Adaptive Regularized Tri-Factor Non-Negative Matrix Factorization for Cell Type Deconvolution

Tianyi Liu[1] Quefeng Li[1,*] Xiaojing Zheng[1,2,*] Fei Zou[1,3,*]

[1]Department of Biostatistics, The University of North Carolina at Chapel Hill

[2]Department of Pediatrics, The University of North Carolina at Chapel Hill

[3]Department of Genetics, The University of North Carolina at Chapel Hill

Abstract

Accurate deconvolution of cell types from bulk gene expression is a fundamental step in characterizing cellular compositions, shedding light on cell-type specific differential expression and physiological states of diseased tissues. However, existing deconvolution methods either require complete cellular gene expression signatures or are entirely reference-free, neglecting partial biological information. Moreover, these methods often disregard varying cell-type mRNA amounts, resulting in biased proportion estimates. Additionally, they fail to utilize useful reference information from external studies such as population cell-type proportions. To address these challenges, we propose an Adaptive Regularized Tri-factor non-negative matrix factorization method for deconvolution, known as ARTdeConv. We rigorously prove the numerical convergence of the algorithm. In benchmark simulations, we demonstrate that ARTdeConv can surpass the performance of state-of-the-art reference-free methods. In addition, estimated proportions by ARTdeConv show a nearly perfect Pearson's correlation against flow cytometry measurements in a real dataset obtained from a trivalent influenza vaccine study. An R package for our method is available for access on GitHub.

Keywords: Cell-type deconvolution, Convergence analysis, Multiplicative update algorithm, Non-negative matrix factorization

1. Introduction

Heterogeneity in cell-type proportions exists across biological samples. Neglecting this heterogeneity in bulk gene expression can introduce bias into subsequent analyses such as differential expression analyses. Conversely, acknowledging and accounting for this heterogeneity has shown clear advantages, yielding more accurate in survival time predictions and tumor type classifications (Elloumi et al., 2011; Avila Cobos et al., 2020).

*Co-corresponding authors

While laboratory techniques like flow cytometry or immunohistochemistry are available for physically sorting cells into cell types and quantifying their abundances, they are often limited by the availability of cell samples, the particularity of antibodies to separate cells, as well as labor and time investments (Newman et al., 2015; Spitzer and Nolan, 2016). Cell type deconvolution (referred to as "deconvolution" in this article), a computational process aimed at digitally separating heterogeneous mixture signals into their constituent components, has been critical in expediting the estimation of cell-type proportions from bulk gene expression data, such as RNA sequencing (RNA-seq). Recent years have seen the emergence of several deconvolution methods extensive applications in the field of computational biology. These methods can generally be categorized into two groups: reference-based and reference-free, depending on whether they require individual cell-type gene expression signatures, referred to as a signature matrix, as prior knowledge (Li and Wu, 2019; Avila Cobos et al., 2020; Jin and Liu, 2021).

Gene signatures can be derived from either single-cell RNA-seq (scRNA-seq) data or sorted bulk RNA profiles of individual cell types (Chen et al., 2018a). Although gene signatures have been successfully established for some cell types, acquiring those for other cell types might be labor-intensive or even unfeasible (Morris and Kopetz, 2016; Wigerblad et al., 2022). Neglecting to incorporate gene signatures for a prevalent cell type can induce biases in the proportional estimates of other cell types within reference-based deconvolution (Avila Cobos et al., 2020). On the other hand, reference-free techniques enable the unsupervised estimation of cell-type proportions, but at the cost of disregarding information embedded in known gene signatures (Zaitsev et al., 2019; Chen et al., 2020). A deconvolution method that can utilize partial reference gene signatures presents a plausible compromise.

Moreover, external information on the population cell-type proportions may also help with estimating cell-type proportions. For example, in blood samples, the proportion medians (or means) and ranges of major cell types can are available from complete blood count (CBC) tests. However, deconvolution techniques that effectively incorporate such information are currently lacking in the field.

Many deconvolution methods operate under the assumption of a generative model where the bulk gene expression matrix equals the product of the gene signature and proportion matrices, plus some stochastic errors (Avila Cobos et al., 2020). However, a critical factor, cell-type mRNA amount, denoting the quantity of RNA molecules contained within an individual cell, is lost after library normalization. Zaitsev et al. (2019) has found that failing to account for mRNA amounts also produce biased cell-type proportion estimates. Therefore, incorporating them into the generative model can be pivotal.

In this article, we propose an *A*daptive *R*egularized *T*ri-factor non-negative matrix factorization method for *deConv*olution (ARTdeConv). It addresses the three outstanding issues discussed above: utilizing partial reference gene signatures, incorporating external information on cell-type proportions, and accounting for cell-type mRNA amounts. Our contribution includes deriving and implementing a multiplicative update (MU) algorithm for solving ARTdeConv, proving the algorithm's convergence, and demonstrating its merits by simulations and real data analysis. An R package implementing ARTdeConv is also available on GitHub.

The rest of this article is laid out as follows. The ARTdeConv method, as well as the MU algorithm, are described in Section 2. The algorithm's convergence properties are studied in Section 3. Simulation and real data results are presented in Sections 4 and 5. Technical details of the method and proofs are relegated to the Web Appendix.

2. Model

2.1. Notation

Let $\mathbb{R}_+$ and $\mathbb{R}_{++}$ denote the set of non-negative and positive real numbers respectively. $m, n, K \in \mathbb{R}_{++}$ are used to denote the number of genes, samples, and cell types with $K \leq \min(m, n)$ in deconvolution. Let $K_0 < K$ be the number of cell types for which we have reference gene expression available. Let $\boldsymbol{Y} \in \mathbb{R}_+^{m \times n}$ be the matrix of bulk gene expression and $\boldsymbol{\Theta} \in \mathbb{R}_+^{m \times K}$ be the full gene signature matrix. Let $\boldsymbol{s} \in \mathbb{R}_{++}^K$ be a vector of relative cell-type mRNA amounts, and $\boldsymbol{P} \in \mathbb{R}_+^{K \times n}$ the proportion matrix. Let $\boldsymbol{\epsilon} \in \mathbb{R}^{m \times n}$ be the random error matrix. Unless otherwise specified, we denote $\boldsymbol{\theta}_k$ as the k-th column of $\boldsymbol{\Theta}$. We use $\boldsymbol{A}^\top$ to denote the transpose of $\boldsymbol{A}$. For any $m \times n$ matrix $\boldsymbol{A}$, its Frobenius norm is denoted as $\|\boldsymbol{A}\|_F = \sqrt{\mathrm{tr}(\boldsymbol{A}^\top \boldsymbol{A})} = \sqrt{\sum_{i=1}^m \sum_{j=1}^n A_{ij}^2}$. For two matrices $\boldsymbol{A}, \boldsymbol{B}$ of the same dimensions, $\boldsymbol{A} \odot \boldsymbol{B}$ denotes the element-wise product of $\boldsymbol{A}$ and $\boldsymbol{B}$ and $\frac{\boldsymbol{A}}{\boldsymbol{B}}$ denotes their element-wise quotient. For two square matrices, we write $\boldsymbol{A} \succ \boldsymbol{B}$ and $\boldsymbol{A} \succeq \boldsymbol{B}$ if $\boldsymbol{A} - \boldsymbol{B}$ is positive-definite and positive-semidefinite, respectively.

2.2. Tri-factor Model

We first propose the following tri-factor generative model

$$\boldsymbol{Y} = \boldsymbol{\Theta}\mathrm{diag}(\boldsymbol{s})\boldsymbol{P} + \boldsymbol{\epsilon}, \tag{1}$$

which extends the canonical model $\boldsymbol{Y} = \boldsymbol{\Theta}\boldsymbol{P} + \boldsymbol{\epsilon}$ (Avila Cobos et al., 2020). The difference between them is $\mathrm{diag}(\boldsymbol{s})$, a diagonal matrix of cell-type mRNA amounts. In practice, normalization, for example by library sizes, is frequently employed to alleviate between-sample technical artifacts in sequencing data (Hafemeister and Satija, 2019; Lytal et al., 2020). During this process, information regarding the quantities of RNA molecules packed within different kinds of cells sometimes gets lost. Failures to recover these quantities lead to documented biases in estimating cell-type proportions (Zaitsev et al., 2019). Should this occurs, $\mathrm{diag}(\boldsymbol{s})$ can account for the RNA molecule quantities. Otherwise, $\mathrm{diag}(\boldsymbol{s})$ would be close to the identity matrix $\boldsymbol{I}_K$. We notice that (1) is similar to the model of MuSiC, a fully reference-based method by Wang et al. (2019), where each sample is accorded an additional scaling factor c_i to account for between-sample measurement variation in bulk gene expression.

For performing deconvolution based on (1), ARTdeConv assumes prior knowledge of a partial signature matrix $\boldsymbol{\Theta}_{K_0} \in \mathbb{R}_+^{m \times K_0}$ as well as $\boldsymbol{Y}$. Then, an objective function is defined:

$$f(\boldsymbol{\Theta}, \boldsymbol{s}, \boldsymbol{P}) = \frac{1}{2mn}\|\boldsymbol{Y} - \boldsymbol{\Theta}\mathrm{diag}(\boldsymbol{s})\boldsymbol{P}\|_F^2 + \frac{\alpha_1}{2}R_1 + \frac{\alpha_2}{2}R_2 + \frac{\beta}{2}R_3, \tag{2}$$

where R_1, R_2, and R_3 are regularizers to be explained later and α_1, α_2, and β are their tuning parameters. We discuss the selection of tuning parameter values in Section 2.4. $\boldsymbol{\Theta}$, $\boldsymbol{s}$, and $\boldsymbol{P}$ are then estimated via

$$(\hat{\boldsymbol{\Theta}}, \hat{\boldsymbol{s}}, \hat{\boldsymbol{P}}) = \underset{\boldsymbol{\Theta} \in \mathbb{R}_+^{m \times K},\ \boldsymbol{s} \in \mathbb{R}_{++}^K,\ \boldsymbol{P} \in \mathbb{R}_+^{K \times n}}{\mathrm{argmin}} f(\boldsymbol{\Theta}, \boldsymbol{s}, \boldsymbol{P}). \tag{3}$$

The main objective here is to obtain an estimate of the proportion matrix $\hat{\boldsymbol{P}}$. To follow a common practice of deconvolution and simplify the algorithm (Mohammadi et al.,

2017), ARTdeConv does not directly constrain each column of $\boldsymbol{P}$ to sum to one during the estimation process. Instead, it obtains an unconstrained estimate $\hat{\boldsymbol{P}}$ and then re-normalizes its columns to have the unit sum.

Let $\boldsymbol{\Theta}_0 = \begin{bmatrix}\boldsymbol{\Theta}_{K_0} & \mathbf{0} & \cdots & \mathbf{0}\end{bmatrix} \in \mathbb{R}_+^{m\times K}$ and $\boldsymbol{\Delta}$ be an $m \times K$ matrix such that $\Delta_{jk} = I(k \leq K_0)$ for $1 \leq j \leq m$. The squared Frobenius distance between the estimated $\boldsymbol{\Theta}$ and $\boldsymbol{\Theta}_0$ is penalized through $R_1 = \|\boldsymbol{\Delta} \odot (\boldsymbol{\Theta} - \boldsymbol{\Theta}_0)\|_F^2$.

Though we present a special case where $\boldsymbol{\Theta}_{K_0}$ occupies the first K_0 columns of $\boldsymbol{\Theta}$, by re-defining $\boldsymbol{\Delta}$ correspondingly, $\boldsymbol{\Theta}_{K_0}$ can occupy any K_0 columns, covering all structures of prior knowledge on the signature matrix.

On the other hand, R_1 is not a strictly convex function of $\boldsymbol{\Theta}$, making the guarantee of the numerical convergence of ARTdeConv's algorithm difficult (more details on this later). To atone for this, we make $R_2 = \|(\boldsymbol{J}_{m\times K} - \boldsymbol{\Delta}) \odot \boldsymbol{\Theta}\|_F^2$,

where $\boldsymbol{J}_{m\times K}$ is an $m \times K$ matrix of 1s. It can be shown that any positive linear combination of R_1 and R_2 is strictly convex with respect to $\boldsymbol{\Theta}$.

Information on the cell-type proportions such as medians (or means) and ranges in a population frequently exists in external data and is accessible online, for example through complete blood counts of leukocytes (Takami et al., 2021). To incorporate such information into our deconvolution method, let the reference median or mean of cell type k's proportion be written as m_k and its range as r_k. We propose adding another penalty $R_3 = \sum_{k=1}^{K}(1/r_k)\|\boldsymbol{p}_k^\top - m_k\mathbf{1}_n\|_2^2$,

where $\boldsymbol{p}_k^\top$ is the k-th row of $\boldsymbol{P}$. In R_3, m_k acts as a pivot, and the deviation from which is penalized for the estimated $\boldsymbol{p}_k^\top$. Meanwhile, $1/r_k$ acts as a weight so the proportion estimates of cell types with wider ranges are less penalized for departing from their respective m_k. Letting $\boldsymbol{M}$ be a $K\times n$ matrix such that $\boldsymbol{M} = \begin{bmatrix}m_1\mathbf{1}_n & m_2\mathbf{1}_n & \cdots & m_K\mathbf{1}_n\end{bmatrix}^\top$ and $\boldsymbol{\rho} = \mathrm{diag}(r_1, r_2, \cdots, r_K)$, we can write R_3 in a matrix form $R_3 = \|\boldsymbol{\rho}^{-1/2}(\boldsymbol{P} - \boldsymbol{M})\|_F^2$.

2.3. Multiplicative Update Algorithm

A multiplicative update (MU) algorithm is proposed to solve (3). The MU algorithm was originally designed to solve the canonical bi-factor non-negative matrix factorization (NMF) problem in Lee and Seung (2000) and was extensively studied by Lin (2007). Technically, it is similar to the majorization-minimization algorithm of Lange (2016). It can be readily extended to solving a multi-factor regularized NMF problem like the one for ARTdeConv.

Let t be a non-negative integer denoting the current number of iteration. The multiplicative update steps are derived by finding the gradients of a set of auxiliary functions for each row of $\boldsymbol{\Theta}^t$, each column of $\boldsymbol{P}^t$, and $\boldsymbol{s}^t$. These update steps are:

$$\boldsymbol{\Theta}^{t+1} = \boldsymbol{\Theta}^t \odot \frac{\boldsymbol{Y}(\boldsymbol{P}^t)^\top \mathrm{diag}(\boldsymbol{s}^t) + mn\alpha_1\boldsymbol{\Delta} \odot \boldsymbol{\Theta}_0}{\boldsymbol{\Theta}^t\mathrm{diag}(\boldsymbol{s}^t)\boldsymbol{P}^t(\boldsymbol{P}^t)^\top\mathrm{diag}(\boldsymbol{s}^t) + mn\left\{\alpha_1\boldsymbol{\Delta} \odot \boldsymbol{\Theta}^t + \alpha_2(\boldsymbol{J}_{m\times K} - \boldsymbol{\Delta}) \odot \boldsymbol{\Theta}^t\right\}}; \tag{4}$$

$$\boldsymbol{P}^{t+1} = \boldsymbol{P}^t \odot \frac{\mathrm{diag}(\boldsymbol{s}^t)(\boldsymbol{\Theta}^{t+1})^\top\boldsymbol{Y} + mn\beta\boldsymbol{\rho}^{-1}\boldsymbol{M}}{\mathrm{diag}(\boldsymbol{s}^t)(\boldsymbol{\Theta}^{t+1})^\top\boldsymbol{\Theta}^{t+1}\mathrm{diag}(\boldsymbol{s}^t)\boldsymbol{P}^t + mn\beta\boldsymbol{\rho}^{-1}\boldsymbol{P}^t}; \tag{5}$$

$$\boldsymbol{s}^{t+1} = \boldsymbol{s}^t \odot \frac{\boldsymbol{u}^t}{\boldsymbol{Z}^t\boldsymbol{s}^t}, \tag{6}$$

where $u_k^t = \mathrm{tr}\left\{\boldsymbol{Y}^\top\boldsymbol{\theta}_k^{t+1}\left(\boldsymbol{p}^{t+1}\right)_k^\top\right\}$ and $Z_{kl}^t = \left\{(\boldsymbol{\theta}^{t+1})_k^\top\boldsymbol{\theta}_l^{t+1}\right\}\left\{(\boldsymbol{p}^{t+1})_k^\top\boldsymbol{p}_l^{t+1}\right\}$ for $k, l = 1, 2, \cdots, K$.

Mathematical details for finding the auxiliary functions and the update steps can be found in Web Appendix A. The pseudo-code of the MU algorithm is shown in Algorithm 1. The MU algorithm has two advantages. First, under assumptions discussed later in

Algorithm 1 The MU Algorithm For Solving ARTdeConv

Initialize $\boldsymbol{\Theta}^0 \in \mathbb{R}_+^{m\times K}$ such that $R_1 = 0, \boldsymbol{P}^0 \in \mathbb{R}_+^{K\times n}, \boldsymbol{s}^0 \in \mathbb{R}_{++}^K$. Fix α_1, α_2, and $\beta > 0$.

Repeat

Update $\boldsymbol{\Theta}^t$ to $\boldsymbol{\Theta}^{t+1}$ via (4);

Update $\boldsymbol{P}^t$ to $\boldsymbol{P}^{t+1}$ via (5);

Update $\boldsymbol{s}^t$ to $\boldsymbol{s}^{t+1}$ via (6);

Until

$$\frac{|f(\boldsymbol{\Theta}^{t+1}, \boldsymbol{s}^{t+1}, \boldsymbol{P}^{t+1}) - f(\boldsymbol{\Theta}^t, \boldsymbol{s}^t, \boldsymbol{P}^t)|}{f(\boldsymbol{\Theta}^t, \boldsymbol{s}^t, \boldsymbol{P}^t)} < \delta,$$

where δ is a small tolerance parameter and f defined as in (2). Define the final outputs as $(\hat{\boldsymbol{\Theta}}, \hat{\boldsymbol{s}}, \hat{\boldsymbol{P}})$. Re-normalize $\hat{\boldsymbol{P}}$ by dividing each column of by its sum.

