CONTRIBUTED PAPER

A Critical Survey of Deconvolution Methods for Separating Cell Types in Complex Tissues

This paper focuses on in silico deconvolution of signals associated with complex tissues into their constitutive cell-type components and surveys a variety of models, methods, and assumptions underlying deconvolution techniques.

By Shahin Mohammadi, Neta Zuckerman, Andrea Goldsmith, and Ananth Grama

ABSTRACT | Identifying properties and concentrations of components from an observed mixture, known as deconvolution, is a fundamental problem in signal processing. It has diverse applications in fields ranging from hyperspectral imaging to noise cancellation in audio recordings. This paper focuses on *in silico* deconvolution of signals associated with complex tissues into their constitutive cell-type-specific components and a quantitative characterization of the cell types. Deconvolving mixed tissues/cell types is useful in the removal of contaminants (e.g., surrounding cells) from tumor biopsies, as well as in monitoring changes in the cell population in response to treatment or infection. In these contexts, the observed signal from the mixture of cell types is assumed to be a convolution, using a linear instantaneous (LI) mixing process, of the expression levels of genes in constitutive cell types. The goal is to use known signals corresponding to individual cell types and a model of the mixing process to cast the deconvolution problem as a suitable optimization problem. In this paper, we present a survey and in-depth analysis of models, methods, and assumptions underlying deconvolution techniques. We investigate the choice of the different loss functions for evaluating estimation error, constraints on solutions, preprocessing and data filtering, feature selection, and regularization to enhance the quality of solutions and the impact of these choices on the performance of commonly used regression-based methods for deconvolution. We assess different combinations of these factors and use detailed statistical measures to evaluate their effectiveness. Some of these combinations have been proposed in the literature, whereas others represent novel algorithmic choices for deconvolution. We identify shortcomings of current methods and avenues for further investigation. For many of the identified shortcomings, such as normalization issues and data filtering, we provide new solutions. We summarize our findings in a prescriptive step-by-step process, which can be applied to a wide range of deconvolution problems.

KEYWORDS | Deconvolution; feature selection; gene expression; linear regression; loss function; range filtering; regularization

Manuscript received October 4, 2015; revised May 27, 2016; accepted September 6, 2016. Date of publication October 25, 2016; date of current version January 18, 2017. This work was supported by the Center for Science of Information (CSoI), an NSF Science and Technology Center, under Grant Agreement CCF-0939370, and by NSF, under Grant BIO 1124962. *(Corresponding authors: Shahin Mohammadi and Ananth Grama.)*

S. Mohammadi and **A. Grama** are with the Department of Computer Sciences, Purdue University, West Lafayette, IN 47907 USA (e-mail: mohammadi@purdue.edu; ayg@cs.purdue.edu).

N. Zuckerman is with the Department of Electrical Engineering, Stanford University, Stanford, CA 94305 USA and also with Genentech Inc., South San Francisco, CA 94080 USA (e-mail: zuckerman@gmail.com).

A. Goldsmith is with the Department of Electrical Engineering, Stanford University, Stanford, CA 94305 USA (e-mail: andrea@wsl.stanford.edu).

Digital Object Identifier: 10.1109/JPROC.2016.2607121

I. INTRODUCTION

Source separation, or deconvolution, is the problem of estimating individual signal components from their mixtures. This problem arises when source signals are transmitted through a mixing channel and the mixed sensor readings are observed. Source separation has applications in a variety of fields. One of its early applications was in processing audio signals [1]–[4]. Here, mixtures of different sound sources, such as speech or music, are recorded

simultaneously using several microphones. Various frequencies are convolved by the impulse response of the room, and the goal is to separate one or several sources from this mixture. This has direct applications in speech enhancement, voice removal, and noise cancellation in recordings from populated areas. In hyperspectral imaging, the spectral signature of each pixel is observed. This signal is the combination of pure spectral signatures of constitutive elements mixed according to their relative abundance. In satellite imaging, each pixel represents sensor readings for different patches of land at multiple wavelengths. Individual sources correspond to reflectances of materials at different wavelengths that are mixed according to the material composition of each pixel [5]–[9].

Beyond these domains, deconvolution has applications in removing noise from biomedical sensors. Tracing electrical current in the brain is widely used as a proxy for spatiotemporal patterns of brain activity. These patterns have significant clinical applications in diagnosis and prediction of epileptic seizures as well as characterizing different stages of sleep in patients with sleep disorders. Electroencephalography (EEG) and magnetoencephalography (MEG) are two of the most commonly used techniques for cerebral imaging. These techniques measure voltage fluctuations and changes in the electromagnetic fields, respectively. Superconducting quantum interference device (SQUID) sensors used in the latter technology are susceptible to magnetic coupling due to geometry and must be shielded carefully against magnetic noise. Deconvolution techniques are used to separate different noise sources and ameliorate the effect of electrical and magnetic coupling in these devices [10]–[13].

At a high level, mixing channels can be classified as follows: 1) linear or nonlinear; 2) instantaneous, delayed, or convolutive; and 3) over/under determined. When neither the sources nor the mixing process is available, the problem is known as blind source separation (BSS). This problem is highly under-determined in general, and additional constraints, such as independence among sources, sparsity, or nonnegativity, are typically enforced on the sources in practical applications. On the other hand, a new class of methods has been developed recently, which is known as semi or guided BSS [1], [3], [4], [11]. In these methods, additional information is available *a priori* on the approximate behavior of either sources or the mixing process. In this paper, we focus on the class of overdetermined, linear instantaneous (LI) mixing processes, for which a deformed prior on sources is available. In this case, the parameters of the linear mixer as well as the true identity of the original sources are to be determined.

In the context of molecular biology, deconvolution methods have been used to identify constituent cell types in a tissue and their relative proportions. The inherent heterogeneity of tissue samples makes it difficult to identify separated, cell-type-specific signatures, i.e., the precise gene expression levels for each cell type. Relative changes in cell proportions, combined with variations attributed to the changes in the biological conditions, such as disease state, complicate identification of true biological signals from mere technical variations. Changes in tissue composition are often indicative of disease progression or drug response. For example, coupled depletion of specific neuronal cells with the gradual increase in the glial cell population is indicative of neurodegenerative disorders. An increasing proportion of malignant cells as well as a growing fraction of tumor infiltrating lymphocytes (TIL) compared to surrounding cells, directly influence tumor growth, metastasis, and clinical outcomes for patients [14], [15]. Deconvolving tissue biopsies allows further investigation of the interaction between tumor and micro-environmental cells and its role in the progression of cancer.

The expression level of genes, which is a proxy for the number of present copies of each gene product, is one of the most common source factors used for separating cell types and tissues. In the linear mixing model, the expression of each gene in a complex mixture is estimated as a linear combination of the expression of the same gene in the constitutive cell types. *In silico* deconvolution methods for separating complex tissues can be coarsely classified as either full deconvolution, in which both cell-type-specific expressions and the percentages of each cell type are estimated, or partial deconvolution methods, whereby one of these data sources is used to compute the other [16]. These two classes loosely relate to BSS and guided-BSS problems. Note that in cases in which relative abundances are used to estimate cell-type-specific expressions, the problem is highly under-determined, whereas in the complementary case of computing percentages from noisy expressions of purified cells, it is highly over-determined. In the former case, we typically have only a handful of known cell types with known percentages, and we are trying to estimate unknown expression value for thousands of genes. On the other hand, in the latter case, we are using known expression value of all genes (or a selected subset that is typically much larger than the number of cell types) to compute percentages of a small population of constituting cells. Thus, in the case of an over-determined system, the key is to select the most reliable features that satisfy the linearity assumption. We provide an in-depth review of the recent deconvolution methods in Section II-E.

In contrast to computational methods, a variety of experimental cell separation techniques have been proposed to enrich samples for cell types of interest. However, these experimental methods not only involve significant time, effort, and expense but may also result in insufficient RNA abundance for further quantification of gene expression. In this case, amplification steps may

introduce technical artifacts into the gene expression data. Furthermore, sorting of cell types must be embedded in the experiment design for the desired subset of cells, and any subsequent separation is infeasible. Computational methods, on the other hand, are capable of sorting mixtures at different levels of resolution and for arbitrary cell-type subsets of interest.

The organization of the remainder of the paper is as follows. In Section II-A, we introduce the basic terminology from biology needed to formally define the deconvolution problem in the context of quantifying cell-type fractions in complex tissues. The formal definition of the deconvolution problem and its relationship to linear regression is defined in Section II-B. In Sections II-C and II-D, we review different choices and examples of the objective function used in regression. An overview of computational methods for biological deconvolution is provided in Section II-E. Data sets and evaluation measures used in this study are described in Sections III-A and III-B, respectively. The effect of the loss function, constraint enforcement, range filtering, and feature selection choices on the performance of deconvolution methods is evaluated systematically in Sections III-D–III-H. Finally, a summary of our findings is provided in Section IV.

II. BACKGROUND AND NOTATION

A. Biological Underpinnings

The central dogma of biology describes the flow of genetic information within cells—the genetic code, represented in DNA molecules, is first transcribed to an intermediate construct, called messenger RNA (mRNA), which in turn translates into proteins. These proteins are the functional workhorses of the cell. Genes, defined as the minimal coding sections of the DNA, contain the recipe for making proteins. These instructions are utilized dynamically by the cell to adapt to different conditions. The amounts of various proteins in a cell can be measured at a time point. This corresponds to the level of protein expression. This process is limited by the availability of high-quality antibodies that can specifically target each protein. The amount of active mRNA in a cell, however, can be measured at the genome scale by using high-throughput technologies such as microarrays and RNASeq. The former is an older technology that relies on the binding affinity of complementary base pairs (alphabets used in the DNA/RNA molecules), whereas the latter is a newer technique, using next generation sequencing (NGS). This technique estimates gene expression based on the overlap of mRNA fragments with known genomic features. Since microarrays have been used for years, extensive databases from different studies are publicly available. RNASeq data sets, in comparison, are relatively smaller but growing rapidly in scale and coverage. Both of these technologies provide reliable proxies for the amount of proteins in cells, with RNASeq being more sensitive, especially for lowly expressed genes. A drawback of these methods, however, is that the true protein expression is also regulated by additional mechanisms, such as post-transcriptional modifications, which cannot be assayed at the mRNA level.

The expression level of genes is tightly regulated in different stages of cellular development and in response to environmental changes. In addition to these biological variations due to cellular state, intermediate steps in each technology introduce technical variations in repeated measurement of gene expression in the same cell type. To enhance reproducibility of measurements, one normally includes multiple instances of the same cell type in each experiment, known as technical replicates. The expression profiles from these experiments provide a snapshot of the cell under different conditions. In addition to biological variation of genes within the same cell type, there is an additional level of variation when we look across different cell types. Some genes are ubiquitously expressed in all cell types to perform housekeeping functions, whereas other genes exhibit specificity or selectivity for one or a group of cell types, respectively. A compendium of expression profiles of different cells at different developmental stages is the data substrate for *in silico* deconvolution of complex tissues.

B. Deconvolution: Formal Definition

We introduce formalisms and notation used in discussing different aspects of *in silico* deconvolution of biological signals. We focus on models that assume linearity; that is, the expression signature of the mixture is a weighted sum of the expression profile for its constitutive cell types. In this case, sources are cell-type-specific references and the mixing process is determined by the relative fraction of cell types in the mixture.

We first introduce the mathematical constructs used.

- $\mathbf{M} \in \mathbb{R}^{n \times p}$: Mixture matrix, where each entry $\mathbf{M}(i, j)$ represents the raw expression of gene i, $1 \leq i \leq n$, in heterogeneous sample j, $1 \leq j \leq p$. Each sample, represented by $\boldsymbol{m}$, is a column of the matrix $\mathbf{M}$, and is a combination of gene expression profiles from constituting cell types in the mixture.
- $\mathbf{H} \in \mathbb{R}^{n \times r}$: Reference signature matrix for the expression of primary cell types, with multiple biological/technical replicates for each cell type. In this matrix, rows correspond to the same set of genes as in $\mathbf{M}$, columns represent replicates, and there is an underlying grouping among columns that collects profiles corresponding to the same cell type.
- $\mathbf{G} \in \mathbb{R}^{n \times q}$: Reference expression profile, where the expression of similar cell types in matrix $\mathbf{H}$ is represented by the average value.

- $\mathbf{C} \in \mathbb{R}^{q \times p}$: Relative proportions of each cell type in the mixture sample. Here, rows correspond to cell types, and columns represent samples in mixture matrix $\mathbf{M}$.

Using this notation, we can formally define deconvolution as an optimization problem that seeks to identify "optimal" estimates for matrices $\mathbf{G}$ and $\mathbf{C}$, denoted by $\hat{\mathbf{G}}$ and $\hat{\mathbf{C}}$, respectively. Since $\mathbf{G}$ and/or $\mathbf{C}$ are not known *a priori*, we use an approximation that is based on the linearity assumption. In this case, we aim to find $\hat{\mathbf{G}}$ and $\hat{\mathbf{C}}$ such that their product is close to the mixture matrix, $\mathbf{M}$. Specifically, given a function $\boldsymbol{\delta}$ that measures the distance between the true and approximated solutions, also referred to as the loss function, we aim to solve

$$\min_{0 \leq \hat{\mathbf{G}}, \hat{\mathbf{C}}} \boldsymbol{\delta}(\hat{\mathbf{G}}\hat{\mathbf{C}}, \mathbf{M}). \tag{1}$$

In partial deconvolution, either $\mathbf{C}$ or $\mathbf{G}$, or their noisy representation, is known *a priori* and the goal is to find the other unknown matrix. When matrix $\mathbf{G}$, referred to as the reference profile, is known, the problem is overdetermined, and we seek to distinguish features (genes) that closely conform to the linearity assumption, from the rest of the (variable) genes. In this case, we can solve the problem individually for each mixture sample. Let us denote by $\boldsymbol{m}$ and $\hat{\boldsymbol{c}}$ the expression profile and estimated cell-type proportion of a mixture sample, respectively. Then, we can rewrite (1) as

$$\min_{0 \leq \hat{\boldsymbol{c}}} \boldsymbol{\delta}(\mathbf{G}\hat{\boldsymbol{c}}, \boldsymbol{m}). \tag{2}$$

This formulation is essentially a linear regression problem, with an arbitrary loss function. On the other hand, in the case of full deconvolution, we can still estimate $\mathbf{C}$ in a column-by-column fashion. However, estimating $\mathbf{G}$ is highly under-determined, and we must use additional sources to restrict the search space. One such source of information is the variation across samples in $\mathbf{M}$, depending on the cell-type concentrations in the latest estimated value of $\mathbf{C}$. In general, most regression-based methods for full deconvolution use an iterative scheme that starts from either noisy estimates of $\mathbf{G}$ and $\mathbf{C}$ or a random sample that satisfies given constraints on these matrices, and successively improves over this initial approximation. This iterative process can be formalized as follows:

$$\begin{aligned} \hat{\mathbf{C}} &\leftarrow \underset{0 \leq \hat{\mathbf{C}}}{\mathbf{argmin}} \Big(\boldsymbol{\delta}(\hat{\mathbf{G}}\hat{\mathbf{C}} - \mathbf{M}) \Big) \\ \hat{\mathbf{G}} &\leftarrow \left(\underset{0 \leq \hat{\mathbf{G}}}{\mathbf{argmin}} \Big(\boldsymbol{\delta}(\hat{\mathbf{C}}^T\hat{\mathbf{G}}^T - \mathbf{M})^T \Big) \right)^T . \end{aligned} \tag{3}$$

Please note that the updating $\hat{\mathbf{G}}$ is typically row-wise (for each gene), whereas the update of $\hat{\mathbf{C}}$ is column-wise (for each sample). Non-negative matrix factorization (NMF) is a dimension reduction technique that aims to factor each column of the given input matrix as a non-negative weighted sum of non-negative basis vectors, with the number of basis vectors being equal or less than the number of columns in the original matrix. The alternating nonnegative least squares formulation (ANLS) for solving NMF can be formulated by using the framework introduced in (3). There are additional techniques for solving NMF, including the multiplicative updating rule and the hierarchical alternating least squares (HALS) methods, all of which are special cases of block-coordinate descent [17]. Two of the most common loss functions used in NMF are the Frobenius and Kullback-Leibler (KL) divergence [17].

