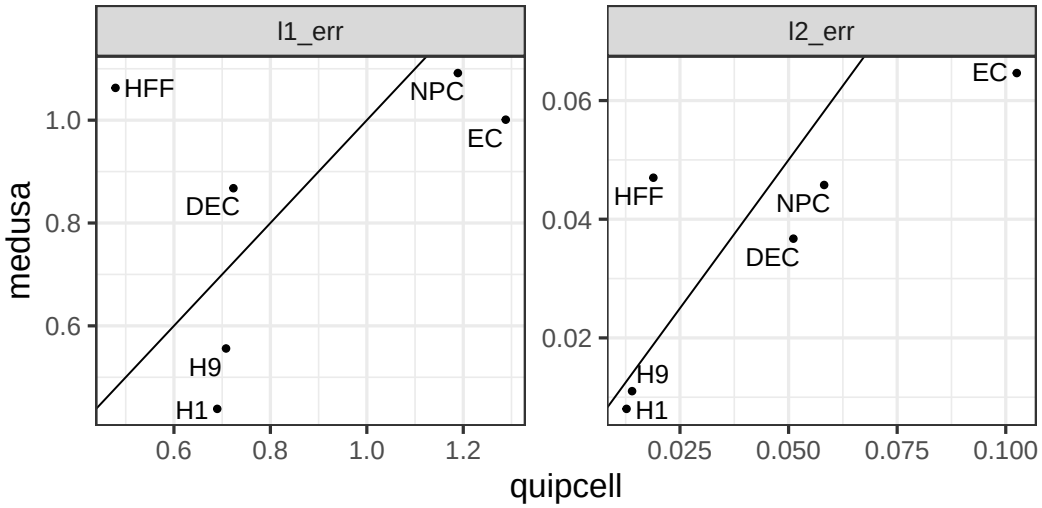



Figure 4: Comparison of quipcell and MeDuSA on the 6 bulk samples from the stem cell dataset.

differential abundance, i.e. estimating density changes *within* a single-cell dataset. While many existing fine-scale differential abundance methods exist (Burkhardt et al., 2021; Reshef et al., 2022; Dann et al., 2022; Zhao et al., 2021), they typically involve quantifying abundances along a nearest-neighbor graph; a generalized entropy approach would represent an alternative, graph-free approach based on information-theoretic principles. Unlike in the bulk deconvolution case, we would not be limited to linear transformations of the gene counts $\mathbf{X}$, since we could explicitly compute the expectation $\mathbb{E}_{\mathbf{x}\sim\mathbb{P}_s}[f(\mathbf{x})]$ even for nonlinear $f$. A notable special case would be to

use a Reproducing Kernel Hilbert Space as the feature space – then the results of Botev and Kroese (2011) show that the solution is a *continuous* density estimator based on a Gaussian mixture model, which is also equivalent to a sparse weighted kernel density estimator with data-driven bandwidth. If this kernel is furthermore a graph diffusion kernel, the estimate may become related to random-walk-based differential abundance methods such as CNA (Reshef et al., 2022) and MELD (Burkhardt et al., 2021).

# Acknowledgments

I thank Bill Forrest, Oleg Mayba, and Jayaram Kancherla for their invaluable feedback.

# Declaration of Interest

JK is employed by Genentech, Inc. and F. Hoffmann-La Roche Ltd.

# References

Altboum, Z., Steuerman, Y., David, E., Barnett-Itzhaki, Z., Valadarsky, L., Keren-Shaul, H., Meningher, T., Mendelson, E., Mandelboim, M., Gat-Viks, I., et al. (2014). Digital cell quantification identifies global immune cell dynamics during influenza infection. *Molecular systems biology*, 10(2):720.

Amezquita, R., Lun, A., Hicks, S., Gottardo, R., and O'Callaghan, A. (2023). Multi-sample single-cell analyses with bioconductor. `https://bioconductor.org/books/3.17/OSCA.multisample/`. [Online; accessed 2023-10-02].

Aubin, R. G., Montelongo, J., Hu, R., and Camara, P. G. (2023). Clustering-independent estimation of cell abundances in bulk tissues using single-cell rna-seq data. *bioRxiv*, pages 2023–02.