Section 3, $\boldsymbol{\Theta}^t$, $\boldsymbol{P}^t$, and $\boldsymbol{s}^t$ remain non-negative throughout all iterations if their initial values are non-negative. It is recommended that the initialized $\boldsymbol{\Theta}^0$ satisfies $R_1 = 0$, so any zero in the known partial signature matrix will remain zero. Second, it can be shown that the MU algorithm can achieve numerical convergence under realistic assumptions. The details will also be discussed in Section 3. Another remark is that if we enforce $\text{diag}(\boldsymbol{s}) = \boldsymbol{I}_K$ and set $\alpha_1 = \alpha_2 = \beta = 0$, our MU algorithm is reduced to that of Lee and Seung (2000) for the canonical NMF. Due to the non-linearity of the objective function (2), it is not guaranteed that the MU algorithm can produce the global minimizer. Following the precedence of other NMF-based deconvolution software, we recommend running the MU algorithms with multiple restarts and choosing the one with the smallest Frobenius norm of the residuals on the estimated bulk expression (Gaujoux and Seoighe, 2010, 2013). We use 10 restarts as the default in our implementation.

2.4. Tuning Parameter Selection

A grid search using a B-fold cross-validation (CV) is designed for selecting tuning parameters. Since R_2 is designed to make the objective function in (2) strictly convex, α_2 is advised to be fixed at a minuscule value. In the implementation of ARTdeConv, $\alpha_2 = 10^{-12}$ is the default value.

To begin, $\boldsymbol{Y}$ is randomly divided into B different folds, each containing $1/B$ columns of $\boldsymbol{Y}$. Two grids $\mathfrak{A}_1, \mathfrak{B} \subset \mathbb{R}_{++}$ containing respective candidates values for α_1 and β are declared. Next, for one $\alpha_1 \in \mathfrak{A}_1$ and $\beta \in \mathfrak{B}$ and a fold $b \in \{1, 2, \ldots, B\}$, the columns in the fold are held out as the test set $\boldsymbol{Y}_{\text{test}}^{(b)}$. The rest of the columns in $\boldsymbol{Y}$ are used as the training set $\boldsymbol{Y}_{\text{train}}^{(b)}$, upon which a deconvolution solution $(\tilde{\boldsymbol{\Theta}}^{(b)}, \tilde{\boldsymbol{s}}^{(b)}, \tilde{\boldsymbol{P}}^{(b)})$ is obtained.

Then, since it is assumed that n samples should share the same $\boldsymbol{\Theta}$ and $\boldsymbol{s}$ in (1), the estimated proportions $\hat{\boldsymbol{P}}^{(b)}$ on the test set $\boldsymbol{Y}_{\text{test}}^{(b)}$ is obtained by

$$\hat{\boldsymbol{P}}^{(b)} = \underset{\boldsymbol{P}\in\mathbb{R}_+^{K\times n/B}}{\text{argmin}} \|\boldsymbol{Y}_{\text{test}}^{(b)} - \tilde{\boldsymbol{\Theta}}^{(b)}\text{diag}(\tilde{\boldsymbol{s}})^{(b)}\boldsymbol{P}\|_F^2.$$

Computationally, this is accomplished by solving a non-negative least square problem

using the `R` package `nnls`. The CV error given α_1 and β is

$$\mathrm{Err}(\alpha_1, \beta) = \frac{1}{B}\sum_{b=1}^{B} \|\boldsymbol{Y}_{\text{test}}^{(b)} - \tilde{\boldsymbol{\Theta}}^{(b)}\mathrm{diag}(\tilde{\boldsymbol{s}})^{(b)}\hat{\boldsymbol{P}}^{(b)}\|_F^2.$$

Finally, we select the best tuning parameters α_1^* and β^* that minimize $\mathrm{Err}(\alpha_1, \beta)$.

2.5. Model Identifiability

In deconvolution, estimating $\boldsymbol{P}$ is the main objective. We investigate whether a solution for $\boldsymbol{P}$ is unique under the model (1). We first denote $\boldsymbol{G} = \boldsymbol{\Theta}\mathrm{diag}(\boldsymbol{s})$, as this bundling of factors helps us simplify the problem. Although it is usually difficult to guarantee the identifiablility of NMF models in general settings (Gillis, 2020), it is achievable in deconvolution with two additional conditions: there exists at least one cell-type specific gene (also called a marker gene) for every cell type involved and the model (1) is noiseless. We derive the following results on these conditions.

Proposition 1. *Assume $\boldsymbol{Y} = \boldsymbol{G}\boldsymbol{P}$ and $\boldsymbol{P}$ is of full rank K. If there exists a known $K \times K$ sub-matrix $\boldsymbol{G}_0$ of $\boldsymbol{G}$, such that $\boldsymbol{G}_0 = diag(\boldsymbol{g}_0)$, and $\boldsymbol{P}$ satisfies $\boldsymbol{P}^\top \boldsymbol{1}_K = \boldsymbol{1}_n$, then $\boldsymbol{Y} = \boldsymbol{G}\boldsymbol{P}$ is the unique non-negative decomposition of $\boldsymbol{Y}$.*

The proof of Proposition 1 can be found in Web Appendix A. Similar theorems have been proposed in other studies (Venet et al., 2001; Tang et al., 2020). Although Proposition 1 represents an ideal form of the deconvolution problem, some conditions of it are reasonable in practice. For example, $\boldsymbol{P}$ is often of full rank and $\boldsymbol{P}^\top \boldsymbol{1}_K = \boldsymbol{1}_n$ indicates that $\boldsymbol{P}$ is a matrix of proportions. Moreover, the $\boldsymbol{G}_0$ in question can exist when there is at least one marker gene for all cell types involved, including the cell types with missing reference. This is a reasonable assumption to make when the cell types with missing reference are distinct, as these genes can often be inferred directly from the bulk expression without the reference expression (Wang et al., 2016; Qiu et al., 2021). If these conditions are violated, such as when marker genes are failed to be found for a cell type, or most often when the cell type with missing reference contains multiple unidentified cell types, we have the following weaker proposition:

Proposition 2. *Assume $\boldsymbol{Y} = \boldsymbol{G}\boldsymbol{P}$ and $\boldsymbol{P}$ is of full rank K. If there exists a known $K \times K$ sub-matrix $\boldsymbol{G}_0$ of $\boldsymbol{G}$, such that*

$$\boldsymbol{G}_0 = \begin{bmatrix} g_{0_1} & \dots & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \dots & g_{0_{K-1}} & 0 \\ c_1 & \dots & c_{K-1} & g_{0_K} \end{bmatrix}, \tag{7}$$

where $c_l \geq 0$, $l = 1, \dots, K-1$, and $\boldsymbol{P}$ satisfies $\boldsymbol{P}^\top \boldsymbol{1}_K = \boldsymbol{1}_n$, then there exists another decomposition of the form $\boldsymbol{Y} = \boldsymbol{G}^\circ \boldsymbol{P}^\circ$, where

$$\boldsymbol{p}_l^\circ = \begin{cases} \frac{g_{0_K}+c_l}{g_{0_K}+c_l^\circ}\boldsymbol{p}_l, & l = 1, \dots, K-1, \\ \sum_{j=1}^{K-1} \frac{c_j^\circ - c_j}{g_{0_K}+c_j^\circ}\boldsymbol{p}_j + \boldsymbol{p}_K, & l = K, \end{cases} \tag{8}$$

where $c_l^\circ \geq \max_{i\in\{1,\dots,n\}} p_{li}(g_{0_K} + c_l) - g_{0_K}$ and $\sum_{l=1}^{K-1} \frac{g_{0_K}+c_l}{g_{0_K}+c_l^\circ} p_{li} \leq 1$ for any $i \in \{1, \dots, n\}$.

The proof of Proposition 2 can also be found in Web Appendix A. Proposition 1 is a special case of Proposition 2 with the constraint $c_1 = c_2 = \cdots = c_{K-1} = 0$. Proposition 2 relaxes the conditions in Proposition 1 in that marker genes are allowed to be absent in one cell type. This could be a cell type with no reference expression whose marker genes are difficult to discover, or when a few unidentified cell types are grouped into one "cell type", for instance in Hoek et al. (2015). Proposition 2 shows that, under this circumstance, the model is no longer strictly identifiable, and gives the relationship between two possible decompositions.

We should also mention that the two propositions above are derived under the ideal noiseless assumption for the model (1). In practice, this is likely violated. A comprehensive consideration of the model identifiability problem with noise is above the scope of this paper. Relevant references can be found in Gillis (2020).

3. Convergence Analysis

Past literature has suggested that the convergence analysis of algorithms for solving NMF problems has been challenging (Lin, 2007; Tong et al., 2016). In this section, we prove the numerical convergence of Algorithm 1 by showing that it is a special case of the block successive upper-bound minimization (BSUM) algorithm given in Razaviyayn et al. (2013). Thus, the convergence is guaranteed by related BSUM theories. Before presenting main result, we state a few technical assumptions.

Assumption 1 (Initial conditions)**.** *(i)* $\forall k$, $\exists j_k \in \{1, \ldots, m\}$ *such that* $\theta^0_{j_k k} > 0$*; (ii)* $\forall k$, $\exists i_k \in \{1, \ldots, n\}$ *such that* $p^0_{k i_k} > 0$*. (iii)* $\forall i' \in \{i_1, i_2, \ldots, i_K\}$, $\forall j$, $Y_{ji'} > 0$*;* $\forall j' \in \{j_1, j_2, \ldots, j_K\}$, $\forall i$, $Y_{j'i} > 0$.

Assumption 2 (Boundedness)**.** *(i) For any* $1 \leq j \leq m$ *and* $1 \leq i \leq n$, $Y_{ji} \leq M_Y < \infty$*. (ii) For any* $1 \leq j \leq m$ *and* $1 \leq k \leq K_0$, $\Theta_{0_{jk}} \leq M_\Theta < \infty$.

These assumptions are reasonable in practice. For every cell type with established reference expression, it is safe to assume that there exist some genes that display non-zero cell-type-specific expression, thereby fulfilling Assumption 1(*i*). For cell types lacking reference expression, an initial value in $\boldsymbol{\Theta}^0$ can be straightforwardly set to a positive value, thereby also satisfying the same assumption. Assumption 1(*ii*) is automatically satisfied due to how $\boldsymbol{P}^0$ is initialized in Algorithm 1. Assumption 1(*iii*) plays a pivotal role in ensuring that positive initialized values remain positive throughout the iterative process (see Web Appendix A for details). This assumption is easy to verify and holds true unless the bulk expression matrix is excessively sparse, which is uncommon in practical applications. Assumption 2 is a mild assumption on upper bounds of gene expression. Consequently, the following theorem holds for Algorithm 1 for solving the problem in (3).

Theorem 1 (Convergence of Algorithm 1 to a Stationary Point)**.** *Under Assumptions 1 and 2, the sequence* $\{(\boldsymbol{\Theta}^t, \boldsymbol{s}^t, \boldsymbol{P}^t)\}_{t=1}^{\infty}$ *in Algorithm 1 converges to a stationary point of* $f(\boldsymbol{\Theta}, \boldsymbol{s}, \boldsymbol{P})$.

The proof of Theorem 1 relies on showing that the sub-level set $\mathcal{X}^0 = \{(\boldsymbol{\Theta}, \boldsymbol{s}, \boldsymbol{P}) : f(\boldsymbol{\Theta}, \boldsymbol{s}, \boldsymbol{P}) \leq f(\boldsymbol{\Theta}^0, \boldsymbol{s}^0, \boldsymbol{P}^0)\}$ is compact and the objective function in (3) is coercive, as well as demonstrating the majorization-minimization (MM) properties and strong convexity of the auxiliary functions used to derive the update steps in Algorithm 1. Details on the definitions of coercive functions and MM properties, and the proof of Theorem 1 are relegated to Web Appendix A.

4. Numerical Simulation

To assess the deconvolution performance of ARTdeConv in comparison to alternative methods, we conduct a benchmarking simulation study by evaluating it against three distinct reference-free methods: NMF (Gaujoux and Seoighe, 2010), debCAM (Chen et al., 2020), and LINSEED (Zaitsev et al., 2019).

We focus on reference-free methods because reference-based methods are inapplicable under the assumption of partial knowledge on the signature matrix. In addition, we also include results of randomly guessing the proportions as a bottom line for deconvolution performance: each proportion is independently guessed from a $U(0,1)$ distribution, and the guessed proportion matrix is then re-normalized to make each column have the unit sum.

We generate a set of pseudo-bulk expression matrices using methodologies similar to those outlined in previous studies (Avila Cobos et al., 2020) by the following formula

$$\boldsymbol{Y}^* = \boldsymbol{\Theta}^* \mathrm{diag}(\boldsymbol{s}^*)\boldsymbol{P}^* + \boldsymbol{\epsilon}. \tag{9}$$

In (9), $K = 5$ hypothetical cell types, CT1 - CT5, are constructed with known partial signatures on $K_0 = 4$ of them (CT1 - CT4). $n = 100$ samples are simulated. For $k = 1, 2, \ldots, 5$, s_k^* is generated independently from a χ^2 distribution with 5 degrees of freedom. To investigate the effects of the true relative abundance of CT5 in bulk tissue samples on deconvolution results, three classes of $\boldsymbol{P}^*$ are generated, representing when CT5 is rare, uniform, and extra compared to other cell types in the tissue. For the rare class, $\boldsymbol{p}_i^* \overset{i.i.d.}{\sim} \mathrm{Dir}(2^5, 2^4, 2^3, 2^2, 2)$, where Dir represents the Dirichlet distribution. For the uniform class, we first draw $\boldsymbol{\alpha}_i$ from a Dirichlet distribution with $\alpha_{ik} \overset{i.i.d.}{\sim} U(0.3, 0.7), k = 1, \ldots, 5$, and then generate $\boldsymbol{p}_i^*$ from $\mathrm{Dir}(\boldsymbol{\alpha}_i)$. For the extra class, $\boldsymbol{p}_i^* \overset{i.i.d.}{\sim} \mathrm{dir}(2, 2^2, 2^3, 2^4, 2^5)$, in reverse of the rare class. Reference medians $(m_1, \ldots, m_5)$ and ranges $(r_1, \ldots, r_5)$ are obtained directly from each row of $\boldsymbol{P}^*$.

The true signature matrix $\boldsymbol{\Theta}^*$ is generated gene-by-gene. We first create expression of a total of $M = 2000$ genes in a matrix $\boldsymbol{\Theta}^*_{\text{full}}$. For each gene $j = 1, 2, \ldots, M$, a 5×1 vector of expression is generated by

$$\boldsymbol{\theta}_j^{*\top} \sim \Pi\left[\chi^2_{200} \cdot \{(1-\gamma)\mathrm{dir}(1, 1, \ldots, 1) + \gamma \mathrm{dir}(1, \ldots, 1, 5)\}\right],$$

where $\Pi(\cdot)$ is the sampling without replacement operator. $\Pi(\cdot)$ allows all of CT1-CT5 have a number of gene signatures. The parameter γ dictates how cell-type-specific each gene is on average. A gene has more chance to be cell-type-specific when $\gamma = 1$ compared to when $\gamma = 0$. Then, for ARTdeConv and NMF, a subset of $m = 500$ highly cell-type-specific genes called marker genes are selected by the following method: first, for each gene, pick the cell type for which it has the highest expression as the target cell type. This gene then becomes a candidate for a marker of this target cell type. Second, for each candidate marker, calculate the ratio of between expression in its target cell type and those in the cell type with the second highest expression. Third, for each cell type, rank its candidate marker genes by the ratio calculated in the second step from highest to lowest. Last, select the top 100 candidate marker genes for each cell type. Expression of those selected genes are stored in $\boldsymbol{\Theta}^*$. The matrix $\boldsymbol{\Theta}^*_{K_0}$ consisting of the first four columns of $\boldsymbol{\Theta}^*$ is used as the partial signature matrix for ARTdeConv. On the other hand, debCAM and LINSEED search for marker gene subsets differently by looking for simplicial vertices of

the vector space spanned by normalized bulk gene expression instead (Zaitsev et al., 2019; Chen et al., 2020).

The matrix of random errors $\boldsymbol{\epsilon}$ is first generated on all of the $M = 2000$ genes. Beginning with $(\boldsymbol{\Theta}^*_{\text{full}}, \boldsymbol{s}^*, \boldsymbol{P}^*)$, we calculate the error-free full bulk expression matrix as $\boldsymbol{\Theta}^*_{\text{full}}\text{diag}(\boldsymbol{s}^*)\boldsymbol{P}^*$. Experiments on bulk RNA-seq data have discovered that bulk gene expression exhibit mean-variance dependency: the higher the mean of a gene's expression, the higher their variation (Anders and Huber, 2010), and $\boldsymbol{\epsilon}$ is generated according to this principle. For each gene j, its error-free mean of bulk expression $\bar{y}_j$ is calculated. Then its associated error for each sample i is generated by $e_{ji} \overset{i.i.d.}{\sim} N(0, \sigma^2\bar{y}_j^2),\ i = 1, 2, \ldots, n$, where σ is a parameter to control the noise level relative to the mean bulk expression. Two levels of noises with $\sigma = 0.1$ or 10 (low vs. high) are investigated to evaluate the robustness of the model to added noises. The final bulk matrix of the full set of genes is calculated using $\boldsymbol{Y}^*_{\text{full}} = \boldsymbol{\Theta}^*_{\text{full}}\text{diag}(\boldsymbol{s}^*)\boldsymbol{P}^* + \boldsymbol{\epsilon}$. From this, a subset of bulk expression matrix $\boldsymbol{Y}^*$ corresponding to the genes in $\boldsymbol{\Theta}^*$ is selected. Having generated all pieces of the equation for simulations, we conduct 100 simulations for each combination of γ, σ, and CT-5 abundance class, where we report the deconvolution performance of benchmarked methods. CV is performed in each simulation to find the best values for α_1 and β according to Section 2.4. A tuning grid $\mathfrak{A}_1 = \mathfrak{B} = (10^{-2}, 10^{-1}, \cdots, 10^3, 10^4)$ was set up with the tolerance parameter chosen as $\delta = 10^{-5}$.