In addition to nonnegativity (NN), an additional *sum-to-one (STO)* constraint is typically applied over columns of the matrix $\hat{C}$, or the sample-specific vector $\hat{\boldsymbol{c}}$. This constraint restricts the search space, which can potentially enhance the accuracy of the results, and simplifies the interpretation of values in $\hat{\boldsymbol{c}}$ as relative percentages. Finally, another fundamental assumption that is mostly neglected in prior work is the **similar cell quantity (SCQ)** constraint. The similar cell quantity assumption states that all reference profiles and corresponding mixtures must be normalized to ensure that they represent the expression level of the "same number of cells." If this constraint is not satisfied, differences in the cell-type counts directly affect concentrations by rescaling the estimated coefficients to adjust for the difference.

In this paper, we focus on different loss functions ($\boldsymbol{\delta}$ functions) as well as the role of constraint enforcement strategies, in estimating $\hat{\boldsymbol{c}}$. These constitute the key building blocks of both partial and full deconvolution methods.

C. Choice of Objective Function

In linear regression, often a slightly different notation is used, which we describe here. We subsequently relate it to the deconvolution problem. Given a set of samples, $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^m$, where $\boldsymbol{x}_i \in \mathbb{R}^k$ and $y_i \in \mathbb{R}$, the regression problem seeks to find a function $f(\boldsymbol{x})$ that minimizes the aggregate error over all samples. Let us denote the fitting error by $r_i = y_i - f(\boldsymbol{x}_i)$. Using this notation, we can write the regression problem as

$$\underset{f \in \mathcal{F}}{\mathbf{argmin}} \sum_{i=1}^{m} \mathcal{L}(r_i) \tag{4}$$

where the loss function $\mathcal{L}$ measures the cost of estimation error. We focus on the class of linear functions; that is, $f_{\boldsymbol{w}}(\boldsymbol{x}) = \boldsymbol{w}^T\boldsymbol{x}$, for which we have $r_i = y_i - \boldsymbol{w}^T\boldsymbol{x}_i$. In this formulation, y_i corresponds to the expression level of a

gene in the mixture, vector $\boldsymbol{x}_i$ is the expression level of the same gene in the reference cell types, and $\boldsymbol{w}$ is the fraction of each cell type in the mixture. We can represent $\{\boldsymbol{x}_i\}_{i=1}^m$ in the compact form by matrix $\mathbf{X}$ in which row i corresponds to $\boldsymbol{x}_i$.

In cases wherein the number of parameters is greater than the number of samples; that is, matrix $\mathbf{X}$ is a fat matrix, minimizing (4) directly can result in the overfitting problem. Furthermore, when features (columns of $\mathbf{X}$) are highly correlated, the solution may change drastically in response to small changes in the samples, specifically among the correlated features. This condition, known as multicollinearity, can result in inaccurate estimates in which coefficients of similar features are greatly different. To remedy these problems, we can add a regularization term, which incorporates additional constraints (such as sparsity or flatness) to enhance the stability of results. We rewrite the problem with the added regularizer as

$$\underset{\boldsymbol{w}\in\mathbb{R}^k}{\mathbf{argmin}}\left\{\underbrace{\sum_{i=1}^{m}\mathcal{L}(y_i-\boldsymbol{w}^T\boldsymbol{x}_i)}_{\text{Overall loss}}+\underbrace{\lambda\mathcal{R}(\boldsymbol{w})}_{\text{Regularizer}}\right\} \tag{5}$$

where the λ parameter controls the relative importance of estimation error versus regularization. There are different choices and combinations for the loss function $\mathcal{L}$ and regularizer function $\mathcal{R}$, which we describe in the following sections.

1) Choice of Loss Functions: There are a variety of options for suitable loss functions. Some of these functions are known to be asymptotically optimal for a given noise density, whereas others may yield better performance in practice when assumptions underlying the noise model are violated. We summarize the most commonly used set of loss functions.

- If we assume that the underlying model is perturbed by Gaussian white noise, the squared or quadratic loss, denoted by $\mathcal{L}_2$, is known to be asymptotically optimal. This loss function is used in classical least squares regression and is defined as

$$\mathcal{L}_2(r_i)=r_i^2=(y_i-\boldsymbol{w}^T\boldsymbol{x}_i)^2.$$

- Absolute deviation loss, denoted by $\mathcal{L}_1$, is the optimal choice if noise follows a Laplacian distribution. Formally, it is defined as

$$\mathcal{L}_1(r_i)=|r_i|=|y_i-\boldsymbol{w}^T\boldsymbol{x}_i|.$$

Compared to $\mathcal{L}_2$, the choice of $\mathcal{L}_1$ is preferred in the presence of outliers, as it is less sensitive to extreme values

- Huber's loss function, denoted by $\mathcal{L}_{\text{huber}}^{(M)}$, is a parametrized combination of $\mathcal{L}_1$ and $\mathcal{L}_2$. The main idea is that $\mathcal{L}_2$ loss is more susceptible to outliers, whereas it is more sensitive to small estimation errors. To combine the best of these two functions, we can define a half-length parameter M, which we use to transition from $\mathcal{L}_2$ to $\mathcal{L}_1$. More formally

$$\mathcal{L}_{\text{Huber}}^{(M)}(r_i)=\begin{cases}r_i^2, & \text{if } |r_i|\le M\\ M(2|r_i|-M), & \text{otherwise.}\end{cases}$$

- The loss function used in support vector regression (SVR) is the ϵ-insensitive loss, denoted by $\mathcal{L}_\epsilon^{(\epsilon)}$. Similar to Huber loss, there is a transition phase between small and large estimation errors. However, ϵ-insensitive loss does not penalize the errors that are smaller than a threshold. Formally, we define ϵ-insensitive loss as

$$\begin{aligned}\mathcal{L}_\epsilon^{(\epsilon)}(r_i)&=\mathbf{max}(0,|r_i|-\epsilon)\\&=\begin{cases}0, & \text{if } |r_i|\le\epsilon\\ |r_i|-\epsilon, & \text{otherwise.}\end{cases}\end{aligned}$$

Fig. 1 provides a visual representation of these loss functions in which we use $M=1$ and $\epsilon=1/2$ for the Huber and ϵ-insensitive loss functions, respectively. Note that for small residual values, $|r_i|\le M=1$, Huber and square loss are equivalent. However, outside this region, Huber loss becomes linear.

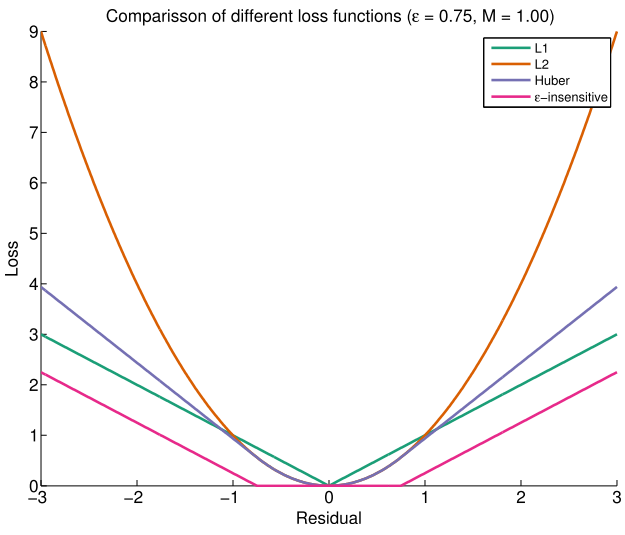


Fig. 1. ***Comparison of different loss functions.***

2) Choice of Regularizers: When the reference profile contains many cell types that may not exist in mixtures, or in cases wherein constitutive cell types are highly correlated, regularizing the objective function can sparsify the solution or enhance the conditioning of the problem. We describe two commonly used regularizers here.

- The norm-2 regularizer is used to shrink the regression coefficient vector $\boldsymbol{w}$ to ensure that it is as flat as possible. A common use of this regularizer is in conjunction with $\mathcal{L}_2$ loss to remedy the multicollinearity problem in classical least squares regression. This regularizer is formally defined as

$$\mathcal{R}_2(\boldsymbol{w}) = \|\boldsymbol{w}\|_2^2 = \sum_{i=1}^{k} w_i^2. \tag{6}$$

- Another common regularizer is the norm-1 regularizer, which is used to enforce sparsity over $\boldsymbol{w}$. Formally, it can be defined as

$$\mathcal{R}_1(\boldsymbol{w}) = \|\boldsymbol{w}\|_1 = \sum_{i=1}^{k} |w_i|. \tag{7}$$

In addition to these two regularizers, their combinations have also been introduced in the literature. One such example is elastic net, which uses a convex combination of the two; that is, $\mathcal{R}_{\text{elastic}}(\boldsymbol{w}) = \alpha\mathcal{R}_1(\boldsymbol{w}) + (1-\alpha)\mathcal{R}_2(\boldsymbol{w})$. Another example is group LASSO, which, given a grouping G among cell types, enforces flatness among members of the group while enhancing the sparsity pattern across groups. This regularizer function can be written as $\mathcal{R}_{\text{group}} = \sum_{G_i} \mathcal{L}_2(\boldsymbol{w}(G_i))$, where $\boldsymbol{w}(G_i)$ is the weight of cell types in the ith group.

D. Examples of Objective Functions Used in Practice

1) Ordinary Least Squares (OLS): The formulation of OLS is based on squared loss, $\mathcal{L}_2$. Formally, we have

$$\begin{aligned}\mathbf{min}_{\boldsymbol{w}} \left\{ \sum_{i=1}^{m} \mathcal{L}_2(r_i) \right\} &= \mathbf{min}_{\boldsymbol{w}} \left\{ \sum_{i=1}^{m} (y_i - \boldsymbol{w}^T \boldsymbol{x}_i)^2 \right\} \\ &= \mathbf{min}_{\boldsymbol{w}} \|y - \mathbf{X}\boldsymbol{w}\|_2^2\end{aligned}$$

where row i of the matrix $\mathbf{X}$, also known as the design matrix, corresponds to $\boldsymbol{x}_i$. This formulation has a closed form solution given by

$$\hat{\boldsymbol{w}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\boldsymbol{y}.$$

In this formulation, we can observe that norm-2 regularization is especially useful in cases wherein the matrix $\mathbf{X}$ is ill-conditioned and near-singular; that is, columns are dependent on each other. By shifting $\mathbf{X}^T\mathbf{X}$ toward the identity matrix, we ensure that the eigenvalues are farther from zero, which enhances the conditioning of the resulting combination.

2) Ridge Regression: One of the main issues with the OLS formulation is that the design matrix, $\mathbf{X}$, should have full column rank k. Otherwise, if we have highly correlated variables, the solution suffers from the multicollinearity problem. This condition can be remedied by incorporating a norm-2 regularizer. The resulting formulation, known as ridge regression, is as follows:

$$\mathbf{min}_{\boldsymbol{w}} \left\{ \sum_{i=1}^{m} \mathcal{L}_2(r_i) + \lambda\mathcal{R}_2(\boldsymbol{w}) \right\} = \mathbf{min}_{\boldsymbol{w}} \|y - \mathbf{X}\boldsymbol{w}\|_2^2 + \lambda\|\boldsymbol{w}\|_2^2.$$

Similar to OLS, we can differentiate w.r.t. $\boldsymbol{w}$ to find the close form solution for Ridge regression given by

$$\hat{\boldsymbol{w}} = (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T\boldsymbol{y}.$$

3) Least Absolute Selection and Shrinkage Operator (LASSO) Regression: Combining the OLS with a norm-1 regularizer, we have the LASSO formulation

$$\mathbf{min}_{\boldsymbol{w}} \left\{ \sum_{i=1}^{m} \mathcal{L}_2(r_i) + \lambda\mathcal{R}_1(\boldsymbol{w}) \right\} = \mathbf{min}_{\boldsymbol{w}} \|y - \mathbf{X}\boldsymbol{w}\|_2^2 + \lambda\|\boldsymbol{w}\|_1.$$

This formulation is especially useful for producing sparse solutions by introducing zero elements in vector $\boldsymbol{w}$. However, while being convex, it does not have a closed form solution.

4) Robust Regression: It is known that $\mathcal{L}_2(\boldsymbol{r})$ is dominated by the largest elements of the residual vector $\boldsymbol{r}$, which makes it sensitive to outliers. To remedy this problem, different robust regression formulations have been proposed that use alternative loss functions. Two of the best-known formulations are based on the $\mathcal{L}_1$ and $\mathcal{L}_{\text{huber}}$ loss functions. The $\mathcal{L}_1$ formulation can be written as

$$\begin{aligned}\mathbf{min}_{\boldsymbol{w}} \left\{ \sum_{i=1}^{m} \mathcal{L}_1(r_i) \right\} &= \mathbf{min}_{\boldsymbol{w}} \left\{ \sum_{i=1}^{m} |y_i - \boldsymbol{w}^T \boldsymbol{x}_i| \right\} \\ &= \mathbf{min}_{\boldsymbol{w}} \|y - \mathbf{X}\boldsymbol{w}\|_1.\end{aligned}$$

However, for the Huber loss function, while it can be defined similarly, it is usually formulated as an

alternative convex Quadratic Program (QP)

$$\begin{aligned}&\mathbf{min}_{\boldsymbol{x},\boldsymbol{z},\boldsymbol{t}}\left\{\frac{1}{2}\|\boldsymbol{z}\|_2^2+M\mathbf{1}^T\boldsymbol{t}\right\}\\&\text{Subject to}:\ -\boldsymbol{t}\le\mathbf{X}\boldsymbol{w}-\boldsymbol{y}-\boldsymbol{z}\le\boldsymbol{t}\end{aligned}\tag{8}$$

which can be solved more efficiently using the following equivalent QP variant [18]:

$$\begin{aligned}&\mathbf{min}_{\boldsymbol{x},\boldsymbol{z},\boldsymbol{r},\boldsymbol{s}}\left\{\frac{1}{2}\|\boldsymbol{z}\|_2^2+M\mathbf{1}^T(\boldsymbol{r}+\boldsymbol{s})\right\}\\&\text{Subject to}:\begin{cases}\mathbf{X}\boldsymbol{w}-\boldsymbol{y}-\boldsymbol{z}=\boldsymbol{r}-\boldsymbol{s}\\0\le\boldsymbol{r},\boldsymbol{s}.\end{cases}\end{aligned}\tag{9}$$

In both of these formulations, the scalar M corresponds to half-length parameter of the Huber's loss function.

5) Support Vector Regression: In machine learning, support vector regression (SVR) is a commonly used technique that finds a regression by maximizing the margins around the estimated separator hyperplane from the closest data points on each side of it. This margin provides the region in which estimation errors are ignored. SVR has been recently used to deconvolve biological mixtures, where it has been shown to outperform other methods [15]. One of the variants of SVR is ϵ-SVR in which parameter ϵ defines the margin or the ϵ-tube. The primal formulation of ϵ-SVR with linear kernel can be written as [19]

$$\begin{aligned}&\mathbf{min}_{\boldsymbol{w},\boldsymbol{\xi}_i^+,\boldsymbol{\xi}_i^-}\left\{\frac{1}{2}\|\boldsymbol{w}\|_2^2+C\sum_{i=1}^{m}\left(\boldsymbol{\xi}_i^++\boldsymbol{\xi}_i^-\right)\right\}\\&\text{Subject to}:\begin{cases}y_i-\boldsymbol{w}\cdot\boldsymbol{x}_i\le\epsilon+\boldsymbol{\xi}_i^+\\-\left(\epsilon+\boldsymbol{\xi}_i^-\right)\le y_i-\boldsymbol{w}\cdot\boldsymbol{x}_i\\0\le\boldsymbol{\xi}_i^+,\boldsymbol{\xi}_i^-\end{cases}\end{aligned}\tag{10}$$

in which, given the unit norm assumption introduced in Section II-B, we assume that $b=0$. The dual problem for the primal in (10) can be written in matrix form as

$$\begin{aligned}&\mathbf{max}_{\boldsymbol{\alpha}^+,\boldsymbol{\alpha}^-}\Big\{\mathbf{1}^T\left((\boldsymbol{\alpha}^+-\boldsymbol{\alpha}^-)\odot\boldsymbol{y}\right)-\epsilon\mathbf{1}^T(\boldsymbol{\alpha}^++\boldsymbol{\alpha}^-)\\&\qquad-(\boldsymbol{\alpha}^+-\boldsymbol{\alpha}^-)^T\mathbf{K}(\boldsymbol{\alpha}^+-\boldsymbol{\alpha}^-)\Big\}\\&\text{Subject to}:\begin{cases}\mathbf{1}^T(\boldsymbol{\alpha}^+-\boldsymbol{\alpha}^-)=0\\0\le\boldsymbol{\alpha}^+,\boldsymbol{\alpha}^-\le C.\end{cases}\end{aligned}\tag{11}$$

In this formulation, $\mathbf{1}$ is a vector of all ones, $\odot$ is the elementwise product, and $\mathbf{K}$ is the kernel matrix defined as $\mathbf{K}=\mathbf{X}\mathbf{X}^T$. The dual formulation is often used to solve ϵ-SVR, because it can be easily extended to use different kernel functions to map $\boldsymbol{x}_i$ to a d-dimensional nonlinear feature space. Additionally, when $m\ll k$, such as the case of high-dimensional feature spaces, it provides a better way to solve the SVR problem. However, the primal problem provides a more straightforward interpretation. In addition, in the case where $k\ll m$, it provides superior performance. To show the similarity with (5), we can rewrite (10) using the ϵ-insensitive loss function as follows:

$$\mathbf{min}_{\boldsymbol{w}}\left\{\sum_{i=1}^{m}\mathcal{L}_\epsilon(y_i-\boldsymbol{w}^T\boldsymbol{x}_i)+\lambda\mathcal{R}_2(\boldsymbol{w})\right\}\tag{12}$$

where $\lambda=1/2C$ [20].