Bach, F. R. and Jordan, M. I. (2005). A probabilistic interpretation of canonical correlation analysis. *Technical Report 688, Department of Statistics, University of California, Berkeley.*

Biancalani, T., Scalia, G., Buffoni, L., Avasthi, R., Lu, Z., Sanger, A., Tokcan, N., Vanderburg, C. R., Segerstolpe, Å., Zhang, M., et al. (2021). Deep learning and alignment of spatially resolved single-cell transcriptomes with tangram. *Nature methods*, 18(11):1352–1362.

Botev, Z. I. and Kroese, D. P. (2011). The generalized cross entropy method, with applications to probability density estimation. *Methodology and Computing in Applied Probability*, 13.

Boyd, S. P. and Vandenberghe, L. (2004). *Convex optimization*. Cambridge university press.

Bradbury, J., Frostig, R., Hawkins, P., Johnson, M. J., Leary, C., Maclaurin, D., Necula, G., Paszke, A., VanderPlas, J., Wanderman-Milne, S., and Zhang, Q. (2018). JAX: composable transformations of Python+NumPy programs.

Burkhardt, D. B., Stanley III, J. S., Tong, A., Perdigoto, A. L., Gigante, S. A., Herold, K. C., Wolf, G., Giraldez, A. J., van Dijk, D., and Krishnaswamy, S. (2021). Quantifying the effect of experimental perturbations at single-cell resolution. *Nature biotechnology*, 39(5):619–629.

Byrd, R. H., Lu, P., Nocedal, J., and Zhu, C. (1995). A limited memory algorithm for bound constrained optimization. *SIAM Journal on scientific computing*, 16(5):1190–1208.

Chu, L.-F., Leng, N., Zhang, J., Hou, Z., Mamott, D., Vereide, D. T., Choi, J., Kendziorski, C., Stewart, R., and Thomson, J. A. (2016). Single-cell rna-seq reveals novel regulators of human embryonic stem cell differentiation to definitive endoderm. *Genome biology*, 17:1–20.

Csiszár, I. (1964). Eine informationstheoretische ungleichung und ihre anwendung auf beweis der ergodizitaet von markoffschen ketten. *Magyer Tud. Akad. Mat. Kutato Int. Koezl.*, 8:85–108.

Dann, E., Henderson, N. C., Teichmann, S. A., Morgan, M. D., and Marioni, J. C. (2022). Differential abundance testing on single-cell data using k-nearest neighbor graphs. *Nature Biotechnology*, 40(2):245–253.

De Boor, C. (1978). *A practical guide to splines*, volume 27. springer-verlag New York.

Diamond, S. and Boyd, S. (2016). Cvxpy: A python-embedded modeling language for convex optimization. *The Journal of Machine Learning Research*, 17(1):2909–2913.

Domahidi, A., Chu, E., and Boyd, S. (2013). Ecos: An socp solver for embedded systems. In *2013 European control conference (ECC)*, pages 3071–3076. IEEE.

Dong, M., Thennavan, A., Urrutia, E., Li, Y., Perou, C. M., Zou, F., and Jiang, Y. (2021). Scdc: bulk gene expression deconvolution by multiple single-cell rna sequencing references. *Briefings in bioinformatics*, 22(1):416–427.

Erdmann-Pham, D. D., Fischer, J., Hong, J., and Song, Y. S. (2021). Likelihood-based deconvolution of bulk gene expression data using single-cell references. *Genome research*, 31(10):1794–1806.

Frishberg, A., Peshes-Yaloz, N., Cohn, O., Rosentul, D., Steuerman, Y., Valadarsky, L., Yankovitz, G., Mandelboim, M., Iraqi, F. A., Amit, I., et al. (2019). Cell composition analysis of bulk genomics using single-cell data. *Nature methods*, 16(4):327–332.

Goulart, P. and Chen, Y. (2024). Clarabel.jl.

Hotelling, H. (1936). Relations between two sets of variates. *Biometrika*, 28(3-4):321–377.

Jaynes, E. T. (1957). Information theory and statistical mechanics. *Physical review*, 106(4):620.

Jew, B., Alvarez, M., Rahmani, E., Miao, Z., Ko, A., Garske, K. M., Sul, J. H., Pietiläinen, K. H., Pajukanta, P., and Halperin, E. (2020). Accurate estimation of cell composition in bulk expression through robust integration of single-cell information. *Nature communications*, 11(1):1971.

Maden, S. K., Huuki-Myers, L. A., Kwon, S. H., Collado-Torres, L., Maynard, K. R., and Hicks, S. C. (2024). lute: estimating the cell composition of heterogeneous tissue with varying cell sizes using gene expression. *bioRxiv*.