We now describe the default initialization provided in the implementation of ARTdeConv. First, ARTdeConv uses the first four columns of $\boldsymbol{\Theta}^0$ as $\boldsymbol{\Theta}^*_{K_0}$ to satisfy $R_1 = 0$. The initial value for each gene in the last column θ^0_{j5} is generated independently by $\theta^0_{j5} \sim |N(\bar{y}_j, \hat{\sigma}_j^2)|$, where $\bar{y}_j$ and $\hat{\sigma}_j^2$ are the mean and variance of the bulk expression values for gene j across all n samples. Then, for each cell type k of sample i, p^0_{ki} is generated by $N(m_k, 0.1)$. After this, negative entries in $\boldsymbol{P}^0$ are corrected to 0.01. Lastly, $\boldsymbol{s}^0$ is initialized by solving

$$\boldsymbol{s}^0 = \underset{\boldsymbol{s}\in\mathbb{R}^K}{\text{argmin}}\frac{1}{2}\|\boldsymbol{Y}^* - \boldsymbol{\Theta}^0\text{diag}(\boldsymbol{s})\boldsymbol{P}^0\|_F^2.$$

Negative values in $\boldsymbol{s}^0$ will be replaced by the mean of positive values.

Due to a lack of identifiability, estimated proportions from NMF, debCAM, and LINSEED need to be manually matched to different cell types. To do so, we utilize $\hat{\boldsymbol{\Theta}}$ and the partial signature matrix $\boldsymbol{\Theta}^*_{K_0}$. Starting from the first column of $\boldsymbol{\Theta}^*_{K_0}$, $\boldsymbol{\theta}^*_{K_{01}}$, we find the column of $\hat{\boldsymbol{\Theta}}$, say column k, that has the highest Pearson's correlation with $\boldsymbol{\theta}^*_{K_{01}}$. Then, the k-th row of $\hat{\boldsymbol{P}}$ is determined to be estimated proportions for CT1. We do this for the rest of the columns of $\boldsymbol{\Theta}^*_{K_0}$ and those of $\hat{\boldsymbol{\Theta}}$, until there is one last unmatched column in $\hat{\boldsymbol{\Theta}}$. That unmatched column, and its corresponding row in $\hat{\boldsymbol{P}}$, is then matched to CT5.

Performance metrics for evaluating deconvolution results include the followings: (a) $\frac{1}{n}cor(\boldsymbol{p}^*_i, \hat{\boldsymbol{p}}_i)$, (b) $\frac{1}{5n}\sum_{i=1}^n\sum_{k=1}^5|p^*_{ki} - \hat{p}_{ki}|$, (c) $cor(\boldsymbol{p}^{*T}_5, \hat{\boldsymbol{p}}^\top_5)$, (d) $\frac{1}{n}\sum_{i=1}^n|p^*_{5i} - \hat{p}_{5i}|$, where cor denotes the Pearson's correlation between two vectors. (a) and (b) delineate the overall deconvolution accuracy, while (c) and (d) show the deconvolution accuracy for CT5, the missing cell type. These results can be found in Figure 1 - 4.

Figures 1 and 2 show that, overall, ARTdeConv performs better than other benchmarked reference-free methods in recovering the correct proportions regardless of the average level of cell-type specificity of signature genes, the level of additive noises, and the relative abundance of the missing cell type to other cell types. It can also be noted that, as expected, debCAM performs better when the signature genes are highly cell-type specific ($\gamma = 1$), while NMF works better when the true cell-type proportions are rather

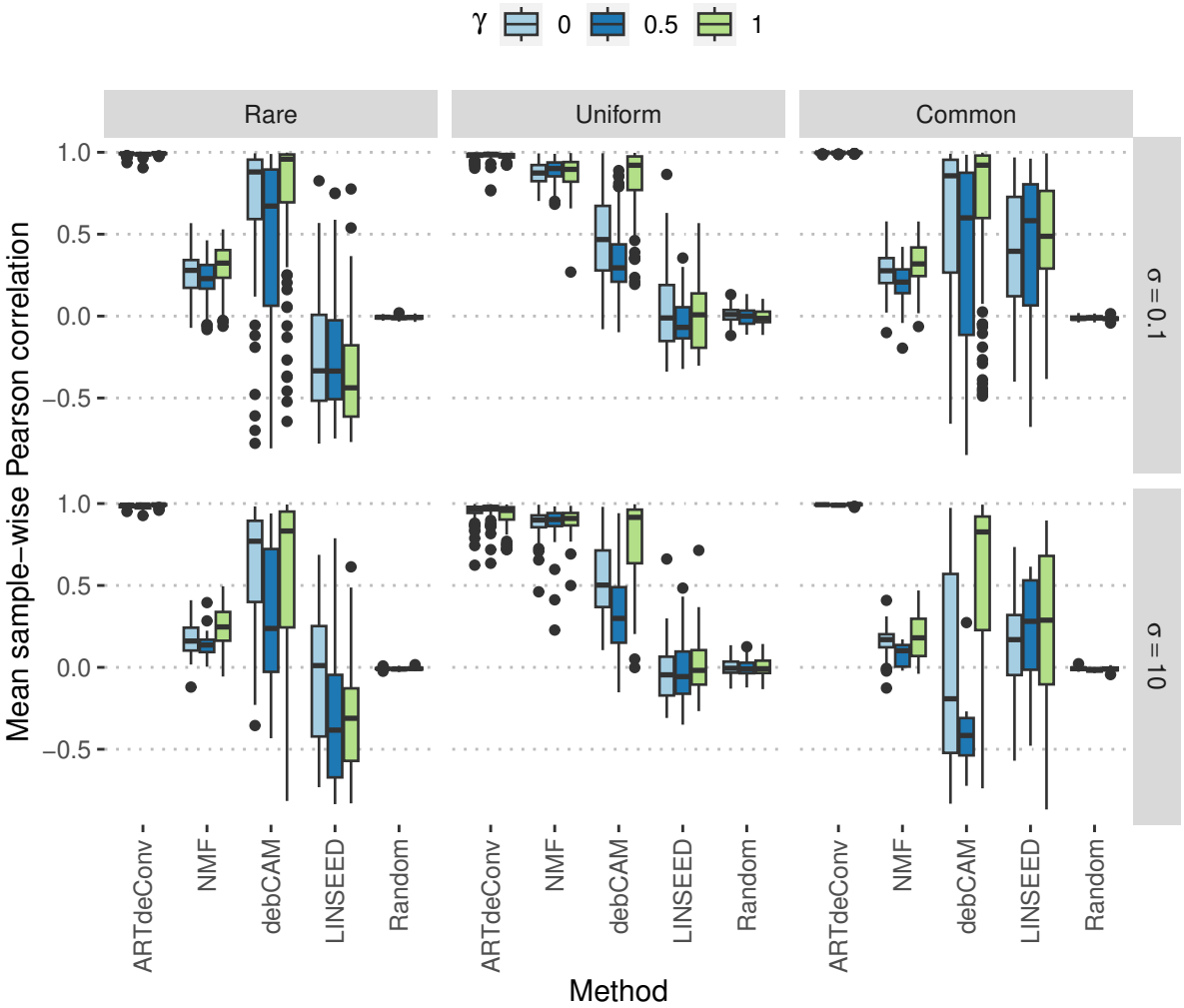

Figure 1: Mean sample-wise Pearson correlation between true proportions and estimated cell-type proportions through five benchmarked methods.

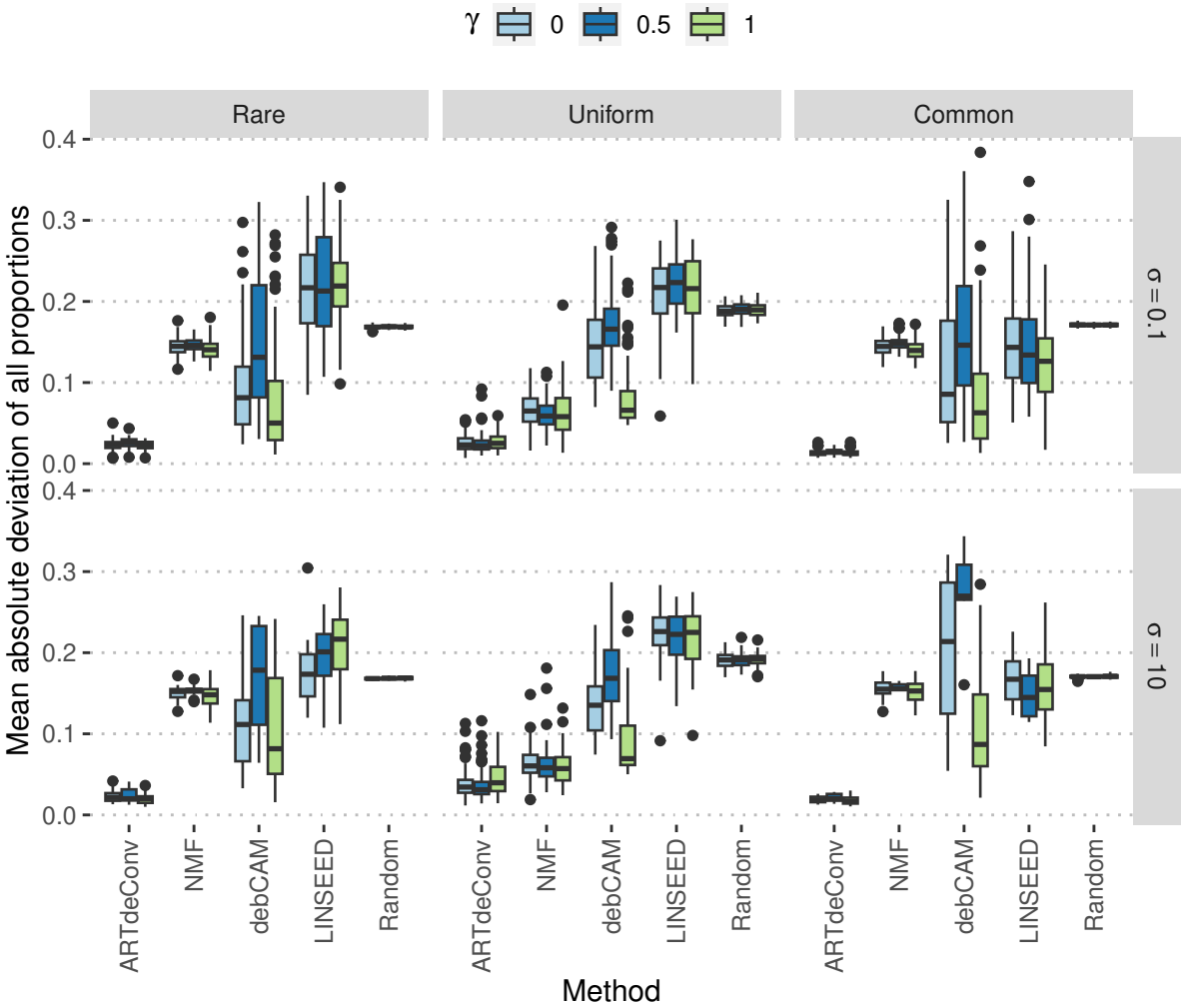

Figure 2: Mean absolute deviation (MAD) between true proportions and estimated cell-type proportions through five benchmarked methods.

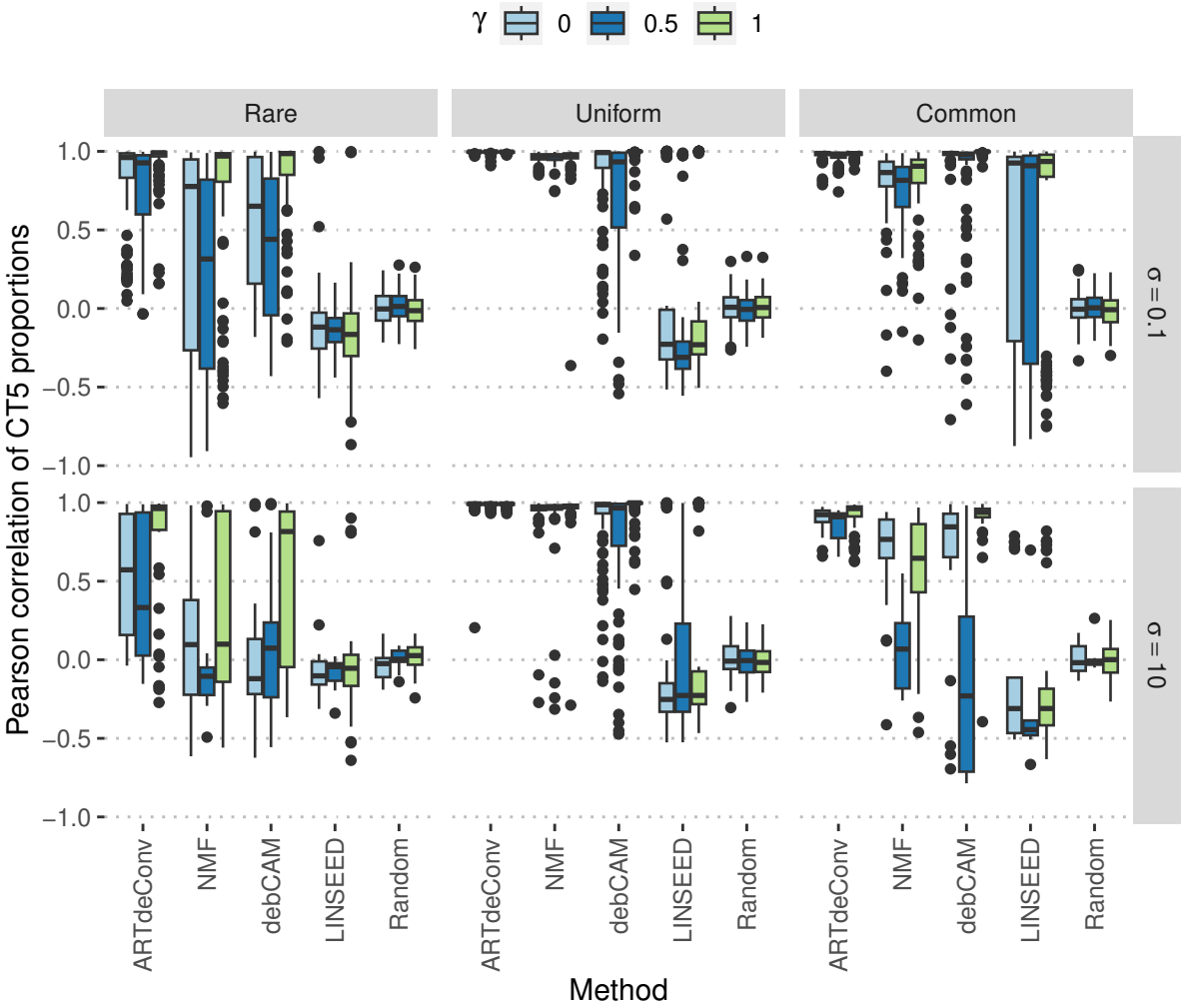

Figure 3: Pearson correlation between true proportions and estimated proportions of CT5 through five benchmarked methods.

uniform. LINSEED, which is known to be insensitive to cell-type mRNA amounts and whose results are at times worse than random guesses, does not perform well at all except in a few cases.

Figures 3 and 4 indicate that ARTdeConv produces better estimation of CT5's proportions on average compared to the other methods, even when this cell type is relatively rare in the bulk samples compared to others. ARTdeConv also achieves much lower mean absolute deviation in estimated proportions compared to other methods under almost all scenarios. Other noted observations are similar to those from Figure 1 and 2.

We further remark that, in the simulations described above, $\mathbf{\Theta}^*$ includes knowledge of marker genes for CT5. In practice, obtaining the marker genes of cell types without reference expression is not as straightforward but achievable through methods such as in Wang et al. (2016) and Qiu et al. (2021). Furthermore, deconvolution is completely feasible even without well defined marker genes for those cell types. See for instance the analysis in Section 5.

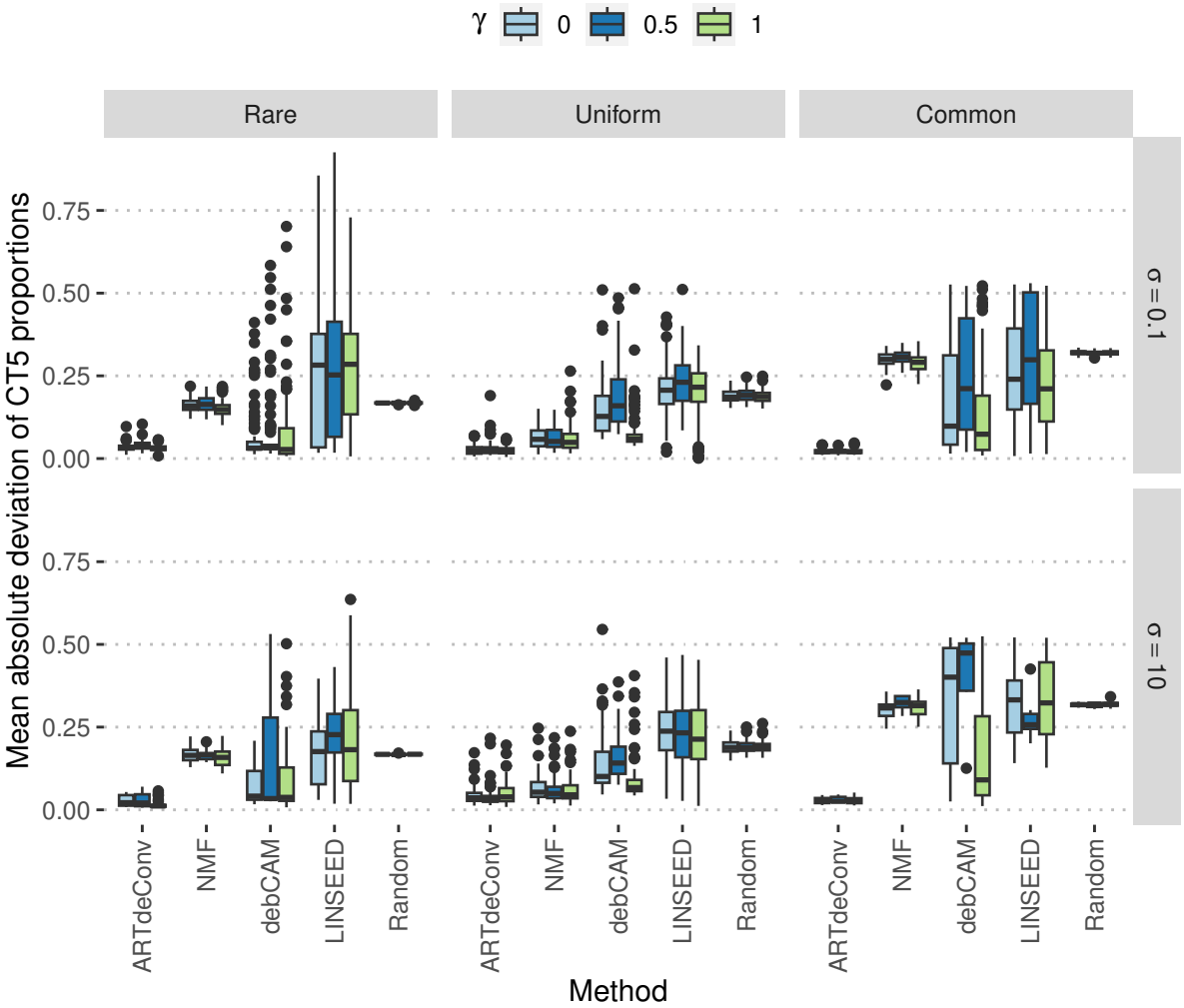

Figure 4: Mean absolute deviation (MAD) between true proportions and estimated proportions of CT5 through five benchmarked methods.