E. Overview of Prior in Silico Deconvolution Methods

A majority of existing deconvolution methods fall into two groups: they either use a regression-based framework to compute **G**, **C**, or both, or perform statistical inference over a probabilistic model. Abbas *et al.* [21] present one of the early regression-based methods for estimating **C**. This method is designed to identify cell-type concentrations from a known reference profile of immune cells. Their method is based on ordinary least squares (OLS) regression and does not consider either nonnegativity or sum-to-one constraints explicitly, but rather it enforces these constraints implicitly after the optimization procedure. An extension of this approach is proposed by Qiao *et al.* [22], which uses non-negative least squares (NNLS) to explicitly enforce nonnegativity as part of the optimization. Gong *et al.* [23] present a quadratic programming (QP) framework to explicitly encode both constraints in the optimization problem formulation. They also propose an extension to this method, called DeconRNASeq, which applies the same QP framework to RNASeq data sets. More recently, Newman *et al.* [15] propose robust linear regression (RLR) and ν-SVR regression instead of $\mathcal{L}_2$ based regression, which is highly susceptible to noise. Digital cell quantification (DCQ) [24] is another approach designed for monitoring the immune system during infection. Compared to prior methods, DCQ forces sparsity by combining $\mathcal{R}_2$ and $\mathcal{R}_1$ regularization into an elastic net. This regularization is essential for successfully identifying the subset of active cells at each stage, given the larger number of cell types included in their panel (213 immune cell subpopulations). In contrast to these techniques, Shen-Orr *et al.* [25] propose a method called *csSAM*, which is specifically designed to identify genes that are differentially expressed among purified cell types. The core of this method is regression over matrix **C** to estimate matrix **G**.

Full regression-based methods use variations of block-coordinate descent to successively identify better estimates for both **C** and **G** [17]. Venet *et al.* [26] present one of the early methods in this class, which uses an NMF-like method coupled with a heuristic to decorrelate columns of **G** in each iteration. Repsilber *et al.* [27] propose an algorithm called *deconf*, which uses alternating non-negative least squares (ANLS) for solving NMF, without the decorrelation step of Vennet *et al.*, while implicitly applying constraints on **C** and **G** at each iteration. Inspired by the work of Pauca *et al.* on hyperspectral image deconvolution [8], Zuckerman *et al.* [28] propose an NMF method based on the Frobenius norm for gene expression deconvolution. They use gradient descent to solve for **C** and **G** at each step, which converges to a local optimum of the objective function. Given that the expression domain of cell-type-specific markers is restricted to unique cells in the reference profile, Gaujoux *et al.* [29] present a semi-supervised NMF (ssNMF) method that explicitly enforces an orthogonality constraint at each iteration over the subset of markers in the reference profile. This constraint both enhances the convergence of the NMF algorithm and simplifies the matching of columns in the estimated cell type expression to the columns of the reference panel, **G**. The digital sorting algorithm (DSA) [30] works as follows. If concentration matrix **C** is known *a priori*, it directly uses quadratic programming (QP) with added constraints on the lower/upper bound of gene expressions to estimate matrix **G**. Otherwise, if fractions are also unknown, it uses the average expression of given marker genes that are expressed in only one cell type, combined with the *STO* constraint, to estimate concentrations matrix **C** first. Population-specific expression analysis (PSEA) [31] performs a linear least squares regression to estimate quantitative measures of cell-type-specific expression levels in a similar fashion as the update equation for estimating $\hat{\mathbf{G}}$ in (3). In cases wherein the matrix **C** is not known *a priori*, *PSEA* exploits the average expression of marker genes that are exclusively expressed in one of the reference profiles as reference signals to track the variation of cell-type fractions across multiple mixture samples. More recently, a new class of methods, collectively referred to as convex analysis of mixtures (CAM), have been proposed to directly infer marker genes from the mixture profiles [32]–[34]. CAM family of methods aim to use a geometric approach to identify corners of the scatter simplex for mixed expression profiles. The key to success of these methods is a recently proven bijection between scatter simplex of mixed profiles and a transformed (rotated and compressed) version of the scatter simplex for constituent cell types [32]. To this end, "marker genes" are concentrated around the corners of this simplex. After identifying these semi-orthogonal markers, one can recover cell type percentages using any of marker-based methods mentioned above, such as PSEA [31] or DSA [30]. Similar techniques have been also proposed earlier to infer tumor phylogeny using microarray measurements of tumor populations [35].

In addition to regression-based methods, a large class of methods is based on probabilistic modeling of gene expression. Erikkila *et al.* [36] introduce a method called *DSection*, which formulates the deconvolution problem using a Bayesian model. It incorporates a Bayesian prior over the noisy observation of given concentration parameters to account for their uncertainty and employs a MCMC sampling scheme to estimate the posterior distribution of the parameters/latent variables, including **G** and a refined version of **C**. The *in silico* NanoDissection method [37] uses a classification algorithm based on linear SVM coupled with an iterative adjustment process to refine a set of provided, positive and negative, marker genes and infer a ranked list of genome-scale predictions for cell-type-specific markers. Quon *et al.* [38] propose a probabilistic deconvolution method called PERT, which estimates a global, multiplicative perturbation vector to correct for the differences between provided reference profiles and the true cell types in the mixture. PERT formulates the deconvolution problem in a similar framework as latent dirichlet allocation (LDA) and uses the conjugate gradient descent method to cyclically optimize the joint likelihood function with respect to each latent variable/parameter. Finally, microarray microdissection with analysis of differences (MMAD) [39] incorporates the concept of the effective RNA fraction to account for source and sample-specific bias in the cell-type fractions for each gene. They propose different strategies depending on the availability of additional data sources. In cases in which no additional information is available, they identify genes with the highest variation in mixtures as markers and assign them to different reference cell types using k-means clustering, and finally use these *de novo* markers to compute cell-type fractions. MMAD uses a MLE approach over the residual sum of squares to estimate unknown parameters in their formulation.

In this paper, we focus on partial deconvolution methods for recovering matrix **C** using given reference profiles for constituent tissues/cell types. Table 1 summarizes different combinations proposed in the literature so far. We will cover all these configurations as well as missing combinations that have not been studied yet.

III. RESULTS AND DISCUSSION

We now present a comprehensive evaluation of various formulations for solving deconvolution problems. Some of these algorithimic combinations have been proposed in literature, while others represent new algorithmic choices. We systematically assess the impact of these algorithmic choices on the performance of in-silico deconvolution.

Table 1 Summary of Regression-Based Methods for Computing Matrix C

Reference	Method	Loss	Non-negativity	Sum-to-one	Regularizer
Abbas *et al.* (2009)	Ordinary Least Squares (OLS)	$\mathcal{L}_2$	Imp	Imp	-
Gong *et al.* (2011)	Quadratic Programming	$\mathcal{L}_2$	Exp	Exp	-
Qiao *et al.* (2012)	Non-negative Least Squares (NNLS)	$\mathcal{L}_2$	Exp	Imp	-
DCQ– Altboum *et al.* (2014)	Elastic Net	$\mathcal{L}_2$	Imp	Imp	$\mathcal{L}_1/\mathcal{L}_2$
RLR– Newman *et al.* (2015)	Robust Linear Regression (RLR)	Huber	Imp	Imp	-
CIBERSORT– Newman *et al.* (2015)	ν-SVR	ϵ-insensitive	Imp	Imp	$\mathcal{L}_2$

A. Data Sets

1) *In vivo* mixtures with known percentages: We use a total of five data sets with known mixtures. We use CellMix to download and normalize these data sets [40], which uses the *soft* format data available from Gene Expression Omnibus (GEO).
   - BreatBlood [23] (GEO ID: *GSE29830*): Breast and blood from human specimens are mixed in three different proportions, and each of the mixtures is measured three times, with a total of nine samples.
   - CellLines [21] (GEO ID: *GSE11058*): Mixture of human cell lines Jurkat (T-cell leukemia), THP-1 (acute monocytic leukemia), IM-9 (B lymphoblastoid multiple myeloma) and Raji (Burkitt B-cell lymphoma) in four different concentrations, each of which is repeated three times, resulting in a total of 12 samples.
   - LiverBrainLung [25] (GEO ID: *GSE19830*): This data set contains three different rat tissues, namely, brain, liver, and lung tissues, which are mixed in 11 different concentrations, with each mixture having three technical replicates, for a total of 33 samples.
   - RatBrain [31] (GEO ID: *GSE19380*): This contains four different cell types, namely, rat's neuronal, astrocytic, oligodendrocytic and microglial cultures, and two replicates of five different mixing proportions, for a total of 10 samples.
   - Retina [41] (GEO ID: *GSE33076*): This data set pools together retinas from two different mouse lines and mixed them in eight different combinations and three replicates for each mixture, resulting in a total of 24 samples.
2) Mixtures with available cell-sorting data through flow-cytometry: For this experiment, we use two data sets available from Qiao *et al.* [22]. We directly download these data sets from the supplementary material of the paper. These data sets are postprocessed by the supervised normalization of microarrays (SNM) method to correct for batch effects. Raw expression profiles are also available for download under GEO ID *GSE40830*. This data set contains two subdata sets:
   - PERT_Uncultured: This data set contains uncultured human cord blood mono-nucleated and lineage-depleted (Lin-) cells on the first day.
   - PERT_Cultured: This data set contains culture-derived lineage-depleted human blood cells after four days of cultivation.

Table 2 summarizes overall statistics related to each of these data sets.

B. Evaluation Measures

Let us denote the actual and estimated coefficient matrices by $\mathbf{C}$ and $\hat{\mathbf{C}}$, respectively. We first normalize these measures to ensure that each column sums to one. Then, we define the corresponding percentages as $\mathbf{P} = 100 \times \mathbf{C}_{\text{norm}}$ and $\hat{\mathbf{P}} = 100 \times \hat{\mathbf{C}}_{\text{norm}}$. Finally, let $r_{jk} = p_{jk} - \hat{p}_{jk}$ be the residual estimation error of cell type k in sample j. Using this notation, we can define three commonly used measures of estimation error as follows.

1) Mean absolute difference (mAD): This is among the easiest measures to interpret. It is defined as the average of all differences for different cell-type percentages in different mixture samples. More specifically

$$\text{mAD} = \frac{1}{p \times q} \sum_{j=1}^{p} \sum_{k=1}^{q} |r_{jk}|.$$

2) Root mean squared distance (RMSD): This measure is one of the most commonly used

Table 2 Summary Statistics of Each Data Set

Dataset	# features	# samples	# references
BreastBlood	54675	9	2
CellLines	54675	12	4
LiverBrainLung	31099	33	3
PERT_Cultured	22215	2	11
PERT_Uncultured	22215	4	11
RatBrain	31099	10	4
Retina	22347	24	2

distance functions in the literature. It is formally defined as

$$mAD = \sqrt{\frac{1}{p \times q} \sum_{j=1}^{p} \sum_{k=1}^{q} r_{jk}^2}.$$

3) Pearson's correlation distance: Pearson's correlation measures the linear dependence between estimated and actual percentages. Let us vectorize percentage matrices as $\boldsymbol{p} = \mathbf{vec}(\mathbf{P})$ and $\hat{\boldsymbol{p}} = \mathbf{vec}(\hat{\mathbf{P}})$. Using this notation, the correlation between these two vectors is defined as

$$\rho_{\boldsymbol{p},\hat{\boldsymbol{p}}} = \frac{\mathbf{cov}(\boldsymbol{p}, \hat{\boldsymbol{p}})}{\sigma(\boldsymbol{p})\sigma(\hat{\boldsymbol{p}})} \tag{13}$$

where **cov** and σ correspond to covariance and standard variation of vectors, respectively. Finally, we define the correlation distance measure as $R^2D = 1 - \rho_{\boldsymbol{p},\hat{\boldsymbol{p}}}$.

C. Implementation

All codes and experiments have been implemented in MATLAB. To implement different formulations of the deconvolution problem, we used CVX, a package for specifying and solving convex programs [42], [43]. We used MOSEK with CVX, which is a high-performance solver for large-scale linear and quadratic programs [44]. All codes and data sets are freely available at https://github.com/shmohammadi86/DeconvolutionReview.

D. Effect of Loss Function and Constraint Enforcement on Deconvolution Performance

We perform a systematic evaluation of the four different loss functions introduced in Section II-C1 as well as implicit and explicit enforcement of nonnegativity (NN) and sum-to-one (STO) constraints over the concentration matrix ($\hat{\mathbf{C}}$) on the overall performance of deconvolution methods for each data set. There are 16 configurations of loss functions/constraints for each test case. Additionally, for Huber and Hinge loss functions, where M and ϵ are unknown, we perform a grid search with 15 values in multiples of 10 spanning the range $\{10^{-7}, \ldots, 10^{7}\}$ to find the best values for these parameters. To evaluate an upper bound on the "potential" performance of these two loss functions, we use the true concentrations in each sample $\mathbf{c}$ to evaluate each parameter choice. In practical applications, the RMSD of residual error between $\boldsymbol{m}$ and $\mathbf{G}\hat{\boldsymbol{c}}$ is often used to select the optimal parameter. This is not always in agreement with the choice made based on known $\boldsymbol{c}$.

For each test data set, we compute the three evaluation measures defined in Section III-B. Additionally, for each of these measures, we compute an empirical p-value by sampling random concentrations from a Uniform distribution and enforcing NN and STO constraints on the resulting random sample. In our study, we sampled 10 000 concentrations for each data set/measure, which results in a lower bound of 10^4 on the estimated p-values. Fig. 2 presents the time each loss function takes to compute per sample, averaged over all constraint combinations. The actual times taken for Huber and Hinge losses are roughly 15 times those reported here, which is the number of experiments performed to find the optimal parameters for these loss functions. From these results, $\mathcal{L}_2$ can be observed to have the fastest computation time, whereas $\mathcal{L}_{\text{Huber}}$ is the slowest.

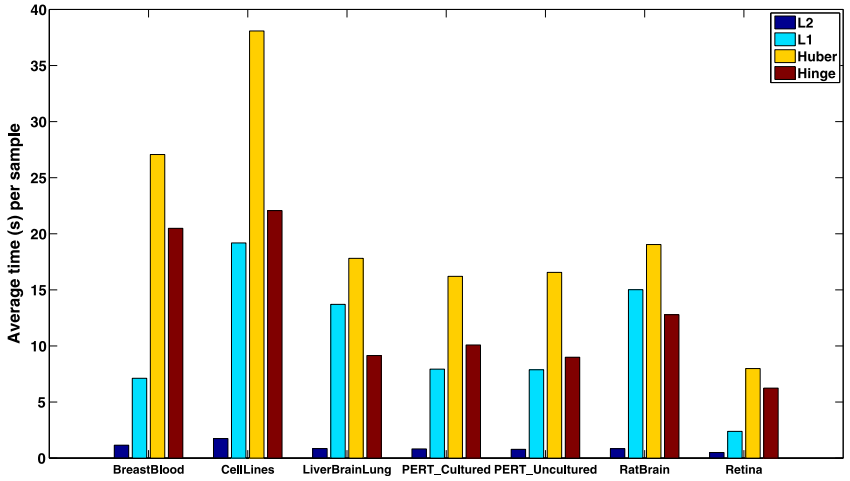


Fig. 2. *Average computational time for each loss function in different data sets.*

Measures $\mathcal{L}_1$ and $\mathcal{L}_{\text{Hinge}}$ fit between these two extremes, with $\mathcal{L}_1$ being faster the majority of times. We can directly compare these computation times, because we formulate all methods within the same framework; thus, differences in implementations do not impact direct comparisons.