Maden, S. K., Kwon, S. H., Huuki-Myers, L. A., Collado-Torres, L., Hicks, S. C., and Maynard, K. R. (2023). Challenges and opportunities to computationally deconvolve heterogeneous tissue with varying cell sizes using single cell rna-sequencing datasets. *arXiv preprint arXiv:2305.06501*.

Newman, A. M., Steen, C. B., Liu, C. L., Gentles, A. J., Chaudhuri, A. A., Scherer, F., Khodadoust, M. S., Esfahani, M. S., Luca, B. A., Steiner, D., et al. (2019). Determining cell type abundance and expression from bulk tissues with digital cytometry. *Nature biotechnology*, 37(7):773–782.

Nguyen, H., Nguyen, H., Tran, D., Draghici, S., and Nguyen, T. (2024). Fourteen years of cellular deconvolution: methodology, applications, technical evaluation and outstanding challenges. *Nucleic Acids Research*, page gkae267.

Oetjen, K. A., Lindblad, K. E., Goswami, M., Gui, G., Dagur, P. K., Lai, C., Dillon, L. W., McCoy, J. P., and Hourigan, C. S. (2018). Human bone marrow assessment by single-cell rna sequencing, mass cytometry, and flow cytometry. *JCI insight*, 3(23).

Ramsay, J. O. and Silverman, B. W. (2005). *Functional Data Analysis*. Springer.

Rao, C. R. (1948). The utilization of multiple measurements in problems of biological classification. *Journal of the Royal Statistical Society. Series B (Methodological)*, 10(2):159–203.

Reshef, Y. A., Rumker, L., Kang, J. B., Nathan, A., Korsunsky, I., Asgari, S., Murray, M. B., Moody, D. B., and Raychaudhuri, S. (2022). Co-varying neighborhood analysis identifies cell populations associated with phenotypes of interest from single-cell transcriptomics. *Nature biotechnology*, 40(3):355–363.

Sikkema, L., Ramírez-Suástegui, C., Strobl, D. C., Gillett, T. E., Zappia, L., Madissoon, E., Markov, N. S., Zaragosi, L.-E., Ji, Y., Ansari, M., et al. (2023). An integrated cell atlas of the lung in health and disease. *Nature Medicine*, pages 1–15.

Song, L., Sun, X., Qi, T., and Yang, J. (2023). Mixed model-based deconvolution of cell-state abundances (medusa) along a one-dimensional trajectory. *Nature Computational Science*, pages 1–14.

Stellato, B., Banjac, G., Goulart, P., Bemporad, A., and Boyd, S. (2020). Osqp: An operator splitting solver for quadratic programs. *Mathematical Programming Computation*, 12(4):637–672.

Travaglini, K. J., Nabhan, A. N., Penland, L., Sinha, R., Gillich, A., Sit, R. V., Chang, S., Conley, S. D., Mori, Y., Seita, J., et al. (2020). A molecular cell atlas of the human lung from single-cell rna sequencing. *Nature*, 587(7835):619–625.

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., et al. (2020). Scipy 1.0: fundamental algorithms for scientific computing in python. *Nature methods*, 17(3):261–272.

Wang, X., Park, J., Susztak, K., Zhang, N. R., and Li, M. (2019). Bulk tissue cell type deconvolution with multi-subject single-cell expression reference. *Nature communications*, 10(1):380.

Wolf, F. A., Angerer, P., and Theis, F. J. (2018). Scanpy: large-scale single-cell gene expression data analysis. *Genome biology*, 19:1–5.

Zhao, J., Jaffe, A., Li, H., Lindenbaum, O., Sefik, E., Jackson, R., Cheng, X., Flavell, R. A., and Kluger, Y. (2021). Detection of differentially abundant cell subpopulations in scrna-seq data. *Proceedings of the National Academy of Sciences*, 118(22):e2100293118.

# A Appendix

## A.1 Processing steps for HLCA dataset

Here we describe the processing steps used for our analyses of the HLCA dataset.

First, we used scanpy (Wolf et al., 2018) to normalize each cell by its total UMI count, select highly variable genes (using study as the batch variable, and otherwise default parameters; 2176 genes selected) and perform PCA with 100 PCs. Note that we did not apply log-scaling of the data; therefore, the PCA is a linear transformation of the normalized counts.