5. Real Data Example

To exemplify the application of ARTdeConv to real-world data, we utilize a dataset sourced from (Hoek et al., 2015). This dataset examines alterations in gene expression within peripheral blood mononuclear cells (PBMCs) and sorted cell lines by magnetic-sorting (MACS). Blood samples were obtained from healthy volunteer subjects who were administered a trivalent inactivated influenza vaccine (TIV). Both bulk PBMCs and sorted PBMC cell lines from two enrolled subjects (labeled "HD30" and "HD31") vaccinated with a single dose of 2011–2012 seasonal TIV were collected at four different time points: before vaccination (Day 0), and at Day 1, 3, and 7 post-vaccination. These samples were obtained and processed fresh for RNA-seq analysis.

The PBMCs in this study encompass four distinct cell types: T cells, B cells, natural killer (NK) cells, and monocytes. Additionally, there were other unspecified cell types within the PBMCs, collectively termed as "others". We performed deconvolution of gene expression from the eight bulk PBMC samples. Expression from the sorted PBMC cell lines of the two subjects, specifically the sorted T cells, B cells, NK cells, and monocytes on Day 0, were used for constructing the partial signature matrix.

We first processed raw gene expression into the desired format. FASTQ files containing 50bp (base pairs) pair-end raw nucleotide sequence reads of transcripts from Illumina Hiseq 2000 sequencers were downloaded from the Sequence Read Archive (SRA) BioProject PRJNA271578. The raw reads were pre-processed and filtered where reads with length $<$ 50bp, with $>$ 30% bases with quality scores $<$ 30, having an average quality score $<$ 25 in any 10bp interval, or corresponding to special adapter sequences were removed using the software `fastp` (Chen et al., 2018b). Then, reads passing the filters were mapped to reference human genome hg38 GRCh38 Release 43 (downloaded from the GENCODE website) and quantified using `salmon` (Patro et al., 2017). The output files were then processed using the `R` package `tximport` (Soneson et al., 2015): only protein-coding genes with an official gene symbol were selected and their transcripts per million (TPM) values were gathered into a gene-by-sample matrix of gene expression. Additionally, genes with a total TPM of less than 10 across the four cell types were excluded from consideration.

Distinct marker genes for the four predefined PBMC cell types were individually selected on gene expression in TPM of sorted cell lines from each subject using sorted bulk profiles of individual cell types. The selection process identifies the top 10 genes exhibiting the highest cell-type specificity for each cell type from each subject according the ratio of expression method outlined in Section 4. The union of the two sets of marker genes, totalling 73 genes, were used in the deconvolution.

To ensure reproducibility, a random seed of 1000 was set in the `R` programming environment. The tuning grid for α_1 was set as the interval from 2^{-4} to 2^0 with a step size of 0.2 in the power, and that for β was set as the interval from 2^0 to 2^4 also with a step size of 0.2 in the power. α_2 was fixed at 10^{-12}. The difference in magnitudes between α_1 and β in the tuning grids ensures proper regularization, for there is a scale difference between gene expression and the proportions. The eventual selected tuning parameter values were within the boundaries of the grids. The tolerance parameter was set as $\delta = 10^{-4}$.

The means and ranges of the cell-types in question are obtained through the reference values in Kleiveland (2015). For each of T cell, B cell, NK cell, and monocyte, the proportion means are calculated by taking the average of the upper and lower bounds of the reference values, while the ranges are calculated by taking the difference between the two bounds. For the "others" cell type, the mean was calculated by subtracting the means

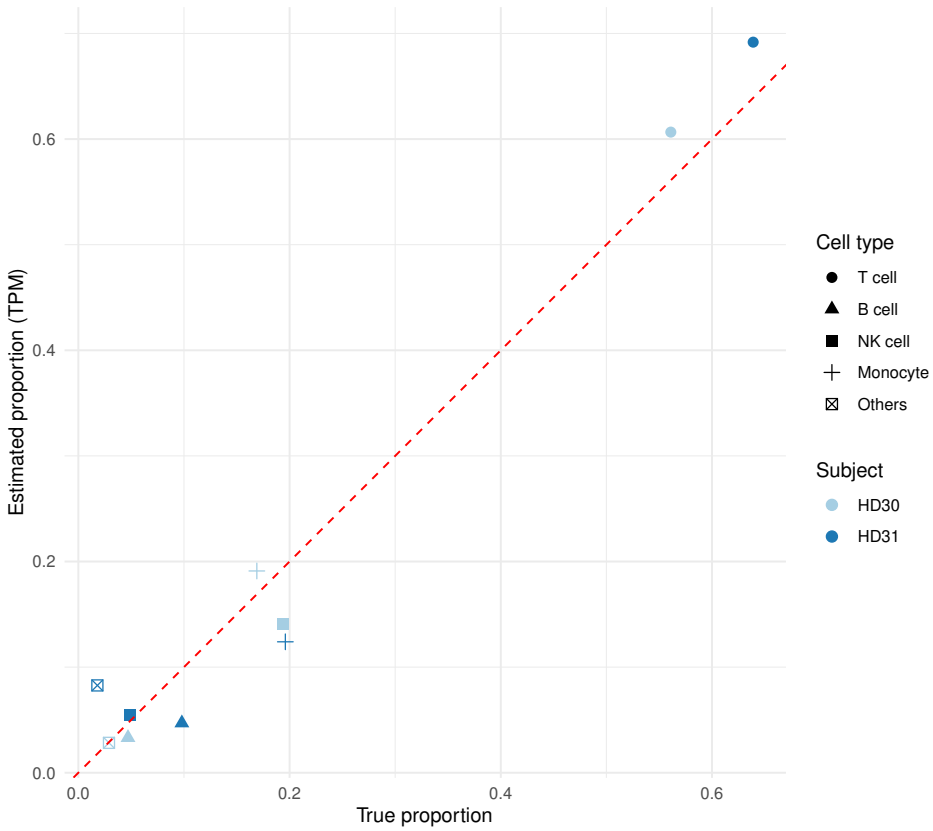

Figure 5: Estimated PBMC cell-type proportions from deconvolution using ARTdeConv versus flow cytometry measurements for the two subjects on Day 0.

of all other cell types from 1, and the range the difference between the upper and lower possible proportions of the other cell types. The exact values of the means and ranges are reported in Supplementary Table 1 in Web Appendix A. The vector of ranges was further normalized such that it has unit L_2 norm before deconvolution.

The PBMC cell-type percentages on Day 0 were measured by flow cytometry and reported by Racle et al. (2017), which were used as the ground truths for evaluating the performance of ARTdeConv. Indeed, as illustrated in Figure 5, ARTdeConv achieved a notable degree of precision in estimating proportions on Day 0 within the bulk expression data as compared to the flow cytometry measurements. Namely, for HD30 on Day 0, the estimated proportions display a Pearson correlation of 0.992 with the flow cytometry measurements, accompanied by a mean absolute deviation (MAD) of 0.027. Similarly, for HD31 on Day 0, a Pearson correlation of 0.977 is observed, along with a MAD of 0.049.

Figure 6 depicts the estimated proportions of PBMC cell types across all time points utilizing the Day 0 partial signature matrix. On the subject HD30, it shows a decline of T cell abundances before Day 1 and an increase between Days 1 and 3 followed by a slight decline until Day 7. It also shows an increase of monocyte abundances before and a decrease after Day 1. Both trends are consistent with the profile of a virus shedder infected with the H1N1 influenza virus in the study of Rahil et al. (2020) with a small time shift. The time shift could be attributed to the different viral strains and strengths between Rahil et al. (2020) and Hoek et al. (2015). The lack of notable changes in cell-type proportions of the subject HD31 is more similar to the profile of a non-virus shedder in Rahil et al. (2020). More information on the viral progression of these two subjects is

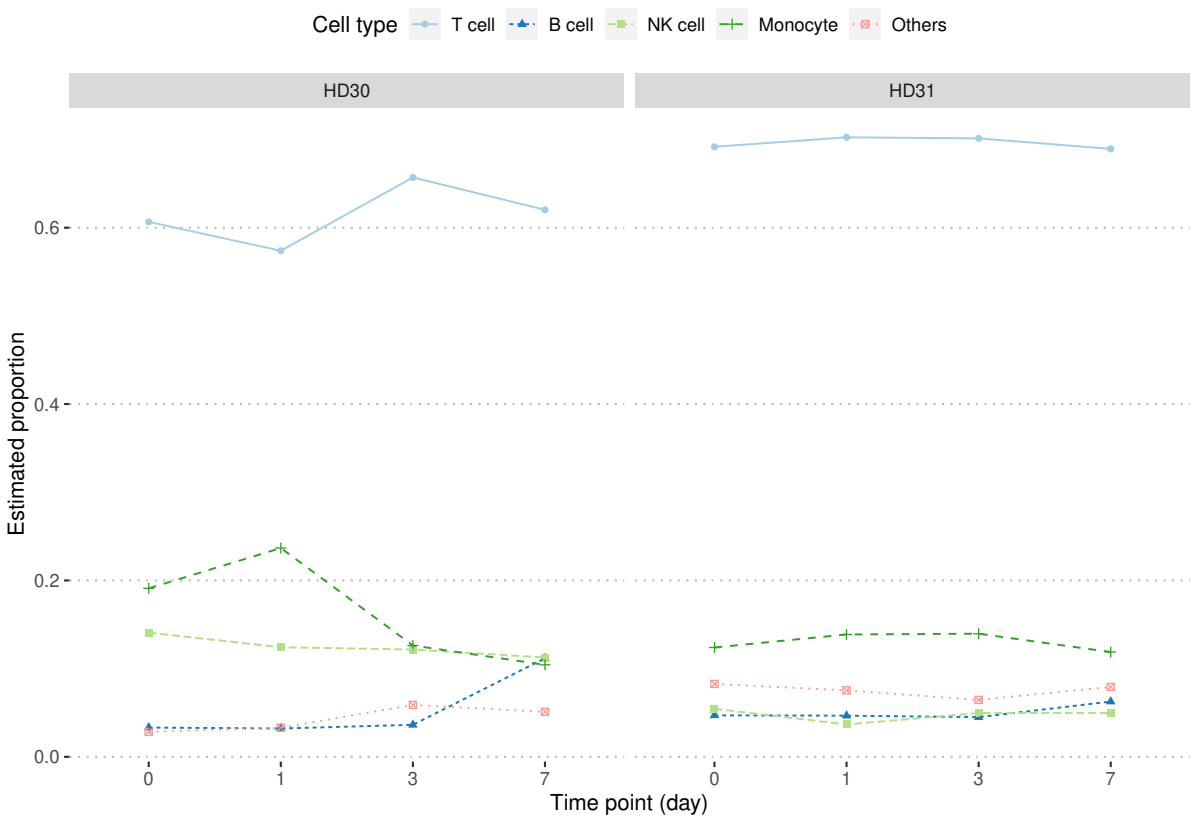

Figure 6: Estimated proportions of PBMC cell types for the two subjects across all time points from deconvolution using ARTdeConv.

needed to confirm these observations.

We wish to point out that normalizing gene expression by TPM is known for losing the information on cell-type mRNA amounts, which is corrected by ARTdeConv via $\text{diag}(\hat{\boldsymbol{s}})$. On the same PBMC bulk samples, when ARTdeConv is re-run with $\text{diag}(\boldsymbol{s})$ forced to be the identity matrix, we observe less accurate estimates of proportions (see Supplementary Figure 1 in Web Appendix A), corroborating the need for the diagonal cell-type mRNA amount matrix in the model (1).

6. Discussion

In this paper, we introduce ARTdeConv, an innovative deconvolution approach. An important feature of ARTdeConv is its adoption of a tri-factor model, which integrates an additional diagonal matrix $\text{diag}(\boldsymbol{s})$ to consider cell-type mRNA amounts during the deconvolution process. ARTdeConv offers enhanced flexibility compared to reference-based methods, as it accommodates cell types whose reference gene expression are not known. Additionally, ARTdeConv presents advantages over reference-free methods by incorporating cell type expression within the partial signature matrix. Moreover, the method makes effective use of reference information on proportion medians (or means) and ranges derived from external studies to provide informed guidance for estimating cell-type proportions within an appropriate range.

In simulation studies, ARTdeConv showcases superior deconvolution accuracy over reference-free methods in benchmarking both overall and on the cell type with missing reference expression. The analysis of real data further underscores the reliability of ARTdeConv when cell-type mRNA amount information is obscured.

Additionally, we derive the MU algorithm for ARTdeConv and present a theorem that establishes the convergence of this algorithm to stationary points. This proof is derived by casting ARTdeConv's algorithm as a special case of the BSUM algorithm introduced by Razaviyayn et al. (2013).

An argument can be made that obtaining precise reference information on proportions demands additional efforts in practice. However, the advantages of integrating this information are substantial. Without it, the task of associating estimated proportions with their respective cell types can prove challenging. In fact, as demonstrated in Section 4, we had to perform manual matching of results for all reference-free methods, which becomes unfeasible if multiple cell types lack reference expression.

While ARTdeConv provides a guarantee of convergence, it does not assure that the estimated proportions are globally optimal. This particular challenge is not unique to ARTdeConv, as it is encountered in other reference-free methods such as NMF (Gaujoux and Seoighe, 2010). To enhance the likelihood of achieving improved results, we recommend employing signature genes for the deconvolution process and considering multiple restarts of ARTdeConv. Other methods like debCAM (Chen et al., 2020) mitigates this issue by assuming that cell-type-specific genes can be found for each involved cell type and solving the deconvolution using a different mathematical framework. However, their performance can diminish under scenarios that deviate from their ideal assumption, which is a common occurrence. Even extensive efforts in optimizing signature matrices struggle to identify purely cell-type-specific marker genes for all cell types (Newman et al., 2015).

Several important points about the real data analysis also warrant attention. Firstly, the data obtained from (Hoek et al., 2015) offer an added advantage, as both the expression

of bulk samples and the partial signature matrix are sequenced using the same platform and are collected approximately concurrently. This minimizes the influence of most technical artifacts. In cases where bulk and cell-type reference expression are derived from different studies, additional caution should be exercised during data preprocessing prior to deconvolution. Secondly, when utilizing ARTdeConv, it is advisable to maintain consistency between the bulk and reference expression in terms of measurement, such as both being subjected to TPM normalization.

Lastly, in our real data analysis, we applied the same partial signature matrix across all bulk samples. While the subjects in this study are both healthy when recruited, it has been recently found that gene signatures of cell types can shift in response to various infectious diseases and their progression (Kelly et al., 2022). This concept might also hold true for immune reactions related to vaccines. Performing deconvolution on samples representing diverse disease states using methods accounting for these changes in gene signatures could potentially enhance the deconvolution accuracy, but no such methods are currently available. Therefore, another avenue for future exploration involves finding techniques that can adjust the signature matrix to accommodate information on disease status.

Software Availability

ARTdeConv is available as an R package for download from the GitHub repository: `https://github.com/gr8lawrence/ARTDeConv`.

Acknowledgements

The authors gratefully acknowledge the support of National Institute of Allergy and Infectious Diseases at the National Institutes of Health through the grant U19AI144181 and that of National Institute of Aging through the grant R01AG073259.

References

S. Anders and W. Huber. Differential expression analysis for sequence count data. *Nature Precedings*, pages 1–1, 2010.

F. Avila Cobos, J. Alquicira-Hernandez, J. E. Powell, P. Mestdagh, and K. De Preter. Benchmarking of cell type deconvolution pipelines for transcriptomics data. *Nature communications*, 11(1):1–14, 2020.

B. Chen, M. S. Khodadoust, C. L. Liu, A. M. Newman, and A. A. Alizadeh. Profiling tumor infiltrating immune cells with cibersort. *Cancer Systems Biology: Methods and Protocols*, pages 243–259, 2018a.

L. Chen, C.-T. Wu, N. Wang, D. M. Herrington, R. Clarke, and Y. Wang. debcam: a bioconductor r package for fully unsupervised deconvolution of complex tissues. *Bioinformatics*, 36(12):3927–3929, 2020.

S. Chen, Y. Zhou, Y. Chen, and J. Gu. fastp: an ultra-fast all-in-one fastq preprocessor. *Bioinformatics*, 34(17):i884–i890, 2018b.

F. Elloumi, Z. Hu, Y. Li, J. S. Parker, M. L. Gulley, K. D. Amos, and M. A. Troester. Systematic bias in genomic classification due to contaminating non-neoplastic tissue in breast tumor samples. *BMC medical genomics*, 4(1):1–12, 2011.