Computation time, while important, is not the critical measure in our evaluation. The true performance of a configuration (selection of loss function and constraints) is measured by its estimation error. To rank different configurations, we first assess the agreement among different measures. To this end, we evaluate each data set as follows: for each experiment, we compute mAD, RMSD, and R^2D independently. Then, we use *Kendall* rank correlation, a nonparametric hypothesis test for statistical dependence between two random variables, between each pair of measures and compute a log-transformed *p*-value for each correlation. Fig. 3 shows the agreement among these measures across different data sets. Overall, RMSD and mAD measures show higher consistency compared to R^2D measure. However, the mAD measure is easier to interpret as a measure of percentage loss for each configuration. Consequently, we choose this measure for our evaluation in this study.

Using mAD as the measure of performance, we evaluate each configuration over each data set and sort the results. Fig. 4 shows various combinations for each data set. The RatBrain, LiverBrainLung, BreastBlood, and CellLines data sets achieve high performance. Among these data sets, RatBrain, LiverBrainLung, and BreastBlood had the $\mathcal{L}_2$ loss function as the best configuration, with the CellLines data set being less sensitive to the choice of the loss function. Another surprising observation is that for the majority of configurations, enforcing the sum-to-one constraint worsens the results. We investigate this issue in greater depth in Section III-E.

For Retina and both PERT data sets, the overall performance is worse than the other data sets. In the case of PERT, this is expected since the flow-sorted proportions are used as an estimate of cell-type proportions. Furthermore, the reference profiles come from a different study and therefore have greater difference with the true cell types in the mixture. However, the Retina data set exhibits unusually low performance, which may be attributed to multiple factors. As an initial investigation, we performed a quality control (QC) over different samples to see if errors are similarly distributed across samples. Fig. 5 presents per-sample error, measured by mAD, with median and median absolute deviation (MAD) marked accordingly. Interestingly, for the fourth, sixth, and eighth mixtures, the third replicate has much higher error than the rest. In the expression matrix, we observed a lower correlation between these replicates and the other two replicates in the batch. Additionally, for the seventh mixture, all three replicates show high error rates. We expand on these results in later sections to identify additional reasons that contribute to the low deconvolution performance of the Retina data set.

Finally, we note that in all test cases the performance of $\mathcal{L}_1$, $\mathcal{L}_{\text{Huber}}$, and $\mathcal{L}_{\text{Hinge}}$ are comparable, whereas $\mathcal{L}_{\text{Huber}}$ and $\mathcal{L}_{\text{Hinge}}$ needed an additional step of parameter tuning. Consequently, we consider only $\mathcal{L}_1$ as a representative of this "robust" group of loss functions in the rest of our study.

E. Agreement of Gene Expressions With Sum-to-One (STO) Constraint

Considering the lower performance of configurations that explicitly enforce STO constraints, we aim to investigate whether features (genes) in each data set respect this constraint. Under the STO and NN constraints, we

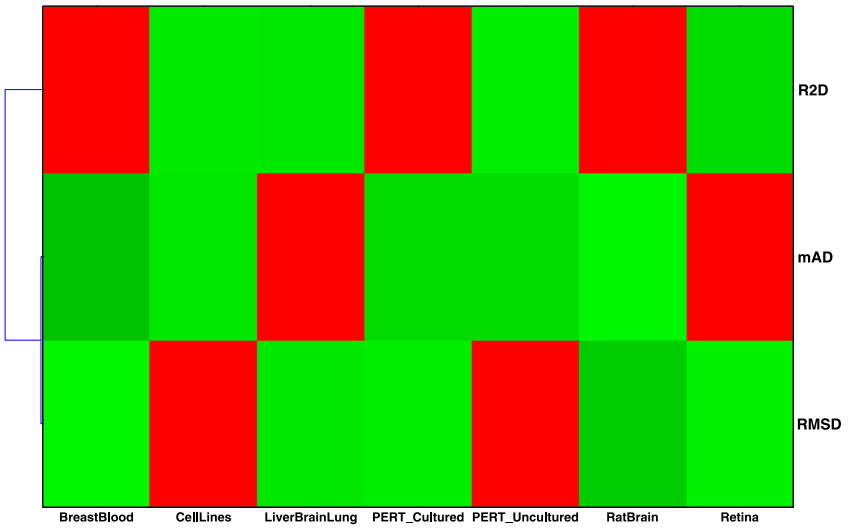


Fig. 3. ***Agreement among different evaluation measures across different data sets.***

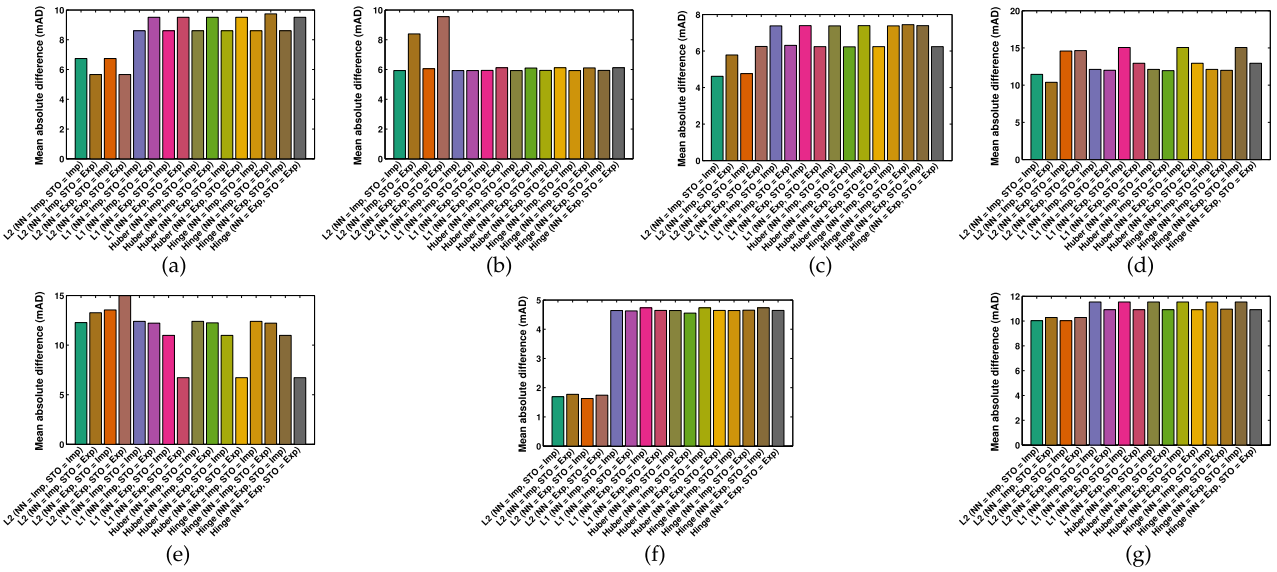


Fig. 4. ***Overall performance of different loss function/constraints combinations over all data sets (the lower, the better). (a) BreastBlood. (b) CellLines. (c) LiverBrainLung. (d) PERT_Cultured. (e) PERT_Uncultured. (f) RatBrain. (g) Retina.***

use simple bounds for identifying violating features for which there is no combination of concentration values that can satisfy both *STO* and *NN*. Let $\boldsymbol{m}(i)$ be the expression value of the ith gene in the given mixture, and $\mathbf{G}(i,1),\ldots,\mathbf{G}(i,q)$ be the corresponding expressions in different reference cell types. Let $\mathbf{G}_{\min}(i) = \mathbf{min}\{\mathbf{G}(i,1),\ldots,\mathbf{G}(i,q)\}$ and $\mathbf{G}_{\max}(i) = \mathbf{max}\{\mathbf{G}(i,1),\ldots,\mathbf{G}(i,q)\}$. Given that all concentrations are bound between $0 \leq \boldsymbol{c}(k) \leq 1$; $\forall 1 \leq k \leq k$, the minimum and maximum values that an estimated mixture value for the ith gene can attain are $\mathbf{G}_{\min}(i)$ and $\mathbf{G}_{\max}(i)$, respectively (by setting $\boldsymbol{c}(k) = 1$ for min/max value, and 0 everywhere else). Using this notation, we can identify features that violate *STO* as follows:

$$\boldsymbol{m}(i) \leq \mathbf{G}_{\min}(i) \quad \forall 1 \leq i \leq n \quad \{\text{Violating reference}\}$$
$$\mathbf{G}_{\max}(i) \leq \boldsymbol{m}(i) \quad \forall 1 \leq i \leq n \quad \{\text{Violating mixture}\}.$$

The first condition holds because expression values in reference profiles are so large that we need the sum of concentrations to be lower than one to be able to match

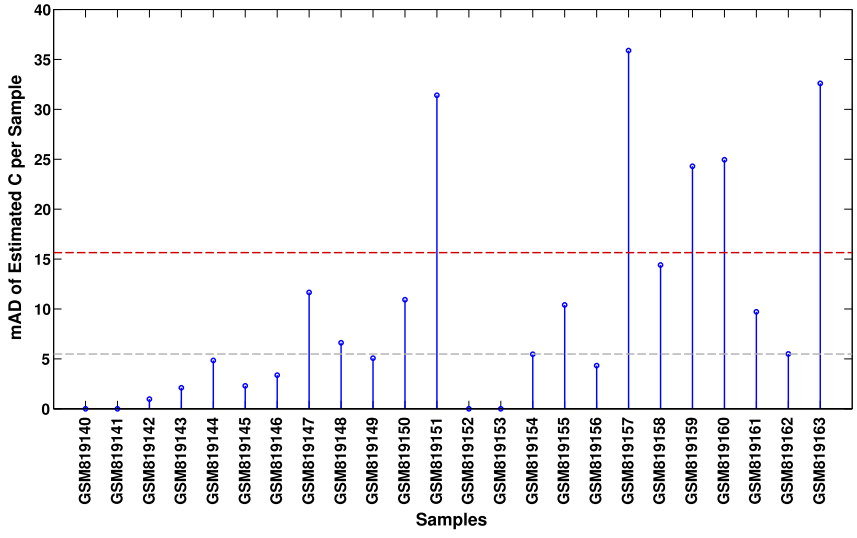


Fig. 5. ***Sample-based error of the Retina data set, based on $\mathcal{L}_2$ with explicit NN and STO.***

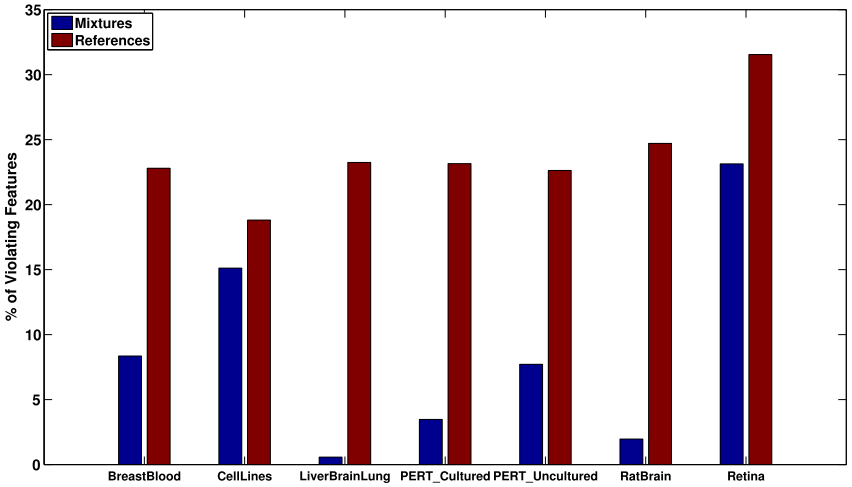


Fig. 6. ***Percent of features in each data set that violate the STO constraint.***

the corresponding gene expression in the mixture. The second condition holds in cases wherein the expression of a gene in the mixture is so high that we need the sum of concentrations to be greater than one to be able to match it. In other words, for feature i, these constraints identify extreme expression values in reference profiles and mixture samples, respectively. Using these conditions, we compute the total number of features violating *STO* condition in each data set.

Fig. 6 presents violating features in mixtures and reference profiles, averaged over all mixture samples in each data set. We normalize and report the percent of features to account for differences in the total number of features in each data set. We first observe that for the majority of data sets, except Retina and BreastBlood, the percent of violating features is much smaller than violating features in reference profiles. These two data sets also have the highest number of violating features in their reference profiles, summing to a total of approximately 60% of all features. This observation is likely due to the normalization used in preprocessing microarray profiles. Specifically, one must not only normalize **M** and **G** independently but also with respect to each other. We suggest using control genes that are expressed in all cell types with low variation to normalize expression profiles. A recent study aimed to identify subsets of housekeeping genes in human tissues that respect these conditions [45]. Another choice is using ribosomal proteins, the basic building blocks of the cellular translation machinery, which are expressed in a wide range of species. The remove unwanted variation (RUV) [46] method is developed not only to remove batch effects from microarray and RNASeq expression profiles but also to normalize them using control genes. A simple extension of this method can be adopted to solve the normalization difference between mixtures and references.

Next, we evaluate how filtering these features affects deconvolution performance of each data set. For each case, we run deconvolution using all configurations and report the change (delta mAD) independently. Fig. 7 presents changes in the mAD estimation error after removing violating features in both $\boldsymbol{m}$ and **G** before performing deconvolution. Similar to previous experiments, the Retina data set exhibits widely different behavior than the rest of the data sets. Removing this data set from further consideration, we find that the overall performance over all data sets improves, with the exception of the RatBrain data set. In the case of the RatBrain data set, we hypothesize that the initially superior performance can be attributed to highly expressed features. These outliers that happens to agree with the true solution result in over-fitting. Finally, we note a correlation between observed enhancements and the level of violation of features in $\boldsymbol{m}$. Consistent with this observation, we obtain similar results when we filter violating features only from mixtures, not from reference profiles.

F. Range Filtering: Finding an Optimal Threshold

Different upper/lower bounds have been proposed in the literature to prefilter expression values prior to deconvolution. For example, Gong *et al.* [23] suggest an effective range of [0.5, 5000], whereas Ahn *et al.* [47] observe an optimal range of $[2^4–2^{14}]$. To facilitate the choice of expression bounds, we seek a systematic way to identify an optimal range for different data sets. Kawaji *et al.* [48] recently report on an experiment to assess whether gene expression is quantified linearly in mixtures. To this end, they mix two cell types (THP-1 and HeLa cell lines) and determine whether experimentally measured expressions match with the computationally simulated data sets. They observe that expression values for microarray measurements are skewed for the lowly expressed

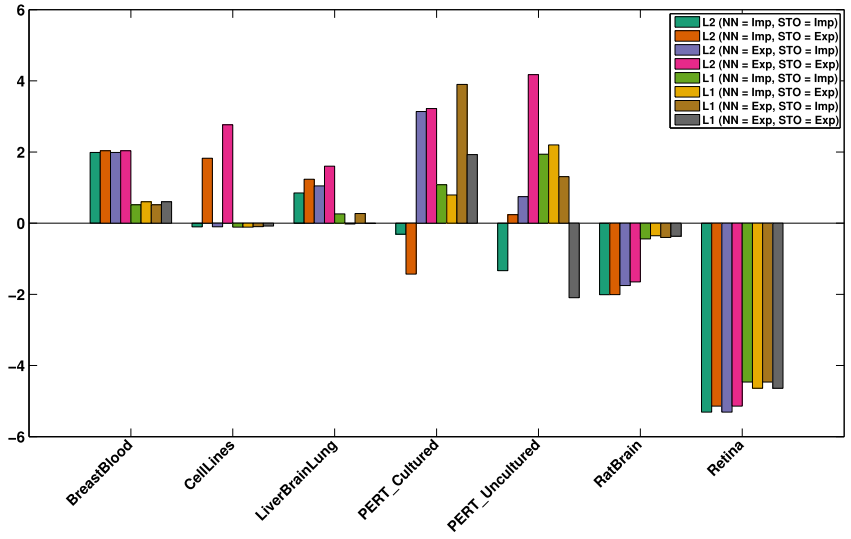


Fig. 7. ***Performance of deconvolution methods after removing violating features.***

genes (approximately < 10). This allows us to choose the lower bound based on experimental evidence. In our study, we search for the optimal bounds over a log_2-linear space; thus, we set a threshold of 2^3 on the minimum expression values, which is closest to the bound proposed by Kawaji *et al.* [48].