Next, we split the dataset into two cohorts:

1. A hold-out "validation" cohort consisting of samples from the `Krasnow_2020` study (Travaglini et al., 2020), used to generate pseudobulk samples $\boldsymbol{\mu}_s$ for deconvolution. We chose to hold out this study because it contained the fewest samples (5) while still having a diversity of cell types, making it easier to visualize the full results as in Figure 1b.
2. A "reference" cohort consisting of the remaining cells. The weights $\hat{w}_{s,c}$ are computed on cells $c$ coming from this reference cohort.
   (a) For speed reasons, some analyses (Figure 2, Figure A.6), use a smaller reference consisting of a random $\frac{1}{8}$ of the full reference.

We list the number of cells, samples, and studies in each subset in Table A.1.

After excluding the hold-out validation cohort, we performed multiclass Linear Discriminant Analysis on the 100 PCs, using the HLCA cell types as the class labels. After examining a plot of the variances along each LDA component, we kept the top 15 components of the LDA (Figure A.1).

For the quadratic program (2), we used the top 15 LDA components of the reference cohort as the reference matrix $\mathbf{X}$. For the validation cohort, we first generate pseudobulk samples by summing gene counts across cells, then normalize the summed counts and successively apply the PCA and LDA rotation matrices to obtain $\boldsymbol{\mu}_s$.

| | Cells | Samples | Donors | Studies |
|---|---|---|---|---|
| Full | 584884 | 166 | 107 | 11 |
| Validation set (`Krasnow_2020`) | 60977 | 5 | 3 | 1 |
| Reference set (full) | 523,907 | 161 | 104 | 10 |
| Reference set (random $\frac{1}{8}$ subset) | 65369 | 161 | 104 | 10 |

Table A.1: Table of number of cells, samples, and studies in each of the data subsets. The validation set consists of one held-out study, and is used to benchmark bulk deconvolution. For some analyses (Figure 2, Figure A.6), we only use a random $\frac{1}{8}$ of the reference set to improve speed.

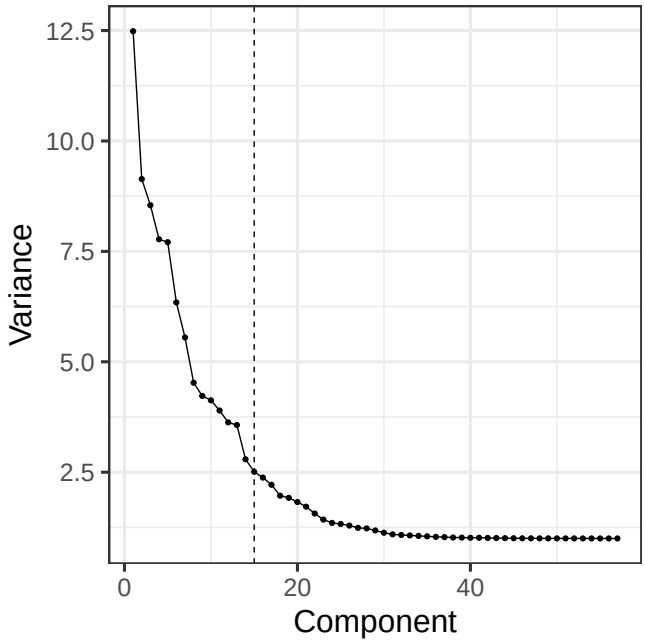



Figure A.1: Variance of the single cell data along each orthogonal component of LDA. The held-out validation set is **not** included in the variance computation. The variance shows a large drop between the 13th and 14th LDA components. For this paper, we chose 15 LDA components for all downstream analyses.

## A.2 Number of Alveolar macrophages vs Alveolar macrophage CCL3+ cells

At the finest resolution of celltype annotation, the bulk deconvolution results of Figure 1b were unable to distinguish the "Alveolar macrophage CCL3+" subpopulation from "Alveolar macrophages". This was in part due to the rarity of "Alveolar macrophage CCL3+" in the reference set, which causes none of the top 15 LDA components to separate these populations well (Figure A.5). See Table A.2 for the number of cells from these subpopulations in each cohort.

| Cohort | Alveolar macrophages | Alveolar macrophage CCL3+ |
|---|---|---|
| Validation (`Krasnow_2020`) | 8597 (14%) | 5785 (9.4%) |
| Reference | 59890 (11.4%) | 2280 (0.4%) |

Table A.2: Number of cells with celltype "Alveolar macrophages" or "Alveolar macrophage CCL3+". Our bulk deconvolution results were unable to distinguish these 2 subpopulations (Figure 1b), assigning both cell types as "Alveolar macrophages". This is in part due to the rarity of "Alveolar macrophage CCL3+" in the reference set (291 cells, or 0.4%), in comparison to the validation set (5785 cells, or 9.4%).