R. Gaujoux and C. Seoighe. A flexible r package for nonnegative matrix factorization. *BMC bioinformatics*, 11(1):1–9, 2010.

R. Gaujoux and C. Seoighe. Cellmix: a comprehensive toolbox for gene expression deconvolution. *Bioinformatics*, 29(17):2211–2212, 2013.

N. Gillis. *Nonnegative matrix factorization.* SIAM, 2020.

C. Hafemeister and R. Satija. Normalization and variance stabilization of single-cell rna-seq data using regularized negative binomial regression. *Genome biology*, 20(1): 1–15, 2019.

K. L. Hoek, P. Samir, L. M. Howard, X. Niu, N. Prasad, A. Galassie, Q. Liu, T. M. Allos, K. A. Floyd, Y. Guo, et al. A cell-based systems biology assessment of human blood to monitor immune responses after influenza vaccination. *PloS one*, 10(2):e0118528, 2015.

H. Jin and Z. Liu. A benchmark for rna-seq deconvolution analysis under dynamic testing environments. *Genome biology*, 22:1–23, 2021.

J. N. Kelly, L. Laloli, P. V'kovski, M. Holwerda, J. Portmann, V. E. Thiel, and R. Dijkman. Comprehensive single cell analysis of pandemic influenza a virus infection in the human airways uncovers cell-type specific host transcriptional signatures relevant for disease progression and pathogenesis. *Frontiers in immunology*, 13(978824), 2022.

J. Kim, Y. He, and H. Park. Algorithms for nonnegative matrix and tensor factorizations: A unified view based on block coordinate descent framework. *Journal of Global Optimization*, 58(2):285–319, 2014.

C. R. Kleiveland. *Peripheral Blood Mononuclear Cells.* Springer, 2015.

K. Lange. *MM optimization algorithms.* SIAM, 2016.

D. Lee and H. S. Seung. Algorithms for non-negative matrix factorization. *Advances in neural information processing systems*, 13, 2000.

Z. Li and H. Wu. Toast: improving reference-free cell composition estimation by cross-cell type differential analysis. *Genome biology*, 20(1):1–17, 2019.

C.-J. Lin. On the convergence of multiplicative update algorithms for nonnegative matrix factorization. *IEEE Transactions on Neural Networks*, 18(6):1589–1596, 2007.

N. Lytal, D. Ran, and L. An. Normalization methods on single-cell rna-seq data: an empirical survey. *Frontiers in genetics*, 11:41, 2020.

S. Mohammadi, N. Zuckerman, A. Goldsmith, and A. Grama. A critical survey of deconvolution methods for separating cell types in complex tissues. *Proc. IEEE*, 105(2): 340–366, Feb. 2017.

J. S. Morris and S. Kopetz. Tumor microenvironment in gene signatures: critical biology or confounding noise? *Clinical Cancer Research*, 22(16):3989–3991, 2016.

A. M. Newman, C. L. Liu, M. R. Green, A. J. Gentles, W. Feng, Y. Xu, C. D. Hoang, M. Diehn, and A. A. Alizadeh. Robust enumeration of cell subsets from tissue expression profiles. *Nature methods*, 12(5):453–457, 2015.

R. Patro, G. Duggal, M. I. Love, R. A. Irizarry, and C. Kingsford. Salmon provides fast and bias-aware quantification of transcript expression. *Nature methods*, 14(4):417–419, 2017.

Y. Qiu, J. Wang, J. Lei, and K. Roeder. Identification of cell-type-specific marker genes from co-expression patterns in tissue samples. *Bioinformatics*, 37(19):3228–3234, 2021.

J. Racle, K. de Jonge, P. Baumgaertner, D. E. Speiser, and D. Gfeller. Simultaneous enumeration of cancer and immune cell types from bulk tumor gene expression data. *elife*, 6:e26476, 2017.

Z. Rahil, R. Leylek, C. M. Schürch, H. Chen, Z. Bjornson-Hooper, S. R. Christensen, P. F. Gherardini, S. S. Bhate, M. H. Spitzer, G. K. Fragiadakis, et al. Landscape of coordinated immune responses to h1n1 challenge in humans. *The Journal of clinical investigation*, 130(11):5800–5816, 2020.

M. Razaviyayn, M. Hong, and Z.-Q. Luo. A unified convergence analysis of block successive minimization methods for nonsmooth optimization. *SIAM Journal on Optimization*, 23 (2):1126–1153, 2013.

C. Soneson, M. I. Love, and M. D. Robinson. Differential analyses for rna-seq: transcript-level estimates improve gene-level inferences. *F1000Research*, 4, 2015.

M. H. Spitzer and G. P. Nolan. Mass cytometry: single cells, many features. *Cell*, 165(4): 780–791, 2016.

A. Takami, S. Watanabe, Y. Yamamoto, H. Miyachi, Y. Bamba, M. Ohata, S. Mishima, H. Kubota, A. Nishiura, T. Inaba, M. Enomoto, T. Mitsuhashi, K. Nakanishi, R. Miura, E. Nonaka, K. Shimbo, Y. Yatomi, K. Tohyama, and Japanese Society for Laboratory Hematology subcommittee on Standardization of Blood Cell Morphology (JSLH-SBCM). Reference intervals of white blood cell parameters for healthy adults in japan. *Int. J. Lab. Hematol.*, 43(5):948–958, Oct. 2021.

D. Tang, S. Park, and H. Zhao. Nitumid: nonnegative matrix factorization-based immune-tumor microenvironment deconvolution. *Bioinformatics*, 36(5):1344–1350, 2020.

L. Tong, J. Zhou, Y. Qian, X. Bai, and Y. Gao. Nonnegative-matrix-factorization-based hyperspectral unmixing with partially known endmembers. *IEEE Transactions on Geoscience and Remote Sensing*, 54(11):6531–6544, 2016. doi: 10.1109/TGRS.2016.2586110.

D. Venet, F. Pecasse, C. Maenhaut, and H. Bersini. Separation of samples into their constituents using gene expression data. *Bioinformatics*, 17(suppl_1):S279–S287, 2001.

N. Wang, E. P. Hoffman, L. Chen, L. Chen, Z. Zhang, C. Liu, G. Yu, D. M. Herrington, R. Clarke, and Y. Wang. Mathematical modelling of transcriptional heterogeneity identifies novel markers and subpopulations in complex tissues. *Scientific reports*, 6(1): 18909, 2016.

X. Wang, J. Park, K. Susztak, N. R. Zhang, and M. Li. Bulk tissue cell type deconvolution with multi-subject single-cell expression reference. *Nature communications*, 10(1):1–9, 2019.

G. Wigerblad, Q. Cao, S. Brooks, F. Naz, M. Gadkari, K. Jiang, S. Gupta, L. O'Neil, S. Dell'Orso, M. J. Kaplan, et al. Single-cell analysis reveals the range of transcriptional states of circulating human neutrophils. *bioRxiv*, 2022.

K. Zaitsev, M. Bambouskova, A. Swain, and M. N. Artyomov. Complete deconvolution of cellular mixtures based on linearity of transcriptional signatures. *Nature communications*, 10(1):1–16, 2019.

Supplementary Material

A. The MU Steps

A.1. A Preliminary Lemma

Similar to Lee and Seung (2000), $f(\boldsymbol{\Theta}, \boldsymbol{s}, \boldsymbol{P})$ is minimized using block-wise auxiliary functions that are quadratic over-estimators, which then leads to the MU steps in (4), (5), and (6). The claims that these updates will keep the positive initialized values positive will be investigated at the end of this section. An auxiliary function is defined next.

Definition 1 (Auxiliary function)**.** *Given a vector space V and $\boldsymbol{v}, \boldsymbol{v}' \in V$, a function $g(\boldsymbol{v}|\boldsymbol{v}') : V \times V \mapsto \mathbb{R}$ is an auxiliary function for the function $f(\boldsymbol{v}) : V \mapsto \mathbb{R}$ at $\boldsymbol{v}'$ if the following conditions are satisfied:*

$$g(\boldsymbol{v}|\boldsymbol{v}') \geq f(\boldsymbol{v}'), \forall \boldsymbol{v} \in V, \text{ and } g(\boldsymbol{v}'|\boldsymbol{v}') = f(\boldsymbol{v}').$$

As we shall see, the vector blocks in the auxiliary functions would be each row of $\boldsymbol{\Theta}$, each column of $\boldsymbol{P}$, and $\boldsymbol{s}$. To find the specific auxiliary functions for $f(\boldsymbol{\Theta}, \boldsymbol{s}, \boldsymbol{P})$, a slightly modified form of a technical lemma given by Lin (2007) is needed and stated next. Its proof is also given in the same paper and thus omitted.

Lemma 1. *Given a positive semi-definite matrix $\boldsymbol{Q} \in \mathbb{R}_+^{b \times b}$ and $\boldsymbol{w} \in \mathbb{R}_+^b$, let $I \subseteq \{1, \ldots, b\}$ be a set of indices such that $w_k > 0$ for any $k \in I$, $\boldsymbol{Q}_{II} \in \mathbb{R}_{++}^{|I| \times |I|}$, and $\boldsymbol{w}_I \in \mathbb{R}_{++}^{|I|}$ be the sub-matrix of $\boldsymbol{Q}$ and the sub-vector of $\boldsymbol{w}$ with indices in I, where $|I|$ denotes the cardinality of I. Then,*

$$diag\left\{\frac{(\boldsymbol{Q}_{II}\boldsymbol{w}_I)}{\boldsymbol{w}_I}\right\} - \boldsymbol{Q}_{II} = \begin{bmatrix} \frac{(\boldsymbol{Q}_{II}\boldsymbol{w}_I)_1}{w_{I1}} & & \\ & \ddots & \\ & & \frac{(\boldsymbol{Q}_{II}\boldsymbol{w}_I)_{|I|}}{w_{I|I|}} \end{bmatrix} - \boldsymbol{Q}_{II}$$

is always positive semi-definite.

With Lemma 1, the auxiliary functions and multiplicative update (MU) steps are ready to be derived.

A.2. Derivations of The MU Steps

The auxiliary function and update step for each row $\boldsymbol{\theta}_j^\top$ is derived and then combined for $j = 1, \ldots, m$ into the form of (4). For simplicity in notations, the index j is omitted in subscripts. The iteration number t is also omitted for $\boldsymbol{s}$ and $\boldsymbol{P}$ since both are treated as fixed. That is, we denote $\boldsymbol{y}_j^\top$ as $\boldsymbol{y}^\top$, $\boldsymbol{\theta}_j^\top$ as $\boldsymbol{\theta}^\top$, and $\boldsymbol{\delta}_j^\top$ as $\boldsymbol{\delta}$. We also define $\boldsymbol{\delta}_j^c = \boldsymbol{J} - \boldsymbol{\delta}_j$ for all j. Thus, for the j-th row of $\boldsymbol{\Theta}$, we can define the sub-problem of $f(\boldsymbol{\Theta}, \boldsymbol{s}, \boldsymbol{P})$ as

$$\underset{\boldsymbol{\theta} \in \mathbb{R}_+^K}{\operatorname{argmin}} \frac{1}{2mn} \|\boldsymbol{y}^\top - \boldsymbol{\theta}^\top \operatorname{diag}(\boldsymbol{s}) \boldsymbol{P}\|_2^2 + \frac{\alpha_1}{2} \|\boldsymbol{\delta}^\top \odot (\boldsymbol{\theta}^\top - \boldsymbol{\theta}_0^\top)\|_2^2 + \frac{\alpha_2}{2} \|\boldsymbol{\delta}^{c\top} \odot \boldsymbol{\theta}^\top\|_2^2 + C_1(\boldsymbol{s}, \boldsymbol{P}) \tag{S.10}$$

$$= \underset{\boldsymbol{\theta} \in \mathbb{R}_+^K}{\operatorname{argmin}} \frac{1}{2mn} \|\boldsymbol{y}^\top - \boldsymbol{P}^\top \operatorname{diag}(\boldsymbol{s}) \boldsymbol{\theta}\|_2^2 + \frac{\alpha_1}{2} \|\boldsymbol{\delta} \odot (\boldsymbol{\theta} - \boldsymbol{\theta}_0)\|_2^2 + \frac{\alpha_2}{2} \|\boldsymbol{\delta}^c \odot \boldsymbol{\theta}\|_2^2 + C_1(\boldsymbol{s}, \boldsymbol{P}), \tag{S.11}$$

where $\text{diag}(\boldsymbol{s})$ and $\boldsymbol{P}$ are treated as fixed and $C_1(\boldsymbol{s}, \boldsymbol{P})$ is a constant with respect to $\boldsymbol{\theta}$. We call the function in (S.11) $f_\theta(\boldsymbol{\theta})$. (S.11) is derived from (S.10) by transposing. Now, to find the auxiliary function $h_\theta(\boldsymbol{\theta}|\boldsymbol{\theta}^t)$, by performing a second-order Taylor series expansion on $f_\theta(\boldsymbol{\theta})$ at $\boldsymbol{\theta} = \boldsymbol{\theta}^t$, we have

$$f_\theta(\boldsymbol{\theta}) = f_\theta(\boldsymbol{\theta}^t) + \nabla_{\boldsymbol{\theta}} f_\theta(\boldsymbol{\theta}^t)^\top(\boldsymbol{\theta} - \boldsymbol{\theta}^t) + \frac{1}{2}(\boldsymbol{\theta} - \boldsymbol{\theta}^t)^\top \nabla^2_{\boldsymbol{\theta}} f_\theta(\boldsymbol{\theta}^t)(\boldsymbol{\theta} - \boldsymbol{\theta}^t).$$

Next, letting $\boldsymbol{V}(\boldsymbol{\delta}) = \text{diag}(\boldsymbol{\delta})$ and $\boldsymbol{V}^c(\boldsymbol{\delta}^c) = \text{diag}(\boldsymbol{\delta}^c)$, it can be observed that $\boldsymbol{\delta} \odot \boldsymbol{\theta} = \boldsymbol{V}\boldsymbol{\theta}$ and $\boldsymbol{\delta}^c \odot \boldsymbol{\theta} = \boldsymbol{V}^c\boldsymbol{\theta}$. Also, $\boldsymbol{V} + \boldsymbol{V}^c = \boldsymbol{I}$. The first and second derivatives of $f_\theta(\boldsymbol{\theta})$ are thus:

$$\nabla_{\boldsymbol{\theta}} f_\theta(\boldsymbol{\theta}) = \frac{1}{mn}\text{diag}(\boldsymbol{s})\boldsymbol{P}(\boldsymbol{P}^\top \text{diag}(\boldsymbol{s})\boldsymbol{\theta} - \boldsymbol{y}^\top) + \alpha_1 \boldsymbol{V}(\boldsymbol{\theta} - \boldsymbol{\theta}_0) + \alpha_2 \boldsymbol{V}^c\boldsymbol{\theta},$$

and

$$\nabla^2_{\boldsymbol{\theta}} f_\theta(\boldsymbol{\theta}) = \frac{1}{mn}\text{diag}(\boldsymbol{s})\boldsymbol{P}\boldsymbol{P}^\top \text{diag}(\boldsymbol{s}) + \alpha_1 \boldsymbol{V} + \alpha_2 \boldsymbol{V}^c.$$

It can be guaranteed $\nabla^2_{\boldsymbol{\theta}} f_\theta(\boldsymbol{\theta}) \succ \boldsymbol{0}$ given $\alpha_1, \alpha_2 > 0$, as $\nabla^2_{\boldsymbol{\theta}} f_\theta(\boldsymbol{\theta}) \succeq \alpha_1 \boldsymbol{V} + \alpha_2 \boldsymbol{V}^c \succeq \min(\alpha_1, \alpha_2)\boldsymbol{I} \succ \boldsymbol{0}$. Then, define $\boldsymbol{d}(\boldsymbol{\theta}^t) = \text{diag}(\boldsymbol{\theta}^t)^{-1}\nabla^2_{\boldsymbol{\theta}} f_\theta(\boldsymbol{\theta}^t)\boldsymbol{\theta}^t$, and $\boldsymbol{D}(\boldsymbol{\theta}^t) = \text{diag}(\boldsymbol{d}(\boldsymbol{\theta}^t))$. Lemma 1 implies that, on coordinates that correspond to the positive elements of $\boldsymbol{\theta}$, $\boldsymbol{D}(\boldsymbol{\theta}^t) - \nabla^2_{\boldsymbol{\theta}} f_\theta(\boldsymbol{\theta}^t) \succeq 0$. Define $\mathbb{R}^{K_1}_{++}$, where $K_1 \leq K$, as the open half space that contains this positive subset of $\boldsymbol{\theta}$. For simplification, we use the same notations for the full vectors and matrices on those positive coordinates. As mentioned in Section 2, we shall see that once a coordinate of $\boldsymbol{\theta}$ reaches zero, it stays zero thereafter, which justifies the focus on these positive coordinates. Next, define

$$h_\theta(\boldsymbol{\theta}|\boldsymbol{\theta}^t) = f_\theta(\boldsymbol{\theta}^t) + \nabla_{\boldsymbol{\theta}} f_\theta(\boldsymbol{\theta}^t)^\top(\boldsymbol{\theta} - \boldsymbol{\theta}^t) + \frac{1}{2}(\boldsymbol{\theta} - \boldsymbol{\theta}^t)^\top \boldsymbol{D}(\boldsymbol{\theta}^t)(\boldsymbol{\theta} - \boldsymbol{\theta}^t), \tag{S.12}$$

which is an auxiliary function of (S.11) (thus of $f(\boldsymbol{\Theta}, \boldsymbol{s}, \boldsymbol{P})$ when we combine all subproblems together for $j = 1, \ldots m$). To demonstrate that it satisfies Definition 1, simply see that

$$h_\theta(\boldsymbol{\theta}|\boldsymbol{\theta}^t) - f_\theta(\boldsymbol{\theta}) = \frac{1}{2}(\boldsymbol{\theta} - \boldsymbol{\theta}^t)^\top(\boldsymbol{D}(\boldsymbol{\theta}^t) - \nabla^2_{\boldsymbol{\theta}} f_\theta(\boldsymbol{\theta}^t))(\boldsymbol{\theta} - \boldsymbol{\theta}^t) \geq 0, \ \forall \boldsymbol{\theta}, \boldsymbol{\theta}^t \in \mathbb{R}^{K_1}_{++}$$

and due to Lemma 1 and that $h_\theta(\boldsymbol{\theta}^t|\boldsymbol{\theta}^t) - f_\theta(\boldsymbol{\theta}^t) = 0, \ \forall \boldsymbol{\theta}^t \in \mathbb{R}^{K_1}_{++}$. To derive the MU step from this auxiliary function, we find $\boldsymbol{\theta}^{t+1}$ such that

$$\boldsymbol{\theta}^{t+1} = \underset{\boldsymbol{\theta} \in \mathbb{R}^{K_1}_{+}}{\text{argmin}} \ h_\theta(\boldsymbol{\theta}|\boldsymbol{\theta}^t). \tag{S.13}$$