Choosing an upper bound on the expression values is a harder problem, since it relates to enhancing the performance of deconvolution methods by removing outliers. Additionally, there is a known relationship between the mean expression value and its variance [49], which makes these outliers noisier than the rest of the features. This becomes even more important when dealing with purified cell types that come from different labs, since highly expressed time/micro-environment dependent genes would be significantly different from the ones in the mixture [22]. A simple argument is to filter genes that the range of expression values in *Affymetrix* microarray technology is bounded by 2^{16} (due to initial normalization and image processing steps). Measurements close to this bound are not reliable as they might be saturated and inaccurate. However, practical bounds used in previous studies are far from these extreme values. To examine the overall distribution of expression values, we analyze different data sets independently. For each data set, we separately analyze mixture samples and reference profiles, encoded by matrices **M** and **G**, respectively. For each of these matrices, we vectorize the expression values and perform kernel smoothing using the Gaussian kernel to estimate the probability density function.

Fig. 8(a) and (b) shows the distribution of log-transformed expression values for mixtures and reference profiles, respectively. These expression values are greater than our lower bound of 2^3. In agreement with our previous results, we observe an unusually skewed distribution for the Retina data set, which in turn contributes to its lower performance compared to other ideal mixtures. Additionally, we observe that approximately 80% of the features in this data set are smaller than 2^3, which are filtered and not shown in the distribution plot. For the rest of the data sets in both mixtures and references, we observe a bell-shaped distribution with most of the features captured up to an upper bound of 2^8–2^{10}. Another exception to this pattern is the CellLines data set, which has a heavier tail than other data sets, especially in its reference profile.

Next, we systematically evaluate the effect of range filtering by analyzing upper bounds increasing in factors of 10 in the range $\{2^5, \ldots, 2^{16}\}$. In each case, we remove all features for which at least one of the reference profiles or mixture samples has a value exceeding this upper bound. Fig. 9 illustrates the percent of features that are retained, as we increase the upper bound. As mentioned earlier, approximately 80% of the features in the Retina data set are lower than 2^3, which is evident from the maximum percent of features left to be bounded by 20% in this figure. Additionally, consistent with our previous observation over expression densities, more that 80% of the features are covered between 2^8–2^{10}, except for the CellLine data set.

Finally, we perform deconvolution using the remaining features given each upper bound. The results are mixed, but a common trend is that removing highly expressed genes decreases performance of ideal mixtures with known concentrations, while enhancing the performance of PERT data sets. Fig. 10(a) and (b) shows the changes in mAD error, compared to unfiltered deconvolution, for the PERT data set. In each case, we observe

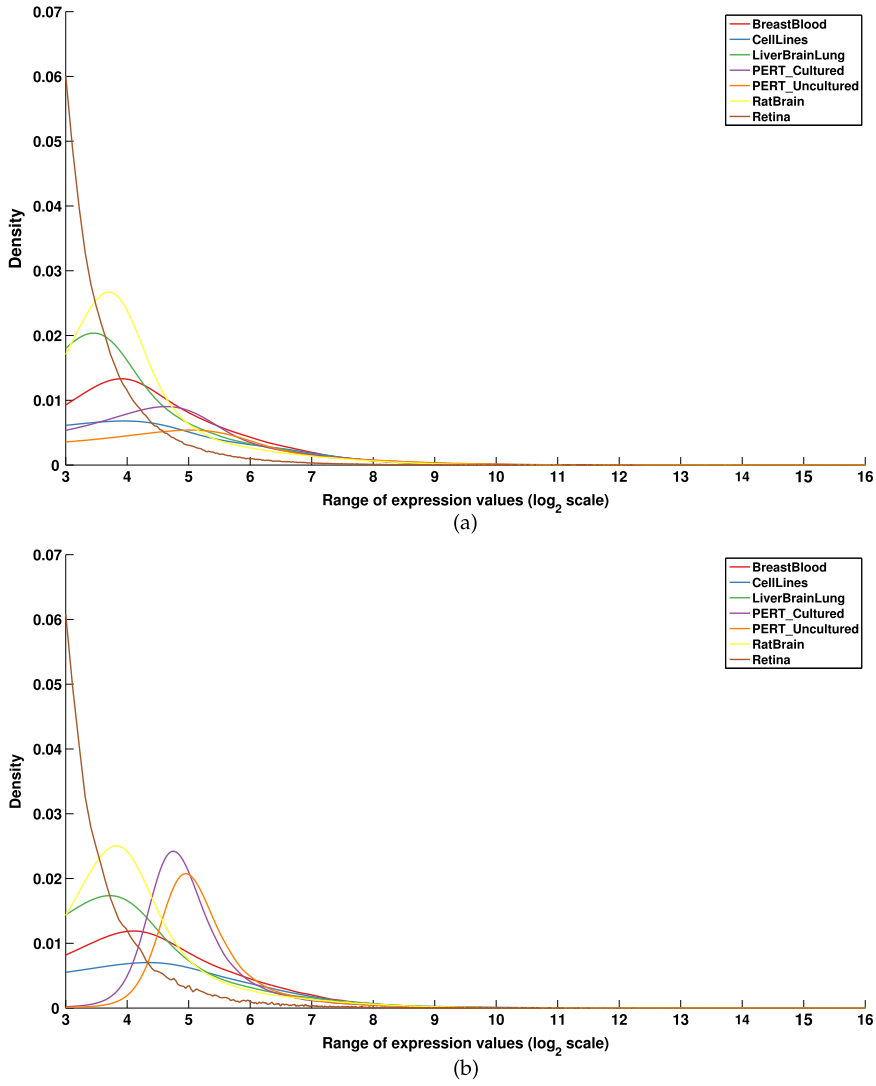


Fig. 8. ***Distribution of expression values. (a) Mixtures. (b) Reference profiles.***

improvements up to 7 and 8 percent, respectively. The red and green points on the diagram show the significance of deconvolution. Interestingly, while both methods show similar improvements, all data points for cultured PERT seem to be insignificant, whereas uncultured PERT shows significance for the majority of data points. This is due to the weakness of our random model, which is dependent on the number of samples and is not comparable across data sets. Uncultured PERT has twice as many samples as cultured PERT, which makes it less likely to have any random samples achieving as good an mAD as the observed estimation error. This dependency on the number of samples can be addressed by defining sample-based p-values. Another observation is that for the uncultured data set, all measures have been improved, except $\mathcal{L}_1$ with explicit NN and STO constraints. On the other hand, for the cultured data set, both $\mathcal{L}_1$ and $\mathcal{L}_2$ with the explicit NN constraint perform well, whereas implicitly enforcing NN deteriorates their performance. Cultured and uncultured data sets have their peak at 2^{10} and 2^{12}, respectively.

For the rest of the data sets, range filtering decreased performance in a majority of cases, except the Retina data set, which had an improved performance at 2^6, with the best result achieved with $\mathcal{L}_1$ with both explicit NN and STO enforcement. This changed the best observed performance of this datasest, measured as mAD, to be close to 7. These mixed results make it harder to choose a threshold for the upper bound, so we average results over all data sets to find a balance between improvements in PERT and overall deterioration in other data sets. Fig. 11 presents the averaged mAD difference across all data sets. This suggests a "general" upper bound filter of 2^{12} to be optimal across all data sets.

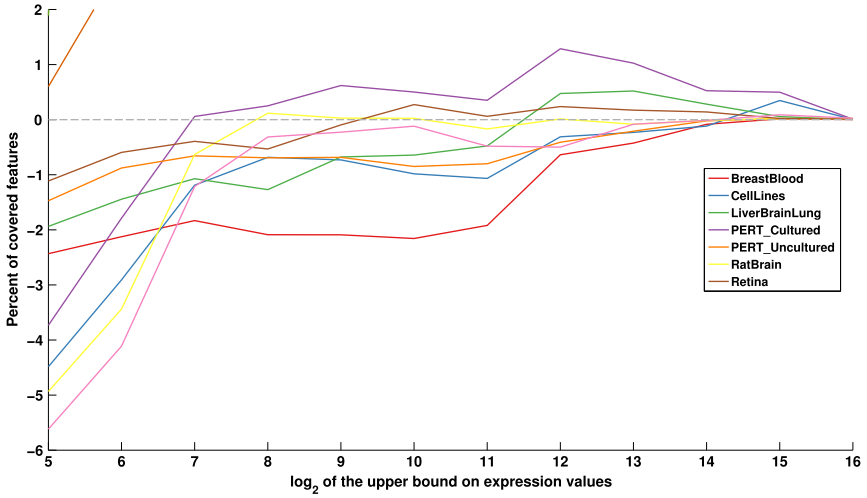
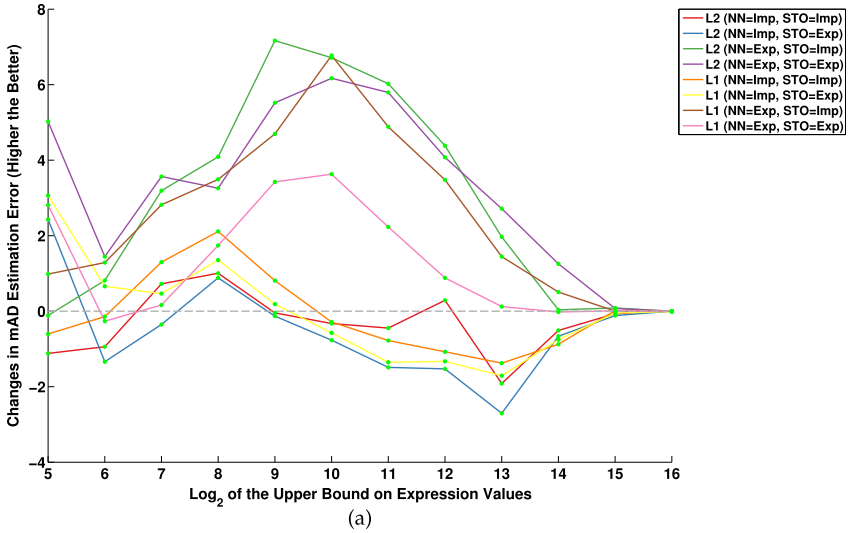
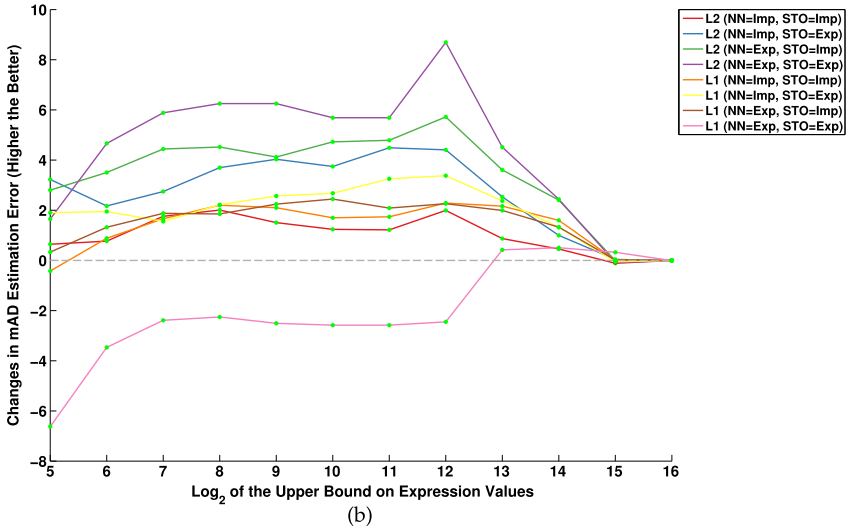


Fig. 9. ***Percent of covered features during range filtering.***



(a)



(b)

Fig. 10. ***Performance of PERT data sets during range filtering. (a) Cultured. (b) Uncultured.***

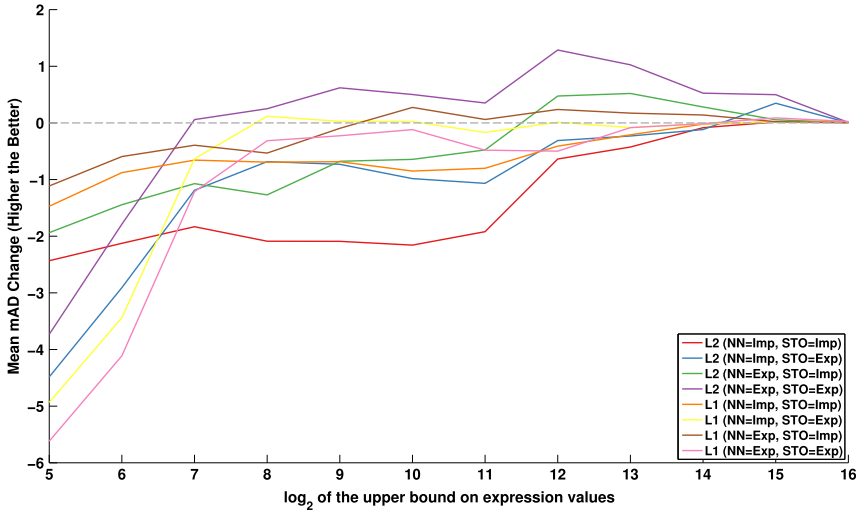


Fig. 11. ***Average performance of range filtering over all data sets.***

We use this threshold to filter all data sets and perform deconvolution on them. Fig. 12 presents the data set-specific performance of range filtering with fixed bounds, measured by changes in the *mAD* value compared to the original deconvolution. As observed from individual performance plots, range filtering is most effective in cases wherein the reference profiles differ significantly from the true cell types in the mixture, such as the case with the PERT data sets. In ideal mixtures, since cell types are measured and mixed at the same time/laboratory, this distinction is negligible. In these cases, highly expressed genes in mixtures and references coincide with each other and provide additional clues for the regression. Consequently, removing these highly expressed genes often degrades the performance of deconvolution methods. This generalization of the upper bound threshold, however, should be adopted with care, since each data set exhibits different behavior in response to range filtering. Ideally, one must filter each data set individually based on the distribution of expression values. Furthermore, in practical applications, gold standards are not available to aid in the choice of cutoff threshold.

We now introduce a new method that adaptively identifies an effective range for each data set. Fig. 13 illustrates the log_2 normalized value of maximal expression for each gene in matrices **M** and **G**, sorted in ascending order. In all cases, intermediate values exhibit a gradual increase, whereas the top and bottom elements in the sorted list show a steep change in their expression. We aim to identify the critical points corresponding

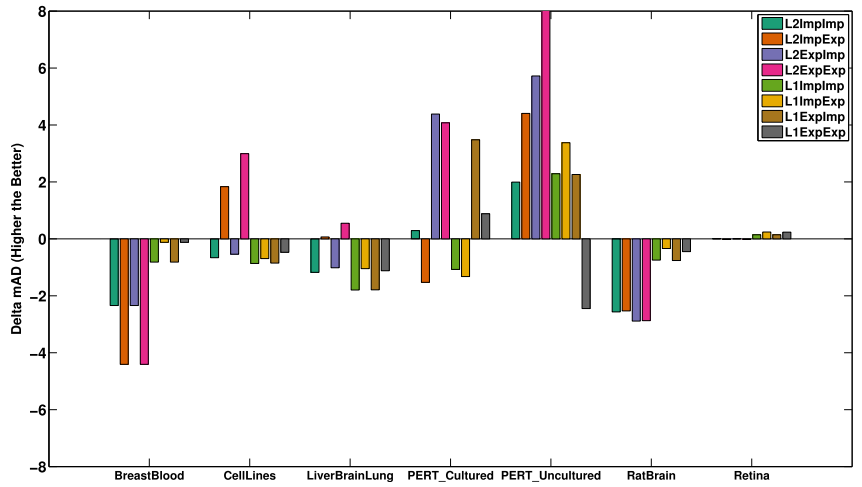


Fig. 12. ***Data set-specific changes in the performance of deconvolution methods after filtering expression ranges to fit within [2^3–2^{12}].***

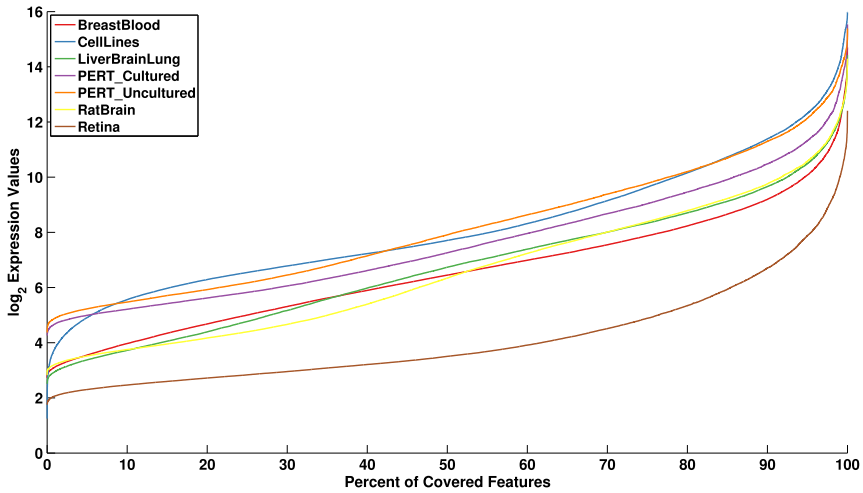


Fig. 13. ***Sorted $\log_2$-transformed gene expressions in different data sets.***

to these sudden changes in the expression values for each data set. To this end, we select the middle point as a point of reference and analyze the upper and lower half, independently. For each half, we find the point on the curve that has the longest distance from the line connecting the first (last) element to the middle element. Application of this process over the CellTypes data set is visualized in Fig. 14. Green points in this figure correspond to the critical points, which are used to define the lower and upper bound for the expression values of this data set.