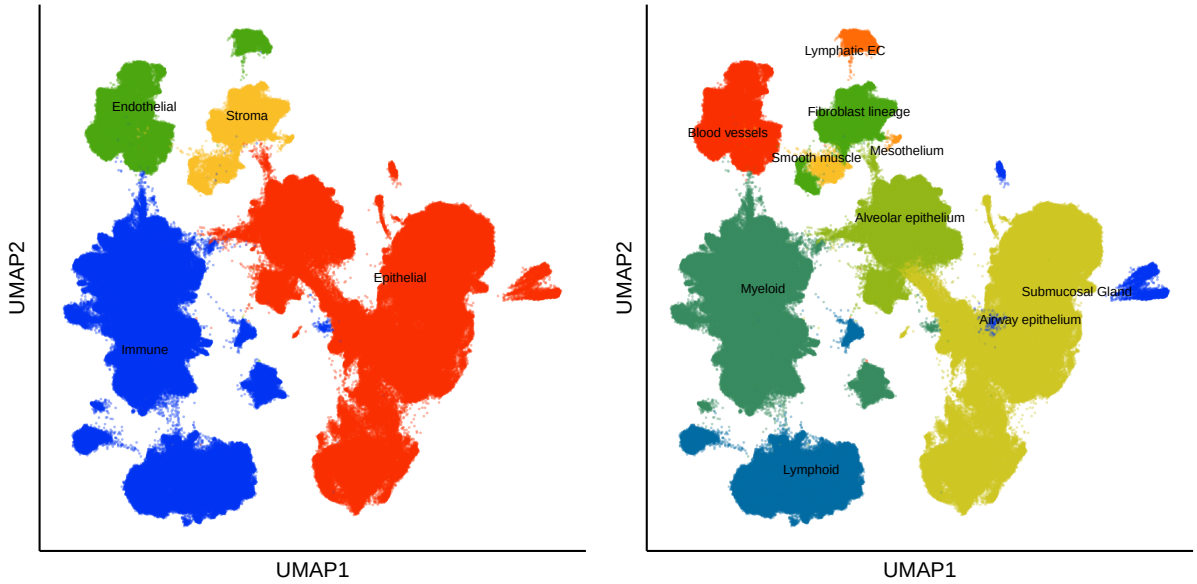
Figure A.2: Number of UMIs per cell across the studies in the HLCA dataset. The studies have different sequencing depths and contain different cell types, which complicates estimating cell-specific size factors to convert between read-level and cell-level abundances in bulk deconvolution.

## A.3 Using Lagrangian dual solution for model diagnostics

In Figure A.4, we illustrate how the Lagrangian dual solution can be used to inspect the influence of each moment constraint. For example, the dual variable for the 14th Linear Discriminant is large, especially in the sample "distal 3". That discriminant separates monocytes from macrophages (Figure A.5), which suggests that small perturbations of the bulk counts may cause the inferred weights to shift between these types of myeloid cells.