To guarantee the existence of a solution for (S.13), the feasible set of $\boldsymbol{\theta}$ is a closed space that includes 0. Since $\boldsymbol{D}(\boldsymbol{\theta}^t) - \nabla^2_{\boldsymbol{\theta}} f_\theta(\boldsymbol{\theta}^t) \succeq 0$ and $\nabla^2_{\boldsymbol{\theta}} f_\theta(\boldsymbol{\theta}^t) \succ 0$ by the assumptions, $\boldsymbol{D}(\boldsymbol{\theta}^t) \succ 0$, which implies that (S.12) is strictly convex. Thus, (S.12) has a global minimizer in the closed half space $\mathbb{R}^{K_1}_{+}$. By the optimality condition, if a feasible $\tilde{\boldsymbol{\theta}} \in \mathbb{R}^{K_1}_{+}$ satisfies $\nabla_{\boldsymbol{\theta}} h_\theta(\tilde{\boldsymbol{\theta}}|\boldsymbol{\theta}^t) = 0$, then $\tilde{\boldsymbol{\theta}}$ minimizes (S.12) globally and $\boldsymbol{\theta}^{t+1} = \tilde{\boldsymbol{\theta}}$ becasue of (S.13). Thus, we solve

$$\nabla_{\boldsymbol{\theta}} h_\theta(\boldsymbol{\theta}^{t+1}|\boldsymbol{\theta}^t) = \nabla_{\boldsymbol{\theta}} f_\theta(\boldsymbol{\theta}^t) + \boldsymbol{D}(\boldsymbol{\theta}^t)(\boldsymbol{\theta}^{t+1} - \boldsymbol{\theta}^t) = 0. \tag{S.14}$$

Because $\boldsymbol{D}(\boldsymbol{\theta}^t)$ is diagonal, for each $k \in 1, \ldots, K_1$, (S.14) implies

$$\theta_k^{t+1} = \theta_k^t - \frac{\nabla_{\boldsymbol{\theta}} f_\theta(\boldsymbol{\theta}^t)_k}{D(\boldsymbol{\theta}^t)_k}. \tag{S.15}$$

We further simplify the notations by letting

$$\begin{aligned}\nabla_{\boldsymbol{\theta}} f_\theta(\boldsymbol{\theta}) &= \frac{1}{mn}\mathrm{diag}(\boldsymbol{s})\boldsymbol{P}(\boldsymbol{P}^\top \mathrm{diag}(\boldsymbol{s})\boldsymbol{\theta} - \boldsymbol{y}^\top) + \alpha_1 \boldsymbol{V}(\boldsymbol{\theta} - \boldsymbol{\theta}_0) + \alpha_2 \boldsymbol{V}^c \boldsymbol{\theta} \\ &= \left(\frac{1}{mn}\mathrm{diag}(\boldsymbol{s})\boldsymbol{P}\boldsymbol{P}^\top \mathrm{diag}(\boldsymbol{s}) + \alpha_1 \boldsymbol{V} + \alpha_2 \boldsymbol{V}^c\right)\boldsymbol{\theta} \\ &- \left(\frac{1}{mn}\mathrm{diag}(\boldsymbol{s})\boldsymbol{P}\boldsymbol{P}^\top \mathrm{diag}(\boldsymbol{s})\boldsymbol{y}^\top + \alpha_1 \boldsymbol{V}\boldsymbol{\theta}_0\right) \\ &= \boldsymbol{A}\boldsymbol{\theta} - \boldsymbol{b}.\end{aligned}$$

It follows immediately that $\nabla^2_{\boldsymbol{\theta}} f_\theta(\boldsymbol{\theta}) = \boldsymbol{A}$. Plugging in the values of $\nabla_{\boldsymbol{\theta}} f_\theta(\boldsymbol{\theta}^t)_k$ and $D(\boldsymbol{\theta}^t)_k$ to (S.15), we have

$$\begin{aligned}\theta_k^{t+1} &= \theta_k^t - \frac{(\boldsymbol{A}\boldsymbol{\theta}^t - \boldsymbol{b})_k}{\frac{1}{\theta_k^t}(\boldsymbol{A}\boldsymbol{\theta}^t)_k} \\ &= \theta_k^t - \frac{(\boldsymbol{A}\boldsymbol{\theta}^t)_k}{(\boldsymbol{A}\boldsymbol{\theta}^t)_k}\theta_k^t + \frac{b_k}{(\boldsymbol{A}\boldsymbol{\theta}^t)_k}\theta_k^t \\ &= \frac{b_k}{(\boldsymbol{A}\boldsymbol{\theta}^t)_k}\theta_k^t \\ &= \frac{\left(\frac{1}{mn}\mathrm{diag}(\boldsymbol{s})\boldsymbol{P}\boldsymbol{y}^\top + \alpha_1 \boldsymbol{V}\boldsymbol{\theta}_0\right)_k}{\left[\left(\frac{1}{mn}\mathrm{diag}(\boldsymbol{s})\boldsymbol{P}\boldsymbol{P}^\top \mathrm{diag}(\boldsymbol{s}) + \alpha_1 \boldsymbol{V} + \alpha_2 \boldsymbol{V}^c\right)\boldsymbol{\theta}^t\right]_k}\theta_k^t \\ &= \frac{\left(\mathrm{diag}(\boldsymbol{s})\boldsymbol{P}\boldsymbol{y}^\top + mn\alpha_1 \boldsymbol{V}\boldsymbol{\theta}_0\right)_k}{\left[(\mathrm{diag}(\boldsymbol{s})\boldsymbol{P}\boldsymbol{P}^\top \mathrm{diag}(\boldsymbol{s}) + mn(\alpha_1 \boldsymbol{V} + \alpha_2 \boldsymbol{V}^c))\boldsymbol{\theta}^t\right]_k}\theta_k^t. \qquad \text{(S.16)}\end{aligned}$$

The multiplicative form of (S.16) guarantees that $\boldsymbol{\theta}^{t+1}$ is feasible. Therefore, $\boldsymbol{\theta}^{t+1}$ satisfies (S.13).

It is straightforward to check that (S.16) also applies to zero coordinates of $\boldsymbol{\theta}^t$ ($\{\theta_k^t : \theta_k^t \in \mathbb{R}_+^K \setminus \mathbb{R}_{++}^{K_1}\}$), as zero coordinates remain zero after any multiplication. Therefore, it is the unified update step for the entire $\boldsymbol{\theta}^t \in \mathbb{R}_+^K$. Since it corresponds to the update for the (j,k)-th element in $\boldsymbol{\Theta}$, combining all elements gives (4).

We remark that (S.15) reflects the block coordinate descent nature of the multiplicative update in $\boldsymbol{\Theta}$, as observed by Kim et al. (2014). In fact, the update steps for $\boldsymbol{P}$ and $\mathrm{diag}(\boldsymbol{s})$ can be written in similar fashions.

The update step for $\boldsymbol{P}$ in (5) can be similarly derived as that for $\boldsymbol{\Theta}$. We begin with specifying the sub-problem for $\boldsymbol{p}_i$, $i = 1, \ldots, n$, treating $\boldsymbol{\Theta}$ and $\mathrm{diag}(\boldsymbol{s})$ as fixed and omitting the subscripts and the iteration number (t for $\boldsymbol{s}$ and $t+1$ for $\boldsymbol{\Theta}$) for convenience in notations:

$$\underset{\boldsymbol{p}\in\mathbb{R}_+^K}{\mathrm{argmin}}\ f_p(\boldsymbol{p}) = \frac{1}{2mn}\|\boldsymbol{y} - \boldsymbol{\Theta}\mathrm{diag}(\boldsymbol{s})\boldsymbol{p}\|_2^2 + \frac{\beta}{2}\boldsymbol{\rho}^{-1}\|\boldsymbol{p} - \boldsymbol{m}\|_2^2 + K_2(\boldsymbol{\Theta}, \boldsymbol{s}), \qquad \text{(S.17)}$$

where $K_2(\boldsymbol{\Theta}, \boldsymbol{s})$ only depends on $\boldsymbol{\Theta}$ and $\boldsymbol{s}$ and is a constant with respect to $\boldsymbol{P}$. Then, its first and second derivatives with respect to $\boldsymbol{p}$ are

$$\begin{aligned}\nabla_{\boldsymbol{p}} f_p(\boldsymbol{p}) &= \left(\frac{1}{mn}\mathrm{diag}(\boldsymbol{s})\boldsymbol{\Theta}^\top\boldsymbol{\Theta}\mathrm{diag}(\boldsymbol{s}) + \beta\boldsymbol{\rho}^{-1}\right)\boldsymbol{p} - \left(\frac{1}{mn}\mathrm{diag}(\boldsymbol{s})\boldsymbol{\Theta}^\top\boldsymbol{y} + \beta\boldsymbol{\rho}^{-1}\boldsymbol{m}\right) \\ &= \boldsymbol{C}\boldsymbol{p} - \boldsymbol{q}, \qquad \text{(S.18)}\end{aligned}$$

and

$$\nabla^2_{\boldsymbol{p}} f_p(\boldsymbol{p}) = \frac{1}{mn}\text{diag}(\boldsymbol{s})\boldsymbol{\Theta}^\top\boldsymbol{\Theta}\text{diag}(\boldsymbol{s}) + \beta\boldsymbol{\rho}^{-1} = \boldsymbol{C}. \tag{S.19}$$

To find the auxiliary function of (S.17), define $\boldsymbol{d}(\boldsymbol{p}^t) = \text{diag}(\boldsymbol{p}^t)^{-1}\nabla^2_{\boldsymbol{p}} f_p(\boldsymbol{p}^t)\boldsymbol{p}^t$, and $\boldsymbol{D}(\boldsymbol{p}^t) = \text{diag}(\boldsymbol{d}(\boldsymbol{p}^t))$, and

$$h_p(\boldsymbol{p}|\boldsymbol{p}^t) = f_p(\boldsymbol{p}^t) + \nabla_{\boldsymbol{p}} f_p(\boldsymbol{p}^t)^\top(\boldsymbol{p}-\boldsymbol{p}^t) + \frac{1}{2}(\boldsymbol{p}-\boldsymbol{p}^t)^\top\boldsymbol{D}(\boldsymbol{p}^t)(\boldsymbol{p}-\boldsymbol{p}^t). \tag{S.20}$$

On the $K_2 \leq K$ positive coordinates in $\boldsymbol{p}^t$, (S.20) is the auxiliary function of (S.17) due to Lemma 1. The verification is the same to that for $h_\theta(\boldsymbol{\theta}|\boldsymbol{\theta}^t)$ and omitted here. Next, we find $\boldsymbol{p}^{t+1}$ such that

$$\boldsymbol{p}^{t+1} = \underset{\boldsymbol{p}\in\mathbb{R}_+^{K_2}}{\text{argmin}}\ h_p(\boldsymbol{p}|\boldsymbol{p}^t).$$

This in turn corresponds to finding $\boldsymbol{p}^{t+1}$ such that

$$\nabla_{\boldsymbol{p}} h_p(\boldsymbol{p}^{t+1}|\boldsymbol{p}^t) = \nabla_{\boldsymbol{p}} f_p(\boldsymbol{p}^t) + \boldsymbol{D}(\boldsymbol{p}^t)(\boldsymbol{p}^{t+1}-\boldsymbol{p}^t) = 0,$$

given that such a $\boldsymbol{p}^{t+1}$ is feasible. After some algebra, for any $k = 1, \ldots, K_2$,

$$\begin{aligned} p_k^{t+1} &= \frac{q_k}{(\boldsymbol{C}\boldsymbol{p}^t)_k} p_k^t \\ &= \frac{\left(\frac{1}{mn}\text{diag}(\boldsymbol{s})\boldsymbol{\Theta}^\top\boldsymbol{y} + \beta\boldsymbol{\rho}^{-1}\boldsymbol{m}\right)_k}{\left\{\left(\frac{1}{mn}\text{diag}(\boldsymbol{s})\boldsymbol{\Theta}^\top\boldsymbol{\Theta}\text{diag}(\boldsymbol{s}) + \beta\boldsymbol{\rho}^{-1}\right)\boldsymbol{p}\right\}_k} p_k^t \\ &= \frac{\left(\text{diag}(\boldsymbol{s})\boldsymbol{\Theta}^\top\boldsymbol{y} + mn\beta\boldsymbol{\rho}^{-1}\boldsymbol{m}\right)_k}{\left\{\left(\text{diag}(\boldsymbol{s})\boldsymbol{\Theta}^\top\boldsymbol{\Theta}\text{diag}(\boldsymbol{s}) + mn\beta\boldsymbol{\rho}^{-1}\right)\boldsymbol{p}\right\}_k} p_k^t, \end{aligned} \tag{S.21}$$

which produces a feasible $\boldsymbol{p}^{t+1}$. Moreover, (S.21) also applies to the zero coordinates in $\boldsymbol{p}^t$. Since (S.21) corresponds to the update step for the (k, i)-th element for the $\boldsymbol{P}$ matrix, combining all of these elements yields (5).

Deriving the updates step for $\text{diag}(\boldsymbol{s})$ in (6) largely follows the same footsteps as for $\boldsymbol{\Theta}$ and $\boldsymbol{P}$. To find a suitable sub-problem, we start from the following identity

$$\boldsymbol{\Theta}\text{diag}(\boldsymbol{s})\boldsymbol{P} = \boldsymbol{\Theta}\text{diag}(\boldsymbol{s})\boldsymbol{P} = \sum_{k=1}^{K} s_k\boldsymbol{\theta}_k\boldsymbol{p}_k^\top. \tag{S.22}$$

Denoting $\boldsymbol{G}_k = \boldsymbol{\theta}_k\boldsymbol{p}_k^\top$ and only assuming $\boldsymbol{s}$ a variable, we transform the objective function

in $f(\boldsymbol{\Theta}, \boldsymbol{s}, \boldsymbol{P})$ with respect to $\boldsymbol{s}$ as follows

$$
\begin{aligned}
f_s(\boldsymbol{s}) &= \frac{1}{2mn}\|\boldsymbol{Y} - \boldsymbol{\Theta}\mathrm{diag}(\boldsymbol{s})\boldsymbol{P}\|_F^2 + C_3(\boldsymbol{\Theta}, \boldsymbol{P}) && \text{(by rewriting } f(\boldsymbol{\Theta}, \boldsymbol{s}, \boldsymbol{P})\text{)} \\
&= \frac{1}{2mn}\|\boldsymbol{Y} - \sum_{k=1}^{K} s_k \boldsymbol{G}_k\|_F^2 + C_3(\boldsymbol{\Theta}, \boldsymbol{P}) && \text{(by (S.22))} \\
&\propto \frac{1}{2}\left\{\sum_{k=1}^{K} s_k^2 \mathrm{tr}\big(\boldsymbol{G}_k^\top \boldsymbol{G}_k\big) - 2\sum_{k=1}^{K} s_k \mathrm{tr}\big(\boldsymbol{G}_k^\top \boldsymbol{Y}\big) + \sum_{k\neq l} s_k s_l \mathrm{tr}\big(\boldsymbol{G}_k^\top \boldsymbol{G}_l\big)\right\} + C_3'(\boldsymbol{\Theta}, \boldsymbol{P}) \\
&\qquad \text{(by expanding the squared Frobenius norm and dropping } m, n\text{)} \\
&= \frac{1}{2}\left\{\sum_{k=1}^{K} s_k^2 \mathrm{tr}\big(\boldsymbol{G}_k^\top \boldsymbol{G}_k\big) + \sum_{k\neq l} s_k s_l \mathrm{tr}\big(\boldsymbol{G}_k^\top \boldsymbol{G}_l\big)\right\} - \sum_{k=1}^{K} s_k \mathrm{tr}\big(\boldsymbol{G}_k^\top \boldsymbol{Y}\big) + C_3'(\boldsymbol{\Theta}, \boldsymbol{P}) \\
&\qquad \text{(by rearranging the last step)} \\
&= \frac{1}{2}\boldsymbol{s}^\top \boldsymbol{Z}\boldsymbol{s} - \boldsymbol{u}^\top \boldsymbol{s} + C_3'(\boldsymbol{\Theta}, \boldsymbol{P}), && \text{(S.23)}
\end{aligned}
$$

where $\boldsymbol{Z} \in \mathbb{R}_+^{K\times K}$, $Z_{ij} = \mathrm{tr}\big(\boldsymbol{G}_i^\top \boldsymbol{G}_j\big) = \mathrm{tr}\big(\boldsymbol{p}_i \boldsymbol{\theta}_i^\top \boldsymbol{\theta}_j \boldsymbol{p}_j^\top\big)$, $\boldsymbol{u} \in \mathbb{R}_+^K$, and $u_k = \mathrm{tr}\big(\boldsymbol{Y}^\top \boldsymbol{G}_k\big) = \mathrm{tr}\big(\boldsymbol{Y}^\top \boldsymbol{\theta}_k \boldsymbol{p}_k^\top\big)$. Both C_3 and C_3' only depend on $\boldsymbol{\Theta}$ and $\boldsymbol{P}$, hence are constant with respect to $\boldsymbol{s}$. (S.23) is then our sub-problem for $\boldsymbol{s}$. From (S.23), the first and second derivatives for $\boldsymbol{s}$ are $\nabla_{\boldsymbol{s}} f_s(\boldsymbol{s}) = \boldsymbol{Z}\boldsymbol{s} - \boldsymbol{u}$, and $\nabla_{\boldsymbol{s}}^2 f_s(\boldsymbol{s}) = \boldsymbol{Z}$. Again, define $\boldsymbol{d}(\boldsymbol{s}^t) = \mathrm{diag}(\boldsymbol{s}^t)^{-1}\nabla_{\boldsymbol{s}}^2 f_s(\boldsymbol{s}^t)\boldsymbol{s}^t$, $\boldsymbol{D}(\boldsymbol{s}^t) = \mathrm{diag}(\boldsymbol{d}(\boldsymbol{s}^t))$, and

$$
h_s(\boldsymbol{s}|\boldsymbol{s}^t) = f_s(\boldsymbol{s}^t) + \nabla_{\boldsymbol{s}} f_s(\boldsymbol{s}^t)^\top(\boldsymbol{s} - \boldsymbol{s}^t) + \frac{1}{2}(\boldsymbol{s} - \boldsymbol{s}^t)^\top \boldsymbol{D}(\boldsymbol{s}^t)(\boldsymbol{s} - \boldsymbol{s}^t). \tag{S.24}
$$