We use this technique to identify adaptive ranges for each data set prior to deconvolution. Table 3 summarizes the identified critical points for each data set. Fig. 15 presents the data set-specific performance of each method after adaptive range filtering. While in most cases the results for fixed and adaptive range filtering are compatible, and in some cases adaptive filtering gives better results, the most notable difference is the degraded performance of LiverBrainLung and, to some extent, RatBrain data sets. To further investigate this observation, we examine individual experiments for these data sets for fixed thresholds. Fig. 16 illustrates individual plots for each data set. The common trend here is that in both cases range filtering, in general, degrades the performance of deconvolution methods for all configurations. In other words, extreme values in these data sets are actually helpful in guiding the regression, and any filtering negatively impacts performance. This suggests that range filtering, in general, is not always helpful in

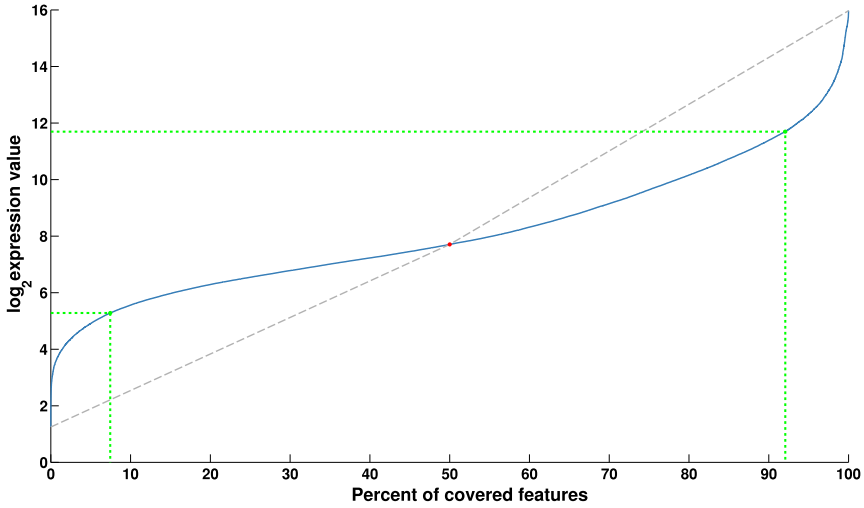


Fig. 14. ***Example of adaptive filtering over the CellLines data set.***

Table 3 Summary of Adaptive Ranges for Each Data Set

	LowerBound	UpperBound
BreastBlood	4.2842	9.4314
CellLines	5.2814	11.6942
LiverBrainLung	3.3245	9.9324
PERT_Cultured	4.9416	10.9224
PERT_Uncultured	5.1674	11.5042
RatBrain	3.3726	9.9698
Retina	2.4063	6.7499

enhancing the deconvolution performance. In some cases, for example, the ideal data sets, such as LiverBrainLung, RatBrain, and BreastBlood, it can be counterproductive.

G. Selection of Marker Genes: The Good, the Bad, and the Ugly

Selecting marker genes that uniquely identify a certain tissue or cell type prior to deconvolution can help in improving the conditioning of matrix **G**, thus improving its discriminating power and stability of results and decreasing the overall computation time. A key challenge in identifying "marker" genes is the choice of method that is used to assess selectivity of genes. Various parametric and nonparametric methods have been proposed in literature to identify differentially expressed genes between two groups [50], [51] or between a group and other groups [52]. Furthermore, different methods have been developed in parallel to identify tissue-specific and tissue-selective genes that are unique markers with high specificity to their host tissue/cell type [53]–[56]. While choosing/developing accurate methods for identifying reliable markers is an important challenge, an in-depth discussion of the matter is beyond the scope of this paper. Instead, we adopt two methods used in the literature. Abbas *et al.* [21] present a framework for choosing genes based on their overall differential expression. For each gene, they use a t-test to compare the cell type with the highest expression with the second and third highest expressing cell type. Then, they sort all genes and construct a sequence of basis matrices with increasing sizes. Finally, they use condition number to identify an "optimal" cut among top-ranked genes that minimizes the condition number. Newman *et al.* [15] propose a modification to the method of Abbas *et al.*, in which genes are not sorted based on their overall differential expression but according to their tissue-specific expression when compared to all other cell types. After prefiltering differentially expressed genes, they sort genes based on their expression fold ratio and use a similar cutoff that optimizes the condition number. Note that the former method increases the size of the basis matrix by one at each step, whereas the latter method increases it by q (number of cell types). The method of Newman *et al.* has the benefit that it chooses a similar number of markers per cell type, which is useful in cases wherein one of the references has a significantly higher number of markers.

We implement both methods and assess their performance over the data sets. We observe slightly better performance with the second method and use it for the rest of our experiments. Due to unexpected behavior of the Retina data set and a low number of significant markers in all our trials, we eliminate this data set from further study. In identifying differentially expressed genes, a key parameter is the q-value cutoff to report significant features. The distribution of corrected p-values exhibits high similarity among ideal mixtures, while differing significantly in CellLines mixtures and both PERT data sets. We find the range of 10^3–10^5 to be an optimal balance between these two cases and perform experiments to test different cutoff values. Fig. 17 shows changes in the mAD measure after applying marker detection, using a q-value

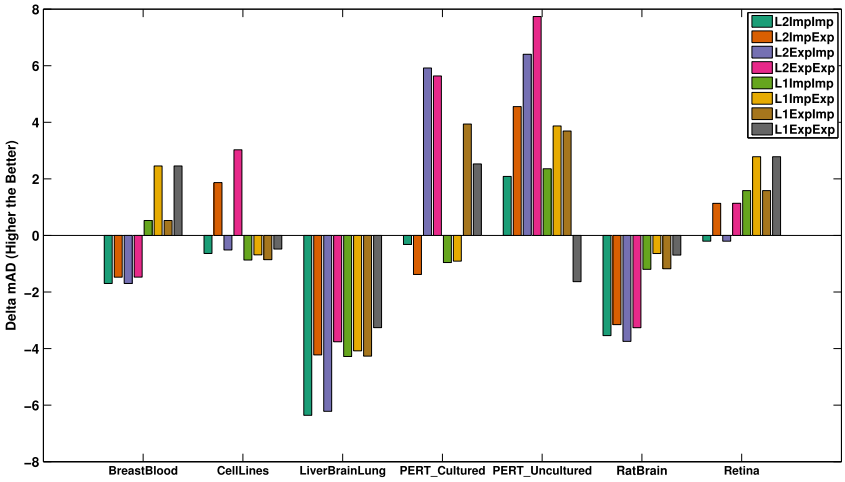


Fig. 15. ***Data set-specific changes in the performance of deconvolution methods after adaptive range filtering.***

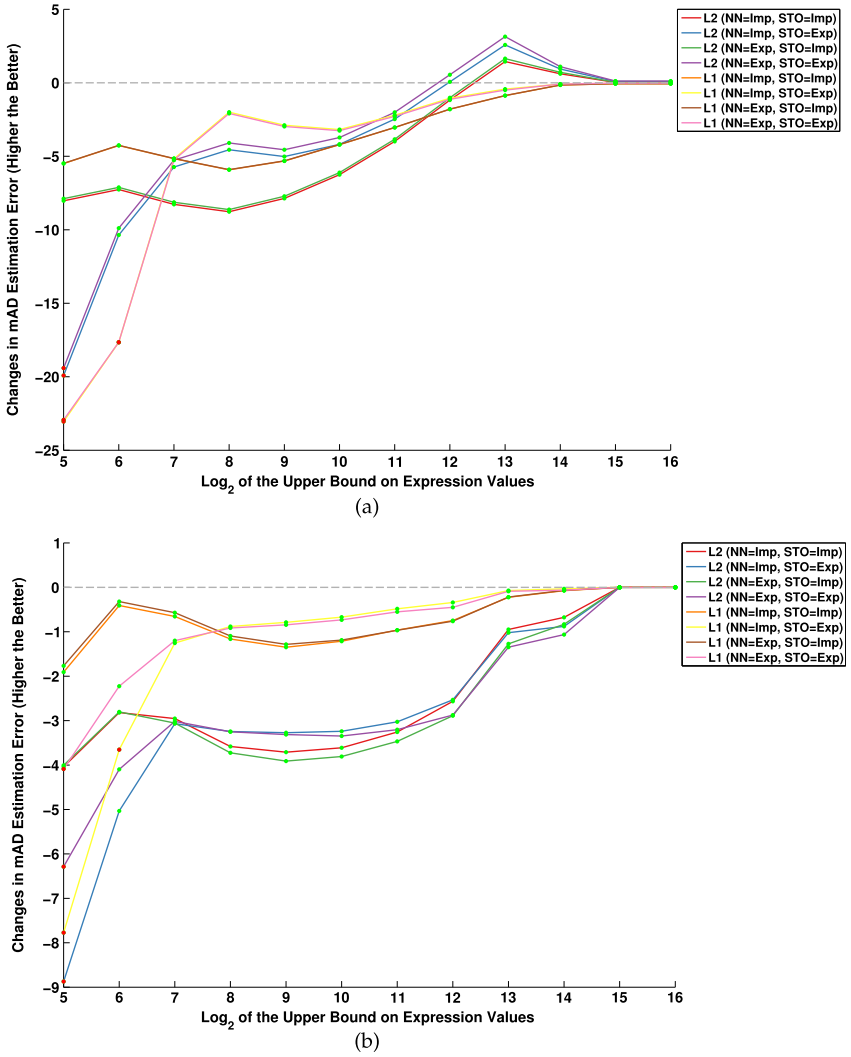


Fig. 16. ***Individual performance plots for range filtering in data sets in which range filtering exhibits a negative effect on the deconvolution. (a) LiverBrainLung. (b) RatBrain.***

cutoff of 10^3, which resulted in the best overall performance in our study. We observe that the PERT_Uncultured and LiverBrainLung data sets have the highest gain across the majority of configurations, while BreastBlood and RatBrain exhibit an improvement in experiments with $\mathcal{L}_1$ while their $\mathcal{L}_2$ performance is greatly decreased. Finally, for the PERT_Cultured and CellLines data sets, we observe an overall decrease in performance in almost all configurations.

Next, we note that the internal sorting based on fold-ratio intrinsically prioritizes highly expressed genes and is susceptible to noisy outliers. To test this hypothesis, we perform a range selection using a global upper bound of 10^{12} prior to the marker selection method and examine if this combination can enhance our previous results. We find the q-value threshold of 10^5 to be the better choice in this case. Fig. 18 shows changes in performance of different methods when we prefilter expression ranges prior to marker selection. The most notable change is that both the PERT_Cultured and the CellLines, which were among the data sets with the lowest performance in the previous experiment, are now among the best-performing data sets in terms of overall mAD enhancement. We still observe a higher negative impact on $\mathcal{L}_2$ in this case, but the overall amplitude of the effect has been dampened in both BreastBlood and RatBrain data sets.

We note that there is no prior knowledge as to the "proper" choice of the marker selection method in the literature and that their effect on the deconvolution performance is unclear. An in-depth comparison of marker detection methods can benefit future developments in this field. An ideal marker should serve two purpose: 1) be highly informative of the cell type in which it is expressed; and 2) shows low variance due to

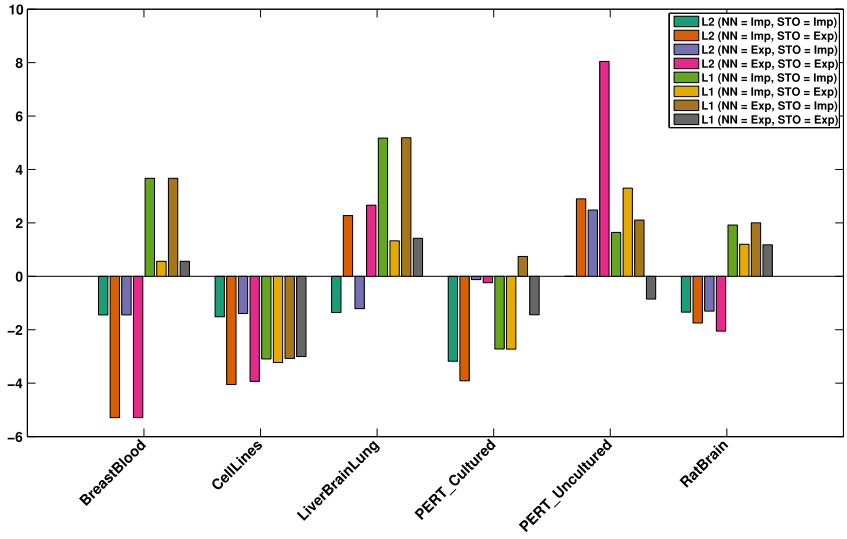


Fig. 17. ***Effect of marker selection on the performance of deconvolution methods.***

spatiotemporal changes in the environment (changes in time or microenvironment). Fig. 19 shows a high-level classification of genes. An ideal marker is an invariant, cell-type-specific gene, marked with green in the diagram. On the other hand, variant genes, both universally expressed and tissue-specific, are not good candidates, especially in cases in which references are adopted from a different study. These genes, however, comprise ideal subsets of genes that should be updated in full deconvolution while updating matrix **G**, since their expression in the reference profile may differ significantly from the true cell types in the mixture. It is worth mentioning that the proper ordering to identify best markers is to first identify tissue-specific genes and then prune them based on their variability. Otherwise, when selecting invariant genes, we may select many housekeeping genes, since their expression is known to be more uniform compared to tissue-specific genes.

Another observation relates to the case in which groups of profiles of cell types have high similarity within the group but are significantly distant from the rest. This makes identifying marker genes more challenging for these groups of cell types. An instance of this problem is when we consider markers in the PERT data

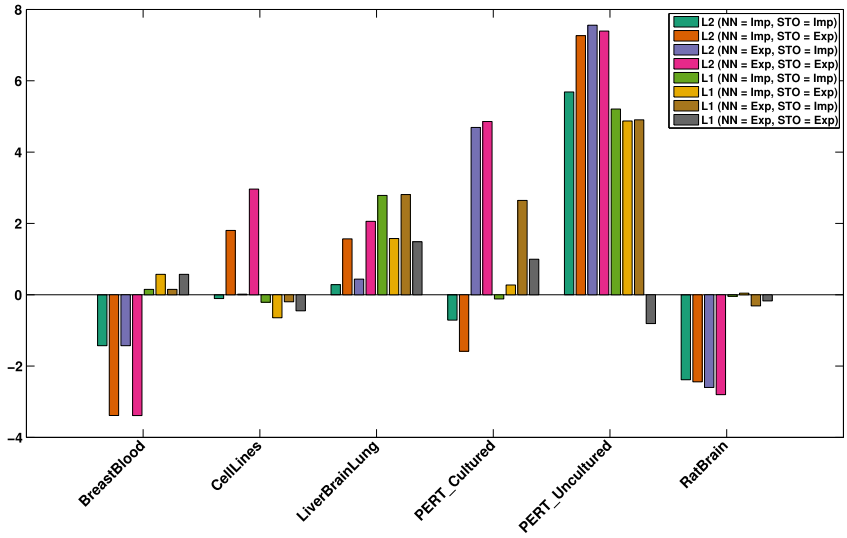


Fig. 18. ***Effect of marker selection, after range filtering, on the performance of deconvolution methods.***

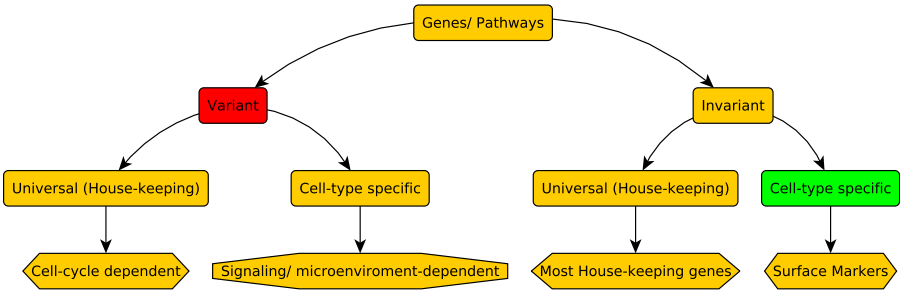


Fig. 19. ***High-level classification of genes.***

sets. In this case, erythrocytes have a much larger number of distinguishing markers compared to other references. This phenomenon is primarily attributed to the underlying similarity between undifferentiated cell types in the PERT data sets, and their distance from the fully differentiated red blood cells. In these cases, it is beneficial to summarize each group of similar tissues using a "representative profile" for the whole group, and to use a hierarchical structure to recursively identify markers at different levels of resolution [56].

Finally, we examine the common choice of condition number as the optimal choice to select the number of markers. First, unlike the "U" shape plot reported in previous studies in which condition number initially decreases to an optimal point and then starts increasing, we observe variable behavior in the condition number plot, both for Newman *et al.* and Abbas *et al.* methods. This makes the generalization of condition number as a measure applicable to all data sets infeasible. Additionally, we note that the lowest condition number is achieved if $\mathbf{G}$ is fully orthogonal, that is $\mathbf{G}^{\mathrm{T}}\mathbf{G} = \kappa\mathbf{I}$ for any constant κ. By selecting tissue-selective markers, we can ensure that the product of columns in the resulting matrix is close to zero. However, the norm-2 of each column can still be different. We developed a method that specifically grows the basis matrix by accounting for the norm equality across columns. We find that in all cases our basis matrix has a lower condition number than both the Newman *et al.* and Abbas *et al.* methods, but it did not always improve the overall performance of deconvolution methods using different loss functions. Further study on the optimal choice of number of markers is another key question that needs further investigation.

H. To Regularize or Not to Regularize

We now evaluate the impact of regularization on the performance of different deconvolution methods. To isolate the effect of the regularizer from prior filtering/feature selection steps, we apply regularization on the original data sets. The $\mathcal{R}_1$ regularizer is typically applied in cases in which the solution space is large; that is, the total number of available reference cell types is a superset of the true cell types in the mixture. This type of regularization acts as a "selector" to choose the most relevant cell types and zero-out the coefficients for the rest of the cell types. This has the effect of enforcing a sparsity pattern. Data sets used in this study are all controlled benchmarks in which references are hand-picked to match the ones in the mixture; thus, sparsifying the solution does not add value to the deconvolution process. On the other hand, an $\mathcal{R}_2$ regularizer, also known as Tikhonov regularization, is most commonly used when the problem is ill-posed. This is the case, for example, when the underlying cell types are highly correlated with each other, which introduces dependency among columns of the basis matrix. To quantify the impact of this type of regularization on the performance of deconvolution methods, we perform an experiment similar to the one in Section III-D with an added $\mathcal{R}_2$ regularizer. In this experiment, we use $\mathcal{L}_1$ and $\mathcal{L}_2$ loss functions, as we previously showed that the performance of the other two loss functions is similar to $\mathcal{L}_1$. Instead of using Ridge regression introduced in Section II-D2, we implement an equivalent formulation, $\|\boldsymbol{m} - \mathbf{G}\boldsymbol{c}\|_2 + \lambda\|\boldsymbol{c}\|_1$, which traces the same path but has higher numerical accuracy. To identify the optimal value of the λ parameter that balances the relative importance of solution fit versus regularization, we search over the range of $\{10^{-7}, \ldots, 10^{7}\}$. It is notable here that when λ is close to zero, the solution is identical to the one without regularization, whereas when $\lambda \to \infty$ the deconvolution process is guided by only the solution size. Similar to the range filtering step in Section III-F, we use the minimum *mAD* error to choose the optimal value of λ.

Fig. 20 presents changes in mAD error, compared to original errors, after regularizing loss functions with the $\mathcal{R}_2$ regularizer. From these observations, it appears that PERT_Cultured has the most gain due to regularization, whereas for PERT_Uncultured, the changes are smaller. A detailed investigation, however, suggests that in the majority of cases for PERT_Cultured, the performance gain is due to over shrinkage of vector $\boldsymbol{c}$ to the case of

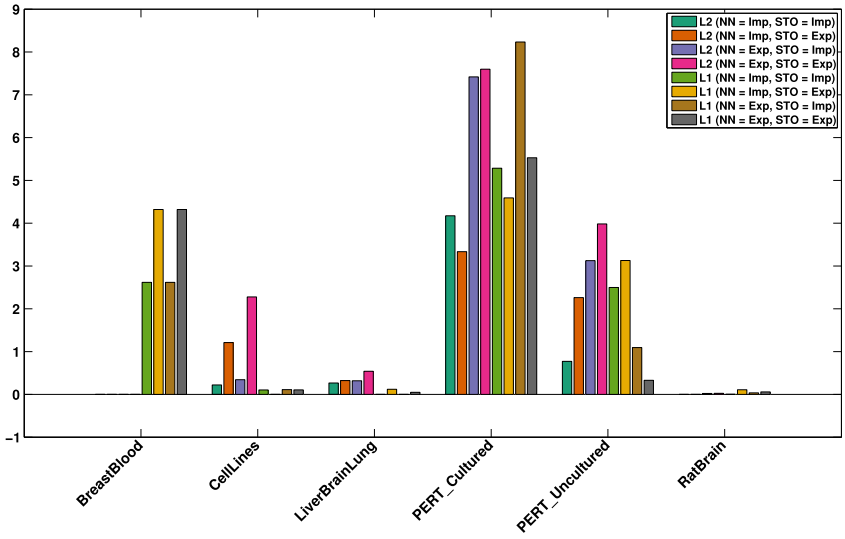


Fig. 20. ***Effect of L2 regularization on the performance of deconvolution methods.***

being almost uniform. Interestingly, the choice of uniform $\boldsymbol{c}$ has lower mAD error for this data set compared to most other results. Overall, both of the PERT data sets show significant improvements compared to the original solution, which can be attributed to the underlying similarity among hematopoietic cells. On the other hand, an unexpected observation is the performance gain over $\mathcal{L}_1$ configurations for the BreastBlood data set. This is primarily explained by the limited number of cell types (only two), combined with the similar concentrations used in all samples (only combinations of 67% and 33%).

To gain additional insight into the parameters used in each case during deconvolution, we plot the optimal λ values for each configuration in each data set. Fig. 21 summarizes the optimal values of the λ parameter. Large values indicate a beneficial effect for regularization, whereas small values are suggestive of negative impact. In all cases where the overall mAD score has been improved, their corresponding λ parameter was large. However, large values of λ do not necessarily indicate a significant impact on the final solution, as is evident in the CellLines and LiverBrainLung data sets. Finally, we

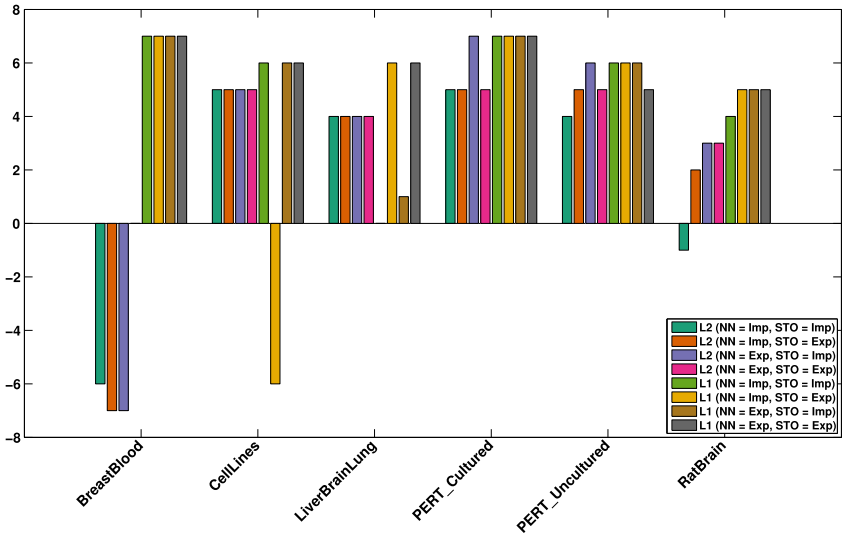


Fig. 21. ***Optimal value of λ for each data set/configuration pair.***

observe that cases wherein the value of λ is close to zero are primarily associated with the $\mathcal{L}_2$ loss function.

I. Putting Sticks in a Bundle

Having analyzed each individual aspect that affects the performance of the deconvolution process, in this section we aim to put all the pieces back together and evaluate the overall performance over each data set. We will remove the Retina data set from this study due to the observed discrepancies. For the remaining six data sets, we will assess performance of both $\mathcal{L}_1$ and $\mathcal{L}_2$ objectives, with different combinations of NN/STO enforcement, for a total of eight configurations. For each particular configuration, we use our previous results to decide about proper feature selection, i.e., whether to remove violating features and/or select marker genes. Finally, we note that our results in Section III-H, while being instructive, are not directly applicable here due to differences in the selected subset of genes. Thus, for each configuration, we reran the experiment without regularization as well as regularized the problem with $\lambda \in \{10^{-7}, \ldots, 10^{7}\}$. Table 4 summarizes the settings used to solve each particular instance. There are some general patterns to note here. First, for the RatBrain data set, we are at the lowest attainable mAD error, and neither removing violating features nor selecting markers can boost that. For this data set, mAD is significantly lower than the rest of data sets, and we argue this might be due to the existence of highly expressed genes where they just happen to align, with low variations, between reference profiles and mixtures. This, however, has the potential downfall of overfitting, in which case the best case configurations identified in this data set are not generalizable to other data sets. Next, we observe that in the majority of cases filtering violating features, on average, either decreases mAD error or at least it does not increase it, with the previously mentioned exception of RatBrain. Similarly, selecting marker genes combined with range filtering in most cases improves deconvolution results, except in RatBrain and BreastBlood. For the BreastBlood data set, we argue that the quality of selected markers might be affected because we have only two cell types, but this needs further validation. Finally, we note that either marker selection or range filtering alone performs much worse than combining them together. The final results of our experiments, before and after feature selection/regularization, are illustrated in Fig. 22. Shaded bars correspond to the original performance for each configuration, and colored bars are the final mAD errors computed. Interestingly, after proper feature selection, the results of most data sets are within similar error ranges, approximately within the range of [5–7] mAD. Furthermore, $\mathcal{L}_2$ seems to perform equally as good as $\mathcal{L}_1$, given the proper subspace of genes to perform deconvolution, if not better. This aligns well with our understanding since $\mathcal{L}_2$ has higher sensitivity compared to $\mathcal{L}_1$. Even in both PERT data sets, we observed only minor differences between these two objectives. In addition, $Loss_2$ has much more efficient solvers compared with $\mathcal{L}_1$. Thus, in general we suggest using $\mathcal{L}_2$ and only resort to $\mathcal{L}_1$ if that does not perform well. However, we can not generalize this claim to all data sets. In cases with a high level of noise and/or imperfect marker selection, $Loss_1$ will still be a better choice. Finally, unlike the traditional wisdom, we observe a data set-dependency for the effect of constraint enforcement. That is, explicitly encoding all constraints in the objective function does not always enhance the quality of final results. A deeper understanding of the reasoning behind this observation can guide one to choose the right formulation for the problem at hand.

Table 4 Best Combination of Choices for Feature Selection/Regularization for Different Data Sets

Loss Function		$\mathcal{L}_2$				$\mathcal{L}_1$			
Non-negativity		-	+	-	+	-	+	-	+
Sum-to-One		-	+	+	+	-	+	+	+
BreastBlood	Remove violating features	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	Filter markers/range	No	No	No	No	No	Yes	No	Yes
	Best lambda	0	0.001	1E-06	1	10000000	100000	10000000	100000
CellLines	Remove violating features	No	Yes	No	Yes	No	No	No	No
	Filter markers/range	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	Best lambda	0.0001	10	100	10	10	10000	0.001	10000
LiverBrainLung	Remove violating features	Yes	Yes	Yes	Yes	Yes	No	Yes	No
	Filter markers/range	No	Yes	No	Yes	Yes	Yes	Yes	Yes
	Best lambda	10000	10000	10000	10000	1000	1000	1000	0.0001
PERT_Cultured	Remove violating features	No	No	Yes	Yes	Yes	Yes	Yes	Yes
	Filter markers/range	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	Best lambda	10000	10000	1000	1000	10000	100000	10000	10000
PERT_Uncultured	Remove violating features	No	Yes	Yes	Yes	Yes	Yes	Yes	No
	Filter markers/range	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No
	Best lambda	1000	1000	1000	1000	10000	10000	1000	100000
RatBrain	Remove violating features	No	No	No	No	No	No	No	No
	Filter markers/range	No	No	No	No	Yes	Yes	No	No
	Best lambda	0.1	100	1000	1000	1E-06	0.001	100000	100000

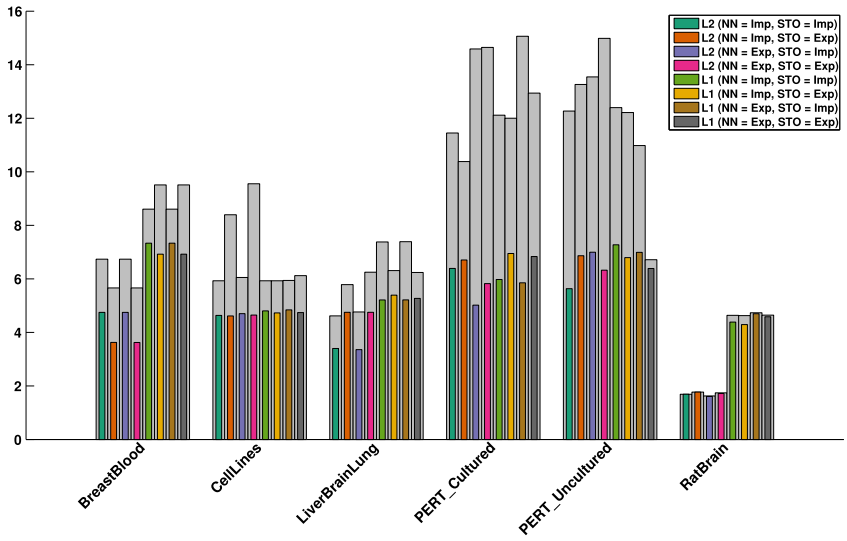


Fig. 22. ***Performance of deconvolution before/after applying combined feature selection/regularization. Gray shade is mAD of the original deconvolutions (the smaller, the better).***

J. Summary

Based on our observations, we propose the following guidelines for the deconvolution of expression data sets.

1) Preprocess reference profiles and mixtures using invariant, universally expressed (housekeeping) genes to ensure that the similar cell quantity (SCQ) constraint is satisfied.
2) Filter violating features that cannot satisfy the STO constraint.
3) Filter lower and upper bounds of gene expressions using adaptive range filtering.
4) Select invariant (among references and between references and samples) cell-type-specific markers to enhance the discriminating power of the basis matrix.
5) Solve the regression using the $\mathcal{L}_2$ loss function together with an $\mathcal{R}_2$ regularizer, or group LASSO if sparsity is desired among groups of tissues/cell types.
6) Use the L-curve method to identify the optimal balance between the regression fit and the regularization penalty.

IV. CONCLUDING REMARKS

In this paper, we present a comprehensive review of different methods for deconvolving linear mixtures of cell types in complex tissues. We perform a systematic analysis of the impact of different algorithmic choices on the performance of the deconvolution methods, including the choice of the loss function, constraints on solutions, data filtering, feature selection, and regularization. We find $\mathcal{L}_2$ loss to be superior in cases wherein the reference cell types are representative of constitutive cell types in the mixture, whereas $\mathcal{L}_1$ outperforms the $\mathcal{L}_2$ in cases in which this condition does not hold. Explicit enforcement of the STO constraint typically degrades the performance of deconvolution. We propose simple bounds to identify features violating this constraint and evaluate the total number of violating features in each data set. We observe an unexpectedly high number of features that cannot satisfy the STO condition, which can be attributed to problems with normalization of expression profiles, specifically normalizing references and samples with respect to each other. In terms of filtering the range of expression values, we find that fixed thresholding is not effective and develop an adaptive method for filtering each data set individually. Furthermore, we observed that range filtering is not always beneficial for deconvolution and, in fact, in some cases, it can deteriorate the performance. We implement two commonly used marker selection methods from the literature to assess their effect on the deconvolution process. Orthogonalizing reference profiles can enhance the discriminating power of the basis matrix. However, due to the known correlation between the mean and variance of expression values, this process alone does not always provide satisfactory results. Another key factor to consider is the low biological variance of genes to enhance the reproducibility of the results and allow deconvolution with noisy references. The combination of range filtering and marker selection eliminates genes with high mean expression, which in turn enhances the observed results. Finally, we address the application of Tikhonov regularization in cases wherein reference cell types are highly correlated and the regression problem is ill-posed.

We summarize our findings in a simple set of guidelines and identify open problems that need further investigation. Areas of particular interest for future research include: 1) identifying the proper set of filters based on the data sets; 2) expanding deconvolution problem to cases with more complex, hierarchical structure among reference vectors; and 3) selecting optimal features to reduce computation time while maximizing the discriminating power. ■

REFERENCES

[1] M. Pedersen, U. Kjems, K. Rasmussen, and L. Hansen, "Semi-blind source separation using head-related transfer functions [speech signal separation]," in *Proc. IEEE Int. Conf. Acoust. Speech, Signal Process.*, vol. 5, 2004, p. V-713-16.

[2] M. Yu, W. Ma, J. Xin, and S. Osher, "Multi-channel l1 regularized convex speech enhancement model and fast computation by the split bregman method," *IEEE Trans. Audio, Speech, Language Process.*, vol. 20, no. 2, pp. 661–675, 2012.

[3] E. Vincent, N. Bertin, R. Gribonval, and F. Bimbot, "From blind to guided audio source separation: How models and side information can improve the separation of sound," *IEEE Signal Process. Mag.*, vol. 31, no. 3, pp. 107–115, May 2014.

[4] N. Souviraà-Labastie, A. Olivero, E. Vincent, and F. Bimbot, "Multi-channel audio source separation using multiple deformed references," *IEEE Trans. Audio, Speech, Language, Process.*, vol. 23, no. 11, pp. 1775–1787, Jun. 2015.

[5] N. Gillis, "Successive nonnegative projection algorithm for robust nonnegative blind source separation," *SIAM J. Imaging Sci.*, vol. 7, no. 2, pp. 1420–1450, Jan. 2014.

[6] W.-K. Ma *et al.*, "A signal processing perspective on hyperspectral unmixing: Insights from remote sensing," *IEEE Signal Process. Mag.*, vol. 31, no. 1, pp. 67–81, Jan. 2014.

[7] D. Nuzillard and A. Bijaoui, "Blind source separation and analysis of multispectral astronomical images," *Astronomy, Astrophysics Suppl. Series*, vol. 147, pp. 129–138, Nov. 2000.

[8] V. P. Pauca, J. Piper, and R. J. Plemmons, "Nonnegative matrix factorization for spectral data analysis," *Linear Algebra Appl.*, vol. 416, no. 1, pp. 29–47, Jul. 2006.

[9] E. Villeneuve and H. Carfantan, "Hyperspectral data deconvolution for galaxy kinematics with mcmc," in *Proc. 20th Eur. Signal Process. Conf.*, 2012, pp. 2477–2481.

[10] A. C. Tang, B. A. Pearlmutter, M. Zibulevsky, and S. A. Carter, "Blind source separation of multichannel neuromagnetic responses," *Neurocomputing*, vol. 3233, pp. 1115–1120, 2000.

[11] C. Hesse and C. James, "On semi-blind source separation using spatial constraints with applications in eeg analysis," *IEEE Trans. Biomed. Eng.*, vol. 53, no. 12, pp. 2525–2534, 2006.

[12] C. Vaya, J. J. Rieta, C. Sanchez, and D. Moratal, "Convolutive blind source separation algorithms applied to the electrocardiogram of atrial fibrillation: Study of performance," *IEEE Trans. Biomed. Eng.*, vol. 54, no. 8, pp. 1530–1533, 2007.

[13] K. Zhang and A. Hyvärinen, "Source separation and higher-order causal analysis of MEG and EEG," *Comput. Res. Repository*, vol. abs/1203.3533, 2012.

[14] A. Kuhn, A. Kumar, A. Beilina, A. Dillman, M. R. Cookson, and A. B. Singleton, "Cell population-specific expression analysis of human cerebellum," *BMC Genomics*, vol. 13, p. 610, 2012.

[15] A. M. Newman *et al.*, "Robust enumeration of cell subsets from tissue expression profiles," *Nature Methods*, vol. 12, no. 5, pp. 453–457, 2015.

[16] S. S. Shen-Orr and R. Gaujoux, "Computational deconvolution: Extracting cell type-specific information from heterogeneous samples," *Current Opinion in Immunology*, vol. 25, no. 5, pp. 571–578, Oct. 2013.

[17] J. Kim, Y. He, and H. Park, "Algorithms for nonnegative matrix and tensor factorizations: A unified view based on block coordinate descent framework," *J. Global Optimization*, vol. 58, no. 2, pp. 285–319, 2013.

[18] O. L. Mangasarian and D. R. Musicant, "Robust linear and support vector regression," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 9, pp. 950–955, 2000.

[19] V. Vapnik, *Statistical Learning Theory*. New York, NY, USA:Wiley, 1998.

[20] A. J. Smola and B. Schölkopf, "A tutorial on support vector regression," *Statist. Comput.*, vol. 14, no. 3, pp. 199–222, 2004.

[21] A. R. Abbas, K. Wolslegel, D. Seshasayee, Z. Modrusan, and H. F. Clark, "Deconvolution of blood microarray data identifies cellular activation patterns in systemic lupus erythematosus," *PloS One*, vol. 4, no. 7, p. e6098, 2009.

[22] W. Qiao, G. Quon, E. Csaszar, M. Yu, Q. Morris, and P. W. Zandstra, "PERT: A method for expression deconvolution of human blood samples from varied microenvironmental and developmental conditions," *PLoS Comput. Biology*, vol. 8, no. 12, p. e1002838, 2012.

[23] T. Gong *et al.*, "Optimal deconvolution of transcriptional profiling data using quadratic programming with application to complex clinical blood samples," *PloS One*, vol. 6, no. 11, p. e27156, 2011.

[24] Z. Altboum *et al.*, "Digital cell quantification identifies global immune cell dynamics during influenza infection," *Molecular Syst. Biology*, vol. 10, no. 2, pp. 720–720, Mar. 2014.

[25] S. S. Shen-Orr *et al.*, "Cell type-specific gene expression differences in complex tissues," *Nature Methods*, vol. 7, no. 4, pp. 287–289, 2010.

[26] D. Venet, F. Pecasse, C. Maenhaut, and H. Bersini, "Separation of samples into their constituents using gene expression data," *Bioinformatics*, vol. 17, no. Suppl 1, pp. S279–S287, 2001.

[27] D. Repsilber *et al.*, "Biomarker discovery in heterogeneous tissue samples-taking the in-silico deconfounding approach," *BMC Bioinformatics*, vol. 11, p. 27, 2010.

[28] N. S. Zuckerman, Y. Noam, A. J. Goldsmith, and P. P. Lee, "A self-directed method for cell-type identification and separation of gene expression microarrays," *PLoS Comput. Biology*, vol. 9, no. 8, p. e1003189, 2013.

[29] R. Gaujoux and C. Seoighe, "Semi-supervised nonnegative matrix factorization for gene expression deconvolution: A case study," *Infection, Genetics and Evolution: J. Molecular Epidemiology and Evolutionary Genetics in Infectious Diseases*, vol. 12, no. 5, pp. 913–921, 2012.

[30] Y. Zhong, Y.-W. Wan, K. Pang, L. M. L. Chow, and Z. Liu, "Digital sorting of complex tissues for cell type-specific gene expression profiles," *BMC Bioinformatics*, vol. 14, p. 89, 2013.

[31] A. Kuhn, D. Thu, H. J. Waldvogel, R. L. M. Faull, and R. Luthi-Carter, "Population-specific expression analysis (PSEA) reveals molecular changes in diseased brain," *Nature Methods*, vol. 8, no. 11, pp. 945–947, 2011.

[32] N. Wang *et al.*, "Mathematical modelling of transcriptional heterogeneity identifies novel markers and subpopulations in complex tissues," *Scientific Rep.*, vol. 6, 2016, Art. no. 18 909.

[33] L. Chen *et al.*, "CAMCM: A signal deconvolution tool for in vivo dynamic contrast-enhanced imaging of complex tissues," *Bioinformatics*, vol. 27, no. 18, pp. 2607–2609, 2011.

[34] T.-H. Chan, W.-K. Ma, C.-Y. Chi, and Y. Wang, "A convex analysis framework for blind separation of non-negative sources," *IEEE Trans. Signal Process.*, vol. 56, no. 10, pp. 5120–5134, 2008.

[35] R. Schwartz and S. E. Shackney, "Applying unmixing to gene expression data for tumor phylogeny inference," *BMC Bioinformatics*, vol. 11, p. 42, 2010.

[36] T. Erkkilä, S. Lehmusvaara, P. Ruusuvuori, T. Visakorpi, I. Shmulevich, and H. Lähdesmäki, "Probabilistic analysis of gene expression measurements from heterogeneous tissues," *Bioinformatics*, vol. 26, no. 20, pp. 2571–2577, 2010.

[37] W. Ju *et al.*, "Defining cell-type specificity at the transcriptional level in human disease," *Genome Res.*, vol. 23, no. 11, pp. 1862–1873, 2013.

[38] G. Quon, S. Haider, A. G. Deshwar, A. Cui, P. C. Boutros, and Q. Morris, "Computational purification of individual tumor gene expression profiles leads to significant improvements in prognostic prediction," *Genome Medicine*, vol. 5, no. 3, p. 29, 2013.

[39] D. A. Liebner, K. Huang, and J. D. Parvin, "MMAD: Microarray microdissection with analysis of differences is a computational tool for deconvoluting cell type-specific contributions from tissue samples," *Bioinformatics*, vol. 30, no. 5, pp. 682–689, 2014.

[40] R. Gaujoux and C. Seoighe, "CellMix: A comprehensive toolbox for gene expression deconvolution," *Bioinformatics*, vol. 29, no. 17, pp. 2211–2212, 2013.

[41] S. Siegert *et al.*, "Transcriptional code and disease map for adult retinal cell types," *Nature Neuroscience*, vol. 15, no. 3, pp. 487–495, 2012.

[42] M. Grant and S. Boyd, "Graph implementations for nonsmooth convex programs," in *Recent Advances in Learning and Control* (Lecture Notes in Control and

Information Sciences), V. Blondel, S. Boyd, and H. Kimura, Eds., London, U.K.: Springer London, 2008, pp. 95–110. [Online]. Available: http://stanford.edu/~boyd/papers/pdf/graph_dcp.pdf

[43] CVX: Matlab Software for Disciplined Convex Programming, Version 2.1. [Online]. Available: http://cvxr.com/cvx, Mar. 2014.

[44] MOSEK Optimization Software. [Online]. Available: http://www.mosek.com/

[45] E. Eisenberg and E. Y. Levanon, "Human Housekeeping Genes, Revisited," *Trends Genet.*, vol. 10, pp. 569–574, 2013.

[46] J. A. Gagnon-Bartsch and T. P. Speed, "Using control genes to correct for unwanted variation in microarray data," *Biostatistics*, vol. 13, no. 3, pp. 539–552, 2012.

[47] J. Ahn *et al.*, "DeMix: Deconvolution for mixed cancer transcriptomes using raw measured data," *Bioinformatics*, vol. 29, no. 15, pp. 1865–1871, 2013.

[48] H. Kawaji *et al.*, "Comparison of CAGE and RNA-seq transcriptome profiling using clonally amplified and single-molecule next-generation sequencing," *Genome Res.*, vol. 24, no. 4, pp. 708–717, Apr. 2014.

[49] Y. Tu, G. Stolovitzky, and U. Klein, "Quantitative noise analysis for gene expression microarray experiments," *Proc. Nat. Academy Sci. USA*, vol. 99, no. 22, pp. 14 031–14 036, 2002.

[50] M. Jeanmougin, A. de Reynies, L. Marisa, C. Paccard, G. Nuel, and M. Guedj, "Should we abandon the t-test in the analysis of gene expression microarray data: A comparison of variance modeling strategies," *PloS One*, vol. 5, no. 9, p. e12336, Jan. 2010.

[51] N. R. Clark *et al.*, "The characteristic direction: A geometrical approach to identify differentially expressed genes," *BMC Bioinformatics*, vol. 15, no. 1, p. 79, 2014.

[52] K. Van Deun, H. Hoijtink, L. Thorrez, L. Van Lommel, F. Schuit, and I. Van Mechelen, "Testing the hypothesis of tissue selectivity: The intersection-union test and a Bayesian approach," *Bioinformatics*, vol. 25, no. 19, pp. 2588–2594, Oct. 2009.

[53] F. M. G. Cavalli, R. Bourgon, W. Huber, J. M. Vaquerizas, and N. M. Luscombe, "SpeCond: A method to detect condition-specific gene expression," *Genome Biology*, vol. 12, no. 10, p. R101, Jan. 2011.

[54] K. Kadota, J. Ye, Y. Nakai, T. Terada, and K. Shimizu, "ROKU: A novel method for identification of tissue-specific genes," *BMC Bioinformatics*, vol. 7, p. 294, Jan. 2006.

[55] K. D. Birnbaum and E. Kussell, "Measuring cell identity in noisy biological systems," *Nucleic Acids Res.*, vol. 39, no. 21, pp. 9093–9107, Nov. 2011.

[56] S. Mohammadi and A. Grama, "A Novel Method to Enhance the Sensitivity of Marker Detection Using a Refined Hierarchical Prior of Tissue Similarities," *bioRxiv*, Jun. 2015, doi: http://dx.doi.org/10.1101/020685

ABOUT THE AUTHORS

Shahin Mohammadi received the Master's degree in computer science from Purdue University, Lafayette, IN, USA, in Dec. 2012 and is currently pursuing the Ph.D. degree at Purdue.

His research interests include computational biology, machine learning, and parallel computing. His current work spans different areas of Bioinformatics/Systems Biology and aims to develop computational methods coupled with statistical models for data-intensive problems, with application in mining the human tissue-specific transcriptome and interactome.

Neta Zuckerman received the Ph.D. degree in computational biology from the University of Bar-Ilan, Ramat Gan, Israel, in June 2010. She completed her post-doctorate in 2015 as a computational biologist at Stanford University, School of Medicine and City of Hope, Stanford, CA, USA.

She was a Visiting Scholar at the Department of Electrical Engineering, Stanford University. Her research interests focus on investigating the role of immune cells in the setting of various diseases, specifically cancer, utilizing algorithm development and microarray data analysis. She is currently a Computational Biologist at Genentech Inc., San Francisco, CA, USA.

Andrea Goldsmith is the Stephen Harris Professor in the School of Engineering and a Professor of Electrical Engineering at Stanford University, Stanford, CA, USA. Her research interests are in information theory and communication theory, and their application to wireless communications as well as biology and neuroscience. She is the author of three textbooks, including *Wireless Communications*, all published by Cambridge University Press, and an inventor of 28 patents.

Prof. Goldsmith has received several awards for her work, including the IEEE ComSoc Edwin H. Armstrong Achievement Award, the IEEE ComSoc and Information Theory Society Joint Paper Award, and the National Academy of Engineering Gilbreth Lecture Award.

Ananth Grama received the Ph.D. degree in computer science from the University of Minnesota, Minneapolis, MN, USA, in 1996.

He is currently a Professor of Computer Sciences and Associate Director of the Center for Science of Information at Purdue University, Lafayette, IN, USA. His research interests span areas of parallel and distributed computing architectures, algorithms, and applications. He has authored several papers and texts on these topics.

Dr. Grama is a member of the American Association for Advancement of Sciences.