## A.4 Comparison of $\alpha$ values on deconvolution results

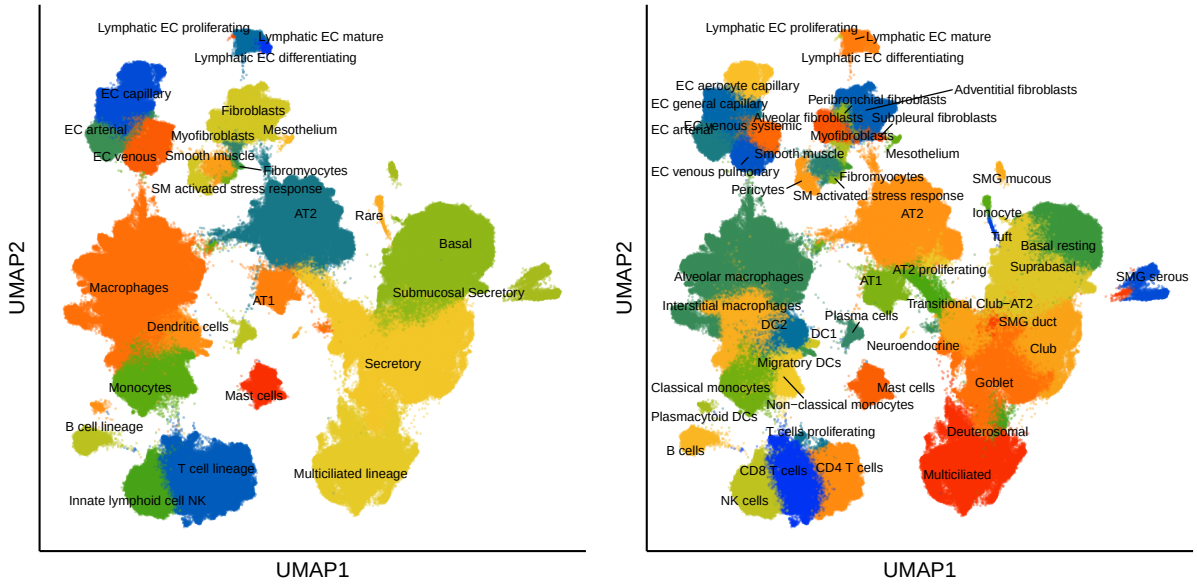
In Figure A.6, we compare deconvolution results on the HLCA example (using the downsampled $\frac{1}{8}$ version of the reference, Table A.1) for $\alpha$-divergences with $\alpha = 1, 1.5, 2$. Note that $\alpha = 1$ corresponds to Maximum Entropy, while $\alpha = 2$ corresponds to the Pearson divergence. $\alpha = 2$ yields the sparsest solution, while $\alpha = 1$ yields the least sparse solution (and indeed, Maximum Entropy will never return probabilities of zero, see Jaynes (1957)). We also note that $\alpha = 1.5$ was the most challenging to fit, both in runtime (taking 3 minutes, compared to about 5 seconds for Maximum Entropy and Pearson divergence), as well as in robustness – CVXPY's default ECOS solver failed to converge, so we used the Clarabel solver (Goulart and Chen, 2024) instead (though note that




(a) Annotation level 1




(b) Annotation level 2

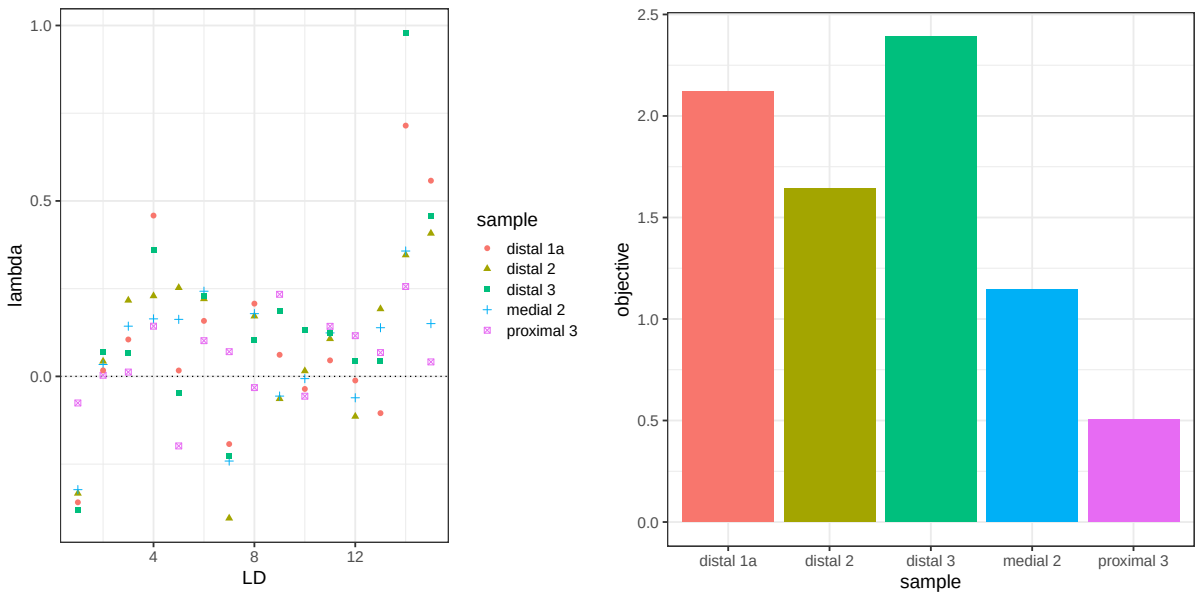



(c) Annotation level 3




(d) Annotation level 4

Figure A.3: HLCA celltypes at annotation resolutions 1-4.