(S.24) is an auxiliary function of $f(\boldsymbol{\Theta}, \boldsymbol{s}, \boldsymbol{P})$ for $\boldsymbol{s}$ according to Lemma 1. Since $\boldsymbol{s} \in \mathbb{R}_{++}^K$, there is no need for sub-setting to positive coordinates. Thus, we obtain $\boldsymbol{s}^{t+1}$ by finding a feasible

$$
\boldsymbol{s}^{t+1} = \underset{\boldsymbol{s}\in\mathbb{R}_+^K}{\mathrm{argmin}}\ h_s(\boldsymbol{s}|\boldsymbol{s}^t). \tag{S.25}
$$

We remark that this is a slight relaxation of the problem to update $\boldsymbol{s}$, since it is originally assumed that $\boldsymbol{s} \in \mathbb{R}_{++}^K$. However, this allows us to obtain a guaranteed closed form global minimizer for (S.25) in the closed half space $\mathbb{R}_+^K$, and the resulting update satisfies is feasible in $\mathbb{R}_{++}^K$ as long as not all values of $\boldsymbol{\theta}_k$ or $\boldsymbol{p}_k^\top$ are 0 for any $k = 1, \ldots, K$, which is stated as Assumption 1(*i*) and 1(*ii*). (S.25) leads to solving

$$
\nabla_{\boldsymbol{s}} h_s(\boldsymbol{s}^{t+1}|\boldsymbol{s}^t) = \nabla_{\boldsymbol{s}} f_s(\boldsymbol{s}^t) + \boldsymbol{D}(\boldsymbol{s}^t)(\boldsymbol{s}^{t+1} - \boldsymbol{s}^t) = 0,
$$

from which we have

$$
s_k^{t+1} = \frac{u_k}{(\boldsymbol{Z}\boldsymbol{s}^t)_k} s_k^t. \tag{S.26}
$$

(S.26) corresponds to the update for the k-th coordinate in (6), which yields feasible s_k^{t+1} under the assumptions, since $Z_{ij} = \mathrm{tr}\big(\boldsymbol{G}_i^\top \boldsymbol{G}_j\big) = (\boldsymbol{\theta}_i^\top \boldsymbol{\theta}_j)\mathrm{tr}\big(\boldsymbol{p}_i \boldsymbol{p}_j^\top\big)$, and $u_k = \mathrm{tr}\big(\boldsymbol{Y}^\top \boldsymbol{G}_k\big) = \mathrm{tr}\big(\boldsymbol{Y}^\top \boldsymbol{\theta}_k \boldsymbol{p}_k^\top\big)$. Moreover,

$$
Z_{ij} = \mathrm{tr}\big(\boldsymbol{p}_i \boldsymbol{\theta}_i^\top \boldsymbol{\theta}_j \boldsymbol{p}_j^\top\big) = (\boldsymbol{\theta}_i^\top \boldsymbol{\theta}_j)(\boldsymbol{p}_j^\top \boldsymbol{p}_i), \tag{S.27}
$$

and that

$$
u_k = \mathrm{tr}\big(\boldsymbol{Y}^\top \boldsymbol{\theta}_k \boldsymbol{p}_k^\top\big) = \sum_{j=1}^{m}\sum_{i=1}^{n} y_{ji}\theta_{jk}p_{ki}. \tag{S.28}
$$

For the denominator in (S.26), we see that

$$(\boldsymbol{Z}\boldsymbol{s}^t)_k = \sum_{l=1}^{K} Z_{kl} s_l^t \geq Z_{kk} s_k^t = \|\boldsymbol{\theta}_k\|_2^2 \|\boldsymbol{p}_k\|_2^2 s_k^t > 0, \tag{S.29}$$

where the last step is a result of (S.27) and the stability of updates based Assumption 1(*i*) and 1(*ii*), which will be discussed in details below. As for the numerator of (S.26), taking from the stability of updates due to Assumption 1(*i*) and 1(*ii*) and assuming $\theta_{j_k k} > 0$ and $p_{ki_k} > 0$, by (S.28), we have

$$u_k = \sum_{j=1}^{m} \sum_{i=1}^{n} y_{ji} \theta_{jk} p_{ki} \geq y_{j_k i_k} \theta_{j_k k} p_{ki_k} > 0,$$

as long as $y_{j_k i_k} > 0$. Given Assumption 1(*iii*), we conclude that $\boldsymbol{s}^{t+1} \in \mathbb{R}_{++}^K$.

A.3. Stability of The Update Steps

To justify the claim that for any j and k, $\theta_{jk}^{t+1} > 0$ whenever $\theta_{jk}^t > 0$ for $t = 0, 1, 2, \ldots$, notice that the denominator of (S.16) is automatically positive whenever $\alpha_1, \alpha_2 > 0$. For guaranteeing that its numerator is also positive, we want to show

$$\left(\mathrm{diag}(\boldsymbol{s}^t)\boldsymbol{P}^t \boldsymbol{y}_j^\top\right)_k = s_k^t \sum_{i=1}^{n} p_{ki}^t y_{ji} \geq s_k^t p_{ki_k}^t y_{ji_k} > 0, \tag{S.30}$$

where the index i_k is defined as in Assumption 1(*ii*). By Assumption 1(*iii*), $y_{ji_k} > 0$, and $s_k^t > 0$ as discussed above. Thus, (S.30) holds whenever $p_{ki_k}^t > 0$. Similarly, for proving the claim that for any k and i, $p_{ki}^{t+1} > 0$ whenever $p_{ki}^t > 0$, notice that the denominator for (S.21) is also positive whenever $\beta > 0$. To prove that its numerator is positive, we need to show

$$\left(\mathrm{diag}(\boldsymbol{s}^t)(\boldsymbol{\Theta}^{t+1})^\top \boldsymbol{y}_i\right)_k = s_k^t \sum_{j=1}^{m} \theta_{jk}^{t+1} y_{ji} \geq s_k^t \theta_{j_k k}^{t+1} y_{j_k i} > 0, \tag{S.31}$$

where j_k is defined as in Assumption 1(*i*). (S.31) holds whenever $\theta_{j_k k}^{t+1} > 0$.

Since by Assumption 1(*ii*) and Assumption 1(*i*), $p_{ki_k}^0 > 0$ and $\theta_{j_k k}^0 > 0$. Therefore, we can show that $p_{ki_k}^t > 0$ and $\theta_{j_k k}^{t+1} > 0$ for any $t = 0, 1, 2, \ldots$ by induction through (S.30) and (S.31). Thus, the claims on the stability of updates hold.

B. Proof of Proposition 1

Let $\boldsymbol{Y}_0 = \boldsymbol{G}_0 \boldsymbol{P}$, where $\boldsymbol{Y}_0$ is the $K \times n$ sub-matrix of $\boldsymbol{Y}$ that corresponds to the rows of $\boldsymbol{G}_0$. Assume another decomposition $\boldsymbol{Y}_0 = \boldsymbol{G}_0^\circ \boldsymbol{P}^\circ$ exists. Since $\boldsymbol{G}_0 = \mathrm{diag}(\boldsymbol{g}_0)$, we have, for each $j = 1, \ldots, K$, $\boldsymbol{y}_{0_j}^\top = \boldsymbol{p}_j^\top g_{0_j} = (\boldsymbol{p}^\circ)_j^\top g_{0_j}^\circ$. Then, $(\boldsymbol{p}^\circ)_j^\top = \boldsymbol{p}_j^\top \frac{g_{0_j}}{g_{0_j}^\circ} = \boldsymbol{p}_j^\top \omega_j$. Then, $(\boldsymbol{P}^\circ)^\top = \boldsymbol{P}^\top \mathrm{diag}(\boldsymbol{\omega})$, where $\boldsymbol{\omega} = \begin{bmatrix}\omega_1 & \ldots & \omega_K\end{bmatrix}^\top$. Post-multiply both sides by $\mathbf{1}_K$, we obtain $\mathbf{1}_n = \boldsymbol{P}^\top \mathrm{diag}(\boldsymbol{\omega}) \mathbf{1}_K = \boldsymbol{P}^\top \boldsymbol{\omega}$. Therefore, $\boldsymbol{P}^\top(\boldsymbol{\omega} - \mathbf{1}_K) = \mathbf{0}_n$ and we conclude that $\boldsymbol{\omega} = \mathbf{1}_K$ because $\boldsymbol{P}$ is full-rank. This finishes our proof.

C. Proof of Proposition 2

Let $\boldsymbol{Y}_0 = \boldsymbol{G}_0\boldsymbol{P}$, where $\boldsymbol{Y}_0$ is the $K \times n$ sub-matrix of $\boldsymbol{Y}$ that corresponds to the rows of $\boldsymbol{G}_0$. Assume another decomposition $\boldsymbol{Y}_0 = \boldsymbol{G}_0^\circ\boldsymbol{P}^\circ$ exists. From (7), we have, for each $l = 1, \ldots, K-1$, $\boldsymbol{y}_{0_l}^\top = \boldsymbol{p}_l^\top g_{0_l} = (\boldsymbol{p}_l^\circ)^\top g_{0_l}^\circ$. This leads to $(\boldsymbol{p}^\circ)_l^\top = \boldsymbol{p}_l^\top \omega_l$.
For $l = K$, we have

$$\boldsymbol{y}_{0_K}^\top = \boldsymbol{p}_K^\top g_{0_K} + \sum_{l=1}^{K-1} \boldsymbol{p}_l^\top c_l = (\boldsymbol{p}^\circ)_K^\top g_{0_K}^\circ + \sum_{l=1}^{K-1} (\boldsymbol{p}^\circ)_l^\top c_l^\circ,$$

which implies

$$(\boldsymbol{p}^\circ)_K^\top = \frac{1}{g_{0_K}} \sum_{l=1}^{K-1} (c_l^\circ \omega_l - c_l)\boldsymbol{p}_l^\top + \frac{g_{0_K}}{g_{0_K}^\circ}\boldsymbol{p}_K^\top = \sum_{l=1}^{K-1} \delta_l \boldsymbol{p}_l^\top + \omega_K \boldsymbol{p}_K^\top,$$

where $\delta_l = \frac{c_l^\circ \omega_l - c_l}{g_{0_K}}$. Altogether, we obtain that

$$(\boldsymbol{P}^\circ)^\top = \boldsymbol{P}^\top \begin{bmatrix} \omega_1 & \ldots & 0 & \delta_1 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \ldots & \omega_{K-1} & \delta_{K-1} \\ 0 & \ldots & 0 & \omega_K \end{bmatrix} = \boldsymbol{P}^\top \boldsymbol{Z}.$$

Post-multiply both sides by $\mathbf{1}_K$, we obtain $\mathbf{1}_n = \boldsymbol{P}^\top \boldsymbol{Z}\mathbf{1}_K$. Therefore, $\boldsymbol{P}^\top(\boldsymbol{Z}\mathbf{1}_K - \mathbf{1}_K) = \mathbf{0}_n$. Since $\boldsymbol{P}^\top$ is full-rank, we know that $\boldsymbol{Z}\mathbf{1}_K = \mathbf{1}_K$, meaning that $\omega_K = 1$ and $\omega_l + \delta_l = 1$ for $l = 1, \ldots, K-1$. Plug in $\delta_l = \frac{c_l^\circ \omega_l - c_l}{g_{0_K}}$, we have $\delta_l = \frac{c_l^\circ - c_l}{g_{0_K} + c_l^\circ}$ and $\omega_l = \frac{g_{0_K} + c_l}{g_{0_K} + c_l^\circ}$. This gives us the conclusion in (8).

To obtain a bound for each c_l°, we notice that from the first case of (8), $0 \le \omega_l \le 1/p_{li}, \forall i$, which leads to $c_l^\circ \ge \max_{i \in \{1,\ldots,n\}} p_{li}(g_{0_K} + c_l) - g_{0_K}$. In the second case, we have $0 \le \sum_{l=1}^{K-1} \delta_l p_{li} + p_{Ki} \le 1, \forall i$. The first inequality holds automatically because $\sum_{l=1}^{K-1} \delta_l p_{li} + p_{Ki} = \sum_{l=1}^{K-1}(1-\omega_l)p_{li} + p_{Ki} = 1 - \sum_{l=1}^{K-1} \omega_l p_{li}$, where we need $0 \le \sum_{l=1}^{K-1} \omega_l p_{li} \le 1$ for any i. The left inequality holds since $\omega_l \ge 0$. The right inequality is just the condition $\sum_{l=1}^{K-1} \frac{g_{0_K} + c_l}{g_{0_K} + c_l^\circ} p_{li} \le 1$. To show that the second equality also holds, we can see that $\sum_{l=1}^{K-1} \delta_l p_{li} + p_{Ki} \le 1$ implies $\sum_{l=1}^{K-1} \delta_l p_{li} \le 1 - p_{Ki} = \sum_{l=1}^{K-1} p_{li}$. This is always satisfied because $\delta_l = 1 - \omega_l \ge 1$ as $\omega_l \ge 0$.

D. Proof of Theorem 1

To prove Theorem 1, we leverage on the convergence properties of the block successive upper-bound minimization (BSUM) algorithms proved by Razaviyayn et al. (2013). Namely, we can view Algorithm 1 as a BSUM algorithm by counting a total of $m + n + 1$ separate blocks which the algorithm minimizes successively: m for updating $\boldsymbol{\Theta}^t$, n for $\boldsymbol{P}^t$, and one for $\boldsymbol{s}^t$. To utilize such properties, we need to prove the following proposition, which states the conditions on the update steps' auxiliary functions for Algorithm 1 to formally qualify as BSUM, which is translated from Assumption 2 of Razaviyayn et al. (2013):

Proposition 3 (Algorithm 1 is BSUM)**.** *$h_\theta(\boldsymbol{\theta}_j|\boldsymbol{\theta}_j^t)$, $h_p(\boldsymbol{p}_i|\boldsymbol{p}_i^t)$, and $h_s(\boldsymbol{s}|\boldsymbol{s}^t)$ satisfy the following conditions for any $j = 1, 2, \ldots, m$ and $i = 1, 2, \ldots, n$: (1) $h_{\theta_j}(\boldsymbol{\theta}_j|\boldsymbol{\theta}_j^t) \geq f_\theta(\boldsymbol{\theta}_j^t)$; $h_p(\boldsymbol{p}_i|\boldsymbol{p}_i^t) \geq f_p(\boldsymbol{p}_i^t)$; $h_s(\boldsymbol{s}|\boldsymbol{s}^t) \geq f_s(\boldsymbol{s}^t)$; (2) $h_{\theta_j}(\boldsymbol{\theta}_j|\boldsymbol{\theta}_j^t)$, $h_p(\boldsymbol{p}_i|\boldsymbol{p}_i^t)$, and $h_s(\boldsymbol{s}|\boldsymbol{s}^t)$ are continous on their domains; (3) Any directional derivative of $h_{\theta_j}(\boldsymbol{\theta}_j|\boldsymbol{\theta}_j^t)$, $h_p(\boldsymbol{p}_i|\boldsymbol{p}_i^t)$, and $h_s(\boldsymbol{s}|\boldsymbol{s}^t)$ within their domains are equal to the direactional derivative of $f_\theta(\boldsymbol{\theta}_j^t)$, $f_p(\boldsymbol{p}_i^t)$, and $f_s(\boldsymbol{s}^t)$ respectively.*

Proof. The proof is quite straightforward. (1) and (2) are direct from the construction of the surrogate functions and Lemma 1, which also manifests the majorization-minimization (MM) property (Lange, 2016) of each block's update. (3) is seen by differentiating (S.12), (S.20), and (S.24), which leads to $\nabla_{\boldsymbol{\theta}_j} h_\theta(\boldsymbol{\theta}_j^t|\boldsymbol{\theta}_j^t) = \nabla_{\boldsymbol{\theta}_j} f_\theta(\boldsymbol{\theta}_j^t), \forall j$, $\nabla_{\boldsymbol{p}_i} h_p(\boldsymbol{p}_i^t|\boldsymbol{p}_i^t) = \nabla_{\boldsymbol{p}_i} f_p(\boldsymbol{p}_i^t), \forall i$, and $\nabla_{\boldsymbol{s}} h_s(\boldsymbol{s}^t|\boldsymbol{s}^t) = \nabla_{\boldsymbol{s}} f_s(\boldsymbol{s}^t)$. Equal gradients automatically guarantee equal directional derivatives, proving the proposition. □

Now that Algorithm 1 is found to be BSUM, Theorem 1 can be proved directly by the results of Theorem 2(b) in (Razaviyayn et al., 2013). But a few additional conditions presented in the following propositions need to be verified, both of which are proved thereafter.

Proposition 4 (Compact Sub-level Sets)**.** *The sub-level set $\mathcal{X}^0 : \{(\boldsymbol{\Theta}, \boldsymbol{s}, \boldsymbol{P}) : f(\boldsymbol{\Theta}, \boldsymbol{s}, \boldsymbol{P}) \leq f(\boldsymbol{\Theta}^0, \boldsymbol{s}^0, \boldsymbol{P}^0)\}$ of $f(\boldsymbol{\Theta}, \boldsymbol{s}, \boldsymbol{P})$ given the update steps in Algorithm 1 is compact.*

Proposition 5 (Unique Global Minimizer of Auxiliary Functions)**.** *Given Assumption 1, $\forall t \in \mathbb{R}_+$ and for all i and j, the auxiliary functions $h_\theta(\boldsymbol{\theta}_j|\boldsymbol{\theta}_j^t)$, $h_p(\boldsymbol{p}_i|\boldsymbol{p}_i^t)$, and $h_s(\boldsymbol{s}|\boldsymbol{s}^t)$ each have a unique global minimizer in $\mathbb{R}_+^K$.*

D.1. Proof of Proposition 4

We first introduce the definition of coercive functions through:

Definition 2 (Coercivity)**.** *A function $f(\boldsymbol{X}_1, \ldots, \boldsymbol{X}_N) : V_1 \times \cdots \times V_N \to \mathbb{R}$, where V_i is a Euclidean vector space endorsed with the L_2 norm if its elements are vectors or with the Frobenius norm if the elements are matrices, is called coercive if $\forall i$, $\|\boldsymbol{X}_i\| \to \infty$ implies $f(\boldsymbol{X}_1, \ldots, \boldsymbol{X}_N) \to \infty$.*

In fact, continuous coercive functions always have compact sub-level sets, as shown by the following lemma:

Lemma 2. *If a continuous function $f(\boldsymbol{X}_1, \ldots, \boldsymbol{X}_N) : V_1 \times \cdots \times V_N \to \mathbb{R}$ is jointly coercive, then its sublevel set $\mathcal{X}^r := \{(\boldsymbol{X}_1, \ldots, \boldsymbol{X}_N) : f(\boldsymbol{X}_1, \ldots, \boldsymbol{X}_N) \leq r\}$ is compact for any $r \in \mathbb{R}$.*

Proof. Because f is continuous and the space $f(\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n) \leq r$ is closed, $\mathcal{X}^r$ is closed. We prove the boundedness of $\mathcal{X}^r$ by contrapositive. We let $\mathcal{X}^r = \mathcal{X}_1^r \times \cdots \times \mathcal{X}_N^r$, where $\mathcal{X}_i^r \subseteq V_i$. If $\mathcal{X}^r$ is unbounded, then one of $\mathcal{X}_i^r$ is unbounded. If $\mathcal{X}_i^r$ is unbounded, we can find a sequence of $\{\boldsymbol{X}_k\}_{k=1}^\infty$ such that $\|\boldsymbol{X}_k\| \to \infty$ as $k \to \infty$, which implies f is not jointly coercive. Therefore, whenever f is jointly coercive, $\mathcal{X}^r = \mathcal{X}_1^r \times \cdots \times \mathcal{X}_N^r$ is bounded. Since Euclidean spaces admit the Heine-Borel property, $\mathcal{X}^r$ is compact. □

Coming back to the proof of Proposition 4, it is easy to check that $f(\boldsymbol{\Theta}, \boldsymbol{s}, \boldsymbol{P})$ is continuously differentiable. To prove that it is also coercive, we need to show that $f(\boldsymbol{\Theta}, \boldsymbol{s}, \boldsymbol{P})$ tends to ∞ whenever $\|\boldsymbol{\Theta}\|_F$, $\|\boldsymbol{s}\|_2$, or $\|\boldsymbol{P}\|_F$ does so.

Beginning with the case for $\boldsymbol{\Theta}$, we have, by $f(\boldsymbol{\Theta}, \boldsymbol{s}, \boldsymbol{P})$,

$$\begin{aligned} f(\boldsymbol{\Theta}, \boldsymbol{s}, \boldsymbol{P}) &\geq \frac{\alpha_1}{2}\|\boldsymbol{\Delta} \odot \boldsymbol{\Theta} - \boldsymbol{\Theta}_0\|_F^2 + \frac{\alpha_2}{2}\|\boldsymbol{\Delta}^c \odot \boldsymbol{\Theta}\|_F^2 \\ &\geq \min\left(\frac{\alpha_1}{2}, \frac{\alpha_2}{2}\right)\left(\|\boldsymbol{\Delta} \odot \boldsymbol{\Theta} - \boldsymbol{\Theta}_0\|_F^2 + \|\boldsymbol{\Delta}^c \odot \boldsymbol{\Theta}\|_F^2\right) \\ &\geq \min\left(\frac{\alpha_1}{2}, \frac{\alpha_2}{2}\right)\left(\|\boldsymbol{\Delta} \odot \boldsymbol{\Theta}\|_F^2 - \|\boldsymbol{\Theta}_0\|_F^2 + \|\boldsymbol{\Delta}^c \odot \boldsymbol{\Theta}\|_F^2\right) \\ &= \min\left(\frac{\alpha_1}{2}, \frac{\alpha_2}{2}\right)\left(\|\boldsymbol{\Theta}\|_F^2 - \|\boldsymbol{\Theta}_0\|_F^2\right) \\ &= \min\left(\frac{\alpha_1}{2}, \frac{\alpha_2}{2}\right)\|\boldsymbol{\Theta}\|_F^2\left(1 - \frac{\|\boldsymbol{\Theta}_0\|_F^2}{\|\boldsymbol{\Theta}\|_F^2}\right). \end{aligned} \tag{S.32}$$

Since $\min\left(\frac{\alpha_1}{2}, \frac{\alpha_2}{2}\right) > 0$ and $\frac{\|\boldsymbol{\Theta}_0\|_F^2}{\|\boldsymbol{\Theta}\|_F^2} \leq \frac{mK_0M_\Theta}{\|\boldsymbol{\Theta}\|_F^2} \to 0$ as $\|\boldsymbol{\Theta}\|_F \to \infty$ by Assumption 2, (S.32) then guarantees that $f(\boldsymbol{\Theta}, \boldsymbol{s}, \boldsymbol{P}) \to \infty$ when $\|\boldsymbol{\Theta}\|_F \to \infty$.

In the case for $\boldsymbol{P}$, whenever $\|\boldsymbol{P}\|_F \to \infty$, there is a k such that $\boldsymbol{p}_k^\top$ satisfies $\|\boldsymbol{p}_k^\top\|_2 \to \infty$. Also, since $\boldsymbol{p}_k \in \mathbb{R}_+^n$, there exists a column indexed i_k as defined in Assumption 1(*ii*) such that $p_{ki_k} \to \infty$. Then,

$$\begin{aligned} f(\boldsymbol{\Theta}, \boldsymbol{s}, \boldsymbol{P}) &\geq \frac{\beta}{2}\sum_{l=1}^K r_l^{-1}\|\boldsymbol{p}_l^\top - \boldsymbol{m}_l\|_2^2 \geq \frac{\beta}{2}r_k^{-1}\|\boldsymbol{p}_k^\top - \boldsymbol{m}_k\|_2^2 \\ &= \frac{\beta}{2}r_k^{-1}\sum_{i=1}^n (p_{ki} - \nu_k)^2 \geq \frac{\beta}{2}r_k^{-1}(p_{ki_k} - \nu_k)^2 \to \infty, \end{aligned}$$

as ν_k is assumed fixed and $\beta > 0$.

Lastly for $\boldsymbol{s}$, we can similarly argue that, because $\boldsymbol{s} \in \mathbb{R}_{++}^K$, $\|\boldsymbol{s}\|_2 \to \infty$ implies there is a k such that $s_k \to \infty$. From the steps leading to (S.23), we have

$$\begin{aligned} f(\boldsymbol{\Theta}, \boldsymbol{s}, \boldsymbol{P}) &\geq \frac{1}{2mn}\|\boldsymbol{Y} - \boldsymbol{\Theta}\mathrm{diag}(\boldsymbol{s})\boldsymbol{P}\|_F^2 = \frac{1}{2mn}\|\boldsymbol{Y} - \sum_{k=1}^K s_k\boldsymbol{G}_k\|_F^2 \\ &\geq \frac{1}{2mn}\left(\|\sum_{l=1}^K s_l\boldsymbol{G}_l\|_F^2 - \|\boldsymbol{Y}\|_F^2\right) \\ &= \frac{1}{2mn}\left(s_k\|\boldsymbol{G}_k\|_F^2 - \|\boldsymbol{Y}\|_F^2\right) \to \infty. \end{aligned}$$

The last line is because $\|\boldsymbol{G}_k\| > 0$ due to Assumption 1(*i*) and Assumption 1(*ii*) and that $\|\boldsymbol{Y}\|_F \leq mnM_Y < \infty$ by Assumption 2. Combining the cases for $\boldsymbol{\Theta}$, $\boldsymbol{P}$, and $\boldsymbol{s}$ completes the proof that $f(\boldsymbol{\Theta}, \boldsymbol{s}, \boldsymbol{P})$ is coercive.

Since $f(\boldsymbol{\Theta}, \boldsymbol{s}, \boldsymbol{P})$ is continuous and coercive, the set $\mathcal{X}^0 : \{(\boldsymbol{\Theta}, \boldsymbol{s}, \boldsymbol{P}) : f(\boldsymbol{\Theta}, \boldsymbol{s}, \boldsymbol{P}) \leq f(\boldsymbol{\Theta}^0, \boldsymbol{s}^0, \boldsymbol{P}^0)\}$ is compact according to Lemma 2. We have concluded the proof of Proposition 4.

D.2. Proof of Proposition 5

Since $h_\theta(\boldsymbol{\theta}_j|\boldsymbol{\theta}_j^t)$, $h_p(\boldsymbol{p}_i|\boldsymbol{p}_i^t)$, and $h_s(\boldsymbol{s}|\boldsymbol{s}^t)$ are quadratic functions, proving Proposition 5 is equivalent to showing their Hessians are positive definite. Also, we only focus on the positive coordinates of $\boldsymbol{\theta}_j^t$ and $\boldsymbol{p}_i^t$ since other coordinates will remain zero through updates, and the proposition is vacuously true. By (S.12), (S.20), and (S.24), these Hessians are

$\boldsymbol{D}(\boldsymbol{\theta}_j^t)$, $\boldsymbol{D}(\boldsymbol{p}_i^t)$, and $\boldsymbol{D}(\boldsymbol{s}^t)$ respectively. Since they are all diagonal matrices, we want to show that their diagonal values are all positive.

For $\boldsymbol{D}(\boldsymbol{\theta}_j^t)$, we have $D(\boldsymbol{\theta}_j^t)_{kk} = \boldsymbol{d}(\boldsymbol{\theta}_j^t)_k = \frac{(\boldsymbol{A}\boldsymbol{\theta}_j^t)_k}{\theta_{jk}^t}$. The denominator of $D(\boldsymbol{\theta}_j^t)_{kk}$ is positive. Dropping the index j, its numerator is

$$\begin{aligned}\left\{\left(\mathrm{diag}(\boldsymbol{s})\boldsymbol{P}\boldsymbol{P}^\top\mathrm{diag}(\boldsymbol{s}) + mn(\alpha_1\boldsymbol{V} + \alpha_2\boldsymbol{V}^c)\right)\boldsymbol{\theta}_j^t\right\}_k &\geq \left\{mn(\alpha_1\boldsymbol{V} + \alpha_2\boldsymbol{V}^c)\boldsymbol{\theta}_j^t\right\}_k \\ &\geq mn\min(\alpha_1, \alpha_2)\theta_{jk}^t > 0,\end{aligned}$$

since $\boldsymbol{V} + \boldsymbol{V}^c = \boldsymbol{I}$. Hence, $D(\boldsymbol{\theta}_j^t)_{kk}$ is positive.

For $\boldsymbol{D}(\boldsymbol{p}_i^t)$, from (S.18) and (S.19), we also have $D(\boldsymbol{p}_i^t)_{kk} = \boldsymbol{d}(\boldsymbol{p}_i^t)_k = \frac{(\boldsymbol{C}\boldsymbol{p}_i^t)_k}{p_{ki}^t}$. Likewise, the denominator is positive, and the numerator after dropping the index i is

$$\begin{aligned}\left\{\left(\mathrm{diag}(\boldsymbol{s})\boldsymbol{\Theta}^\top\boldsymbol{\Theta}\mathrm{diag}(\boldsymbol{s}) + mn\beta\boldsymbol{\rho}^{-1}\right)\boldsymbol{p}_i^t\right\}_k &\geq \left(mn\beta\boldsymbol{\rho}^{-1}\boldsymbol{p}_i^t\right)_k \\ &= mn\beta r_k^{-1}p_{ki}^t > 0.\end{aligned}$$

Lastly, $D(\boldsymbol{s}^t)_{kk}$ is already shown to be positive from the conclusion of (S.29). This completes the proof of Proposition 5.

To finally prove Theorem 1, we combine the results of Propositions 3-5, and apply these conclusions to Theorem 2(b) in Razaviyayn et al. (2013). This way, we have shown that Algorithm 1 converges to a set of stationary points of $f(\boldsymbol{\Theta}, \boldsymbol{s}, \boldsymbol{P})$.

E. Additional Results For The Real Data Example

In Section 5, it was posited that bulk and gene signature matrices whose expression values were measured in TPM would lose the information on cell-type mRNA amounts, necessitating the inclusion of $\boldsymbol{s}$ described in the underlying model (1). To verify this claim, we re-ran the data analysis on the same data in TPM as in Section 5 using the same empirical estimates $\boldsymbol{M}$ and $\boldsymbol{\rho}$, the same tolerance parameter, and the same tuning grid for the hyperparaters, and the same random seed, but coerced $\mathrm{diag}(\boldsymbol{s})$ to be the identity matrix (i.e. coerced the cell-type mRNA amounts to 1) throughout updates in Algorithm 1. We observed that the estimated proportions for PBMC cell types for both samples on Day 0 deviates more from those measured by flow cytometry when compared to when $\mathrm{diag}(\boldsymbol{s})$ is not coerced, as shown in the following Supplementary Figure 1.

Table 1: Means and ranges of PBMC cell type proportions calculated from the reference values.

	T cells	**B cells**	**NK cells**	**Monocytes**	**Other cells**
Means	0.6275	0.0625	0.1075	0.15	0.0525
Ranges	0.275	0.055	0.145	0.1	0.2

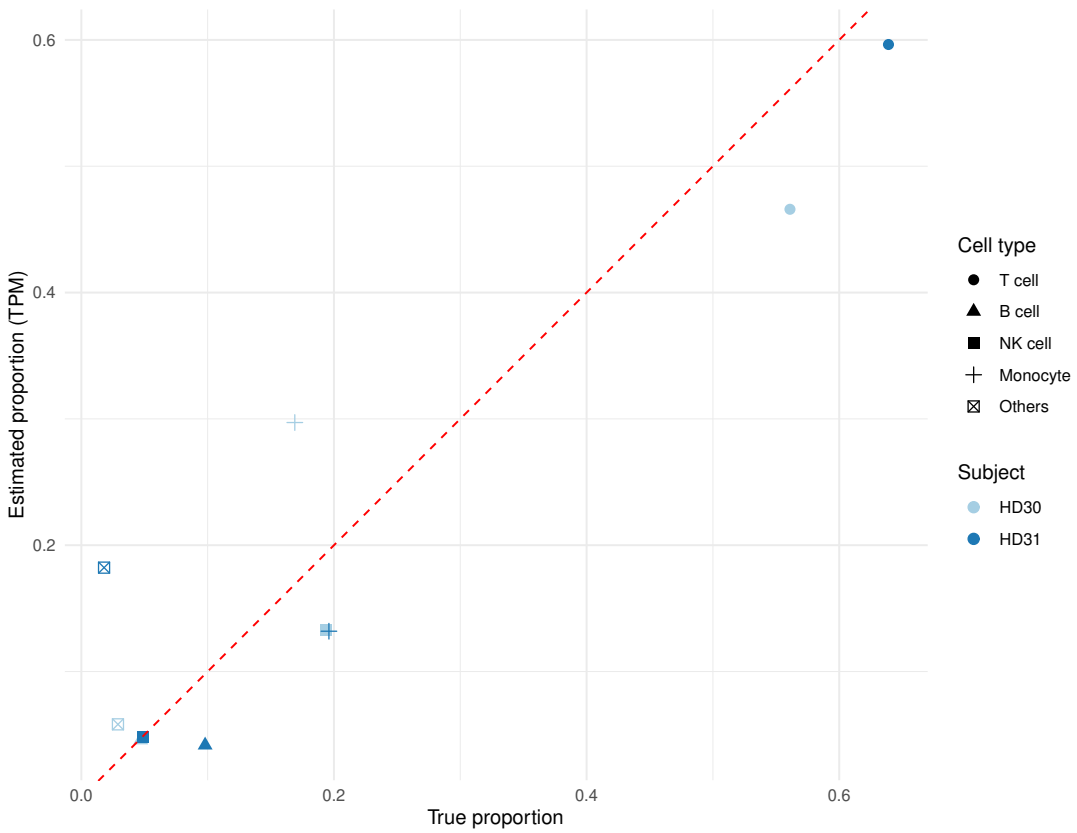

Figure 1: Estimated vs. true proportions of PBMC cell types of two subjects on Day 0 using ARTdeConv with the cell-type mRNA amounts coerced to 1s.

References

Razaviyayn, M., Hong, M., & Luo, Z.-Q. (2013). *A unified convergence analysis of block successive minimization methods for nonsmooth optimization. SIAM Journal on Optimization*, 23(2), 1126–1153.

Lin, C.-J. (2007). *On the convergence of multiplicative update algorithms for nonnegative matrix factorization. IEEE Transactions on Neural Networks*, 18(6), 1589–1596.

Lee, D., & Seung, H. S. (2000). *Algorithms for non-negative matrix factorization.* In *Advances in neural information processing systems*, 13.

Kim, J., He, Y., & Park, H. (2014). *Algorithms for nonnegative matrix and tensor factorizations: A unified view based on block coordinate descent framework. Journal of Global Optimization*, 58(2), 285–319.

Lange, K. (2016). *MM optimization algorithms.*