dmht Digital Medicine and Healthcare Technology

IntechOpen Journals

RESEARCH PAPER

Deep Cell-Type Deconvolution from Bulk Gene Expression Data Using DECODE

Eran Hermush and Roded Sharan*

School of Computer Science, Tel Aviv University, Tel Aviv, Israel
*Corresponding author. E-mail: roded@tau.ac.il

Citation
Eran Hermush and Roded Sharan (2024), Deep Cell-Type Deconvolution from Bulk Gene Expression Data Using DECODE. *Digital Medicine and Healthcare Technology* 3(1), 1–11.

DOI
https://doi.org/10.5772/dmht.26

Received: 26 June 2023
Accepted: 27 March 2024
Published: 18 June 2024

Abstract

It is becoming clear that bulk gene expression measurements represent an average over very different cells. Elucidating the expression and abundance of each of the encompassed cells is key to disease understanding and precision medicine approaches. A first step in any such deconvolution is the inference of cell type abundances in the given mixture. Numerous approaches to cell-type deconvolution have been proposed, yet very few take advantage of the emerging discipline of deep learning and most approaches are limited to input data regarding the expression profiles of the cell types in question. Here we present DECODE, a deep learning method for the task that is data-driven and does not depend on input expression profiles. DECODE builds on a deep unfolded non-negative matrix factorization technique. It is shown to outperform previous approaches on a range of synthetic and real data sets, producing abundance estimates that are closer to and better correlated with the real values.

Keywords: deconvolution, bulk gene expression, non-negative matrix factorization, deep learning

1. Introduction

Biological tissues are composed of a variety of distinct cell types. Identifying the composition of cells in tissues can help generate hypotheses regarding

cell-type-specific biological mechanisms with important biomedical applications. For example, patients with a large number of infiltrating T cells are more likely to respond positively to immunotherapy [1]. Thus, there is a need for deconvolving a tissue of interest to its constituent cells.

Flow cytometry is the main standard for experimental deconvolution of a sample. More recently, single-cell RNA sequencing (scRNA-seq) methods have become available. However, these methods have their limitations: Flow cytometry requires prompt and careful processing of samples as well as tissue disaggregation, which may result in the loss of fragile cell types and the distortion of gene expression profiles. ScRNA-seq methods are expensive for large sample studies. Additionally, in these technologies, cell types such as neurons, myocytes, and adipocytes are difficult to be captured due to cell size and morphology.

Thus, several computational methods were suggested for predicting cell fractions from bulk expression data. Most methods rely on a signature matrix of cell-specific expression profiles to predict the cell type abundance. Recent comparative analyses of deconvolution methods [2–10] have highlighted state-of-the-art methods for this task including non-negative least squares (NNLS) [11], CIBERSORT [12], CIBERSORTx [10] which are based on support vector regression and GEDIT [13], which runs linear regression, and SCADEN [9] which employs a deep learning approach. However, most of these methods heavily rely on the input signature matrices, which are global matrices that do not contain information which is specific to the input tissue. Furthermore, most of these methods employ classical regression approaches and do not make use of the rich expressive power of deep models that are expected to have a considerable advantage as more training data become available.

We propose DECODE (DEep Cell-type DEconvolution), a novel deep-learning algorithm to predict the cell type abundance matrix from bulk gene expression data and signature matrix. The algorithm is based on a deep unfolding algorithm for non-negative matrix factorization (NMF) and combines both supervised learning on synthetic data and unsupervised learning to achieve its task. We benchmark DECODE using both simulated and real datasets and show that it outperforms previous approaches. DECODE introduces several key novelties that explain its good performance: (i) signature matrices are not explicitly represented by the model but only used to initialize the model and to generate training data, thus allowing data-driven behavior; moreover, (ii) NMF techniques for simultaneous prediction of cell fractions and signatures cannot be directly used for this problem since they do not guarantee that cell fraction vectors will sum to one, while DECODE can be adjusted to this constraint as it is based on a flexible neural network architecture; (iii) the generation of synthetic data (and subsequent training on) overcomes the small amount of available training data; and (iv) the combination of supervised and

unsupervised training helps the model tune to these two different goals which are both important on real data as the training is unsupervised but the evaluation is according to true (hidden) cell fractions. DECODE is made available at https://github.com/eranhermush/DECODE.

2. Methods

In gene expression deconvolution, the input is a matrix of bulk gene expression across multiple samples and a signature matrix consisting of expression profiles of specific cell types. The goal is to infer a matrix of cell fractions indicating for each sample its cell-type decomposition. We approach this problem using a deep learning algorithm for NMF that aims to factor the input gene expression matrix into the product of the signature matrix and the cell fraction matrix. Due to scarcity of training data, we train the algorithm parameters using a combination of synthetically generated data and real data. A high level pseudo-code of our algorithm appears in Figure 1 and described in the following subsections.

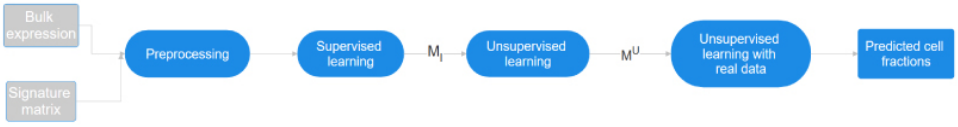

Figure 1. A sketch of the DECODE pipeline. From left to right: DECODE receives a bulk expression matrix and a signature of expression profiles. It first applies supervised training on synthetically generated data using realistic cell fractions for I iterations. The resulting model is trained in an unsupervised fashion for another I iterations on the synthetic data. Finally, the model is further trained on the real data in an unsupervised fashion to produce its predictions.

2.1. Algorithmic background

We assume a bulk expression matrix V of n genes by m samples, where n rows represent the average of cell-type specific gene expression profiles in a sample, weighted by their abundances in that sample. Suppose there are k cell types and S is a *signature matrix*, where k columns are the gene expression profiles of those cell types. Our goal is to infer a matrix F, where m columns are probability vectors denoting the fraction of each cell type in the corresponding sample such that $V \sim S \cdot F$.

Our algorithm, DECODE is based on deep unfolding approach for NMF, DNMF [14]. DNMF is a deep learning algorithm for the decomposition of a non-negative matrix $V_{n\times m}$ into a product of two non-negative matrices $S_{n\times k}$ and $F_{k\times m}$ such that $\|V - SF\|_2$ is minimal. In more detail, DNMF contains several (l)

layers that are updated based on the multiplicative update rule of Lee and Seung [15]. This rule iteratively updates S and F as follows:

$$F_{i+1} \leftarrow F_i \odot \frac{S_i^T V}{S_i^T S_i F_i}; \quad S_{i+1} \leftarrow S_i \odot \frac{V F_i^T}{S_i F_i F_i^T} \tag{1}$$

where $\odot$, $\frac{[.]}{[.]}$ represent entry wise multiplication and division. However, in DNMF this rule is relaxed to allow learning better solutions. Specifically, the algorithm contains a layer for each iteration and aims to optimize F without explicitly representing S. Focusing on some column f of F and the corresponding column v of V , the output f_i of layer i is (up to regularization):

$$f_{i+1} \leftarrow f_i \odot \frac{A_{i+1} v}{B_{i+1} f_i} \tag{2}$$

where A_{i+1} and B_{i+1} are learnable matrices that correspond, in the original update rule, to the matrices S^T and $S^T S$, respectively (ignoring the dependency between the two latter matrices). Starting from an initial matrix F_0, the algorithm rolls it through t layers of the network, imitating the Lee and Seung iterative algorithm, until an output F_t matrix is produced.

DNMF has two model variants: supervised and unsupervised. The supervised variant assumes a known fraction matrix F_{real} which is either experimentally measured or generated in simulations. It is trained with an L_2 loss w.r.t. this matrix: $\|F_l - F_{\text{real}}\|_2$. When no real matrix is available, DNMF uses an unsupervised variant where the loss is the reconstruction error $\|V - SF_l\|_2$. Here we harness these two variants and develop a combined supervised-unsupervised DNMF model that maps from bulk expression (V) to cell fractions (F).

2.2. Model details and training process

We introduce several novelties into the DNMF architecture and training process. First, to account for the fact that each column f of F should represent a probability vector, we normalize f_i after each iteration to sum to one. Second, in order to make use of the given signature matrix S, we initialize a DNMF model M (S) where we set the weights of the first layer by $A_1 = S^T$ and $B_1 = S^T S$ to reflect one iteration of Lee and Seung's update rule. In addition, we initialize F_0 with the result of applying NNLS to V and S. All other parameters are initialized to one as suggested in [14].

Last, we use a combined supervised and unsupervised training process, where the former is based on synthetic data while the latter is based on both synthetic and real data. We first train the model using synthetic data in a supervised fashion for one epoch with $I = 60{,}000$ iterations (or batches). Let M_k be the resulting model

after the k-th iteration (we omit the signature matrix S from the notation for clarity). Next, we take M_I and train it for another I iterations in an unsupervised fashion on synthetic data to yield model M^U. Next, we train M^U for additional I_r = 100 epochs on real data and report the final model. This process is executed for each of a given list of signature matrices and the model with lowest unsupervised error (on the real data) overall is output.

2.3. Supervised training on synthetic data

To deal with data scarcity, we first trained the algorithm on synthetic data. Specifically, we collected known ranges of cell frequencies from [16] and used values within these ranges as parameters for a Dirichlet distribution. We then drew random fraction vectors from this distribution. The resulting fraction matrix was multiplied by the given signature matrix to produce a synthetic bulk expression matrix. We drew multiple matrices in this fashion and fed them sequentially to the training process, viewing each such matrix as representing a batch. We further added a small normally-distributed noise to each