(a) Dual solution $\lambda^*$ at each linear discriminant.

(b) Objective function at the optimum.

Figure A.4: Dual variables and objective at the optimum for the bulk deconvolution example. The dual variables are the gradient of the objective and can be used to identify which features have a strong influence on the solution.

CVXPY has announced it will be switching to Clarabel as the default solver in version 1.5).

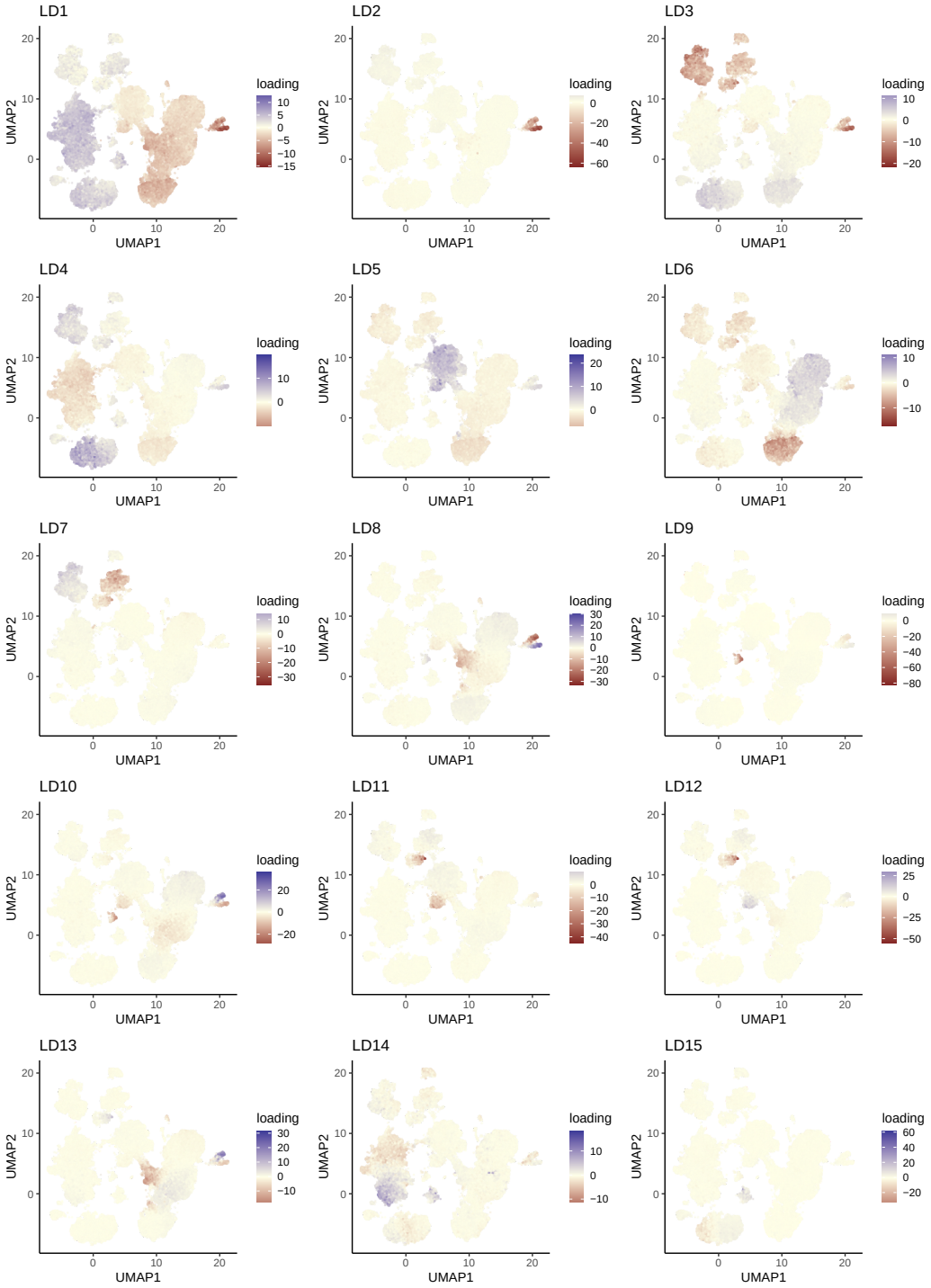



Figure A.5: Cell loadings of each Linear Discriminant on the reference set.

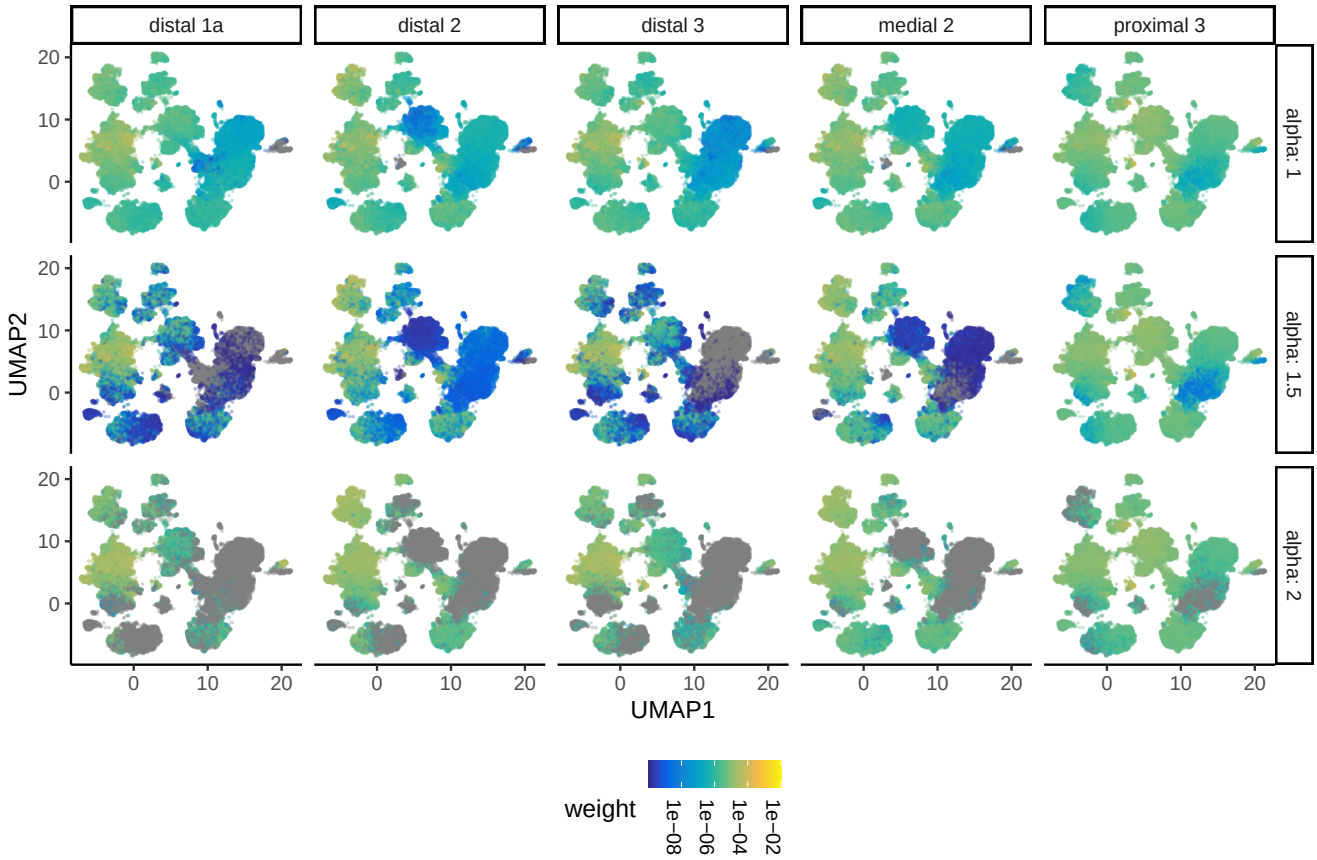



Figure A.6: Bulk deconvolution results for the HLCA data for different values of $\alpha$. For visualization purposes, weights below 1e-10 are set to 0, and appear as grey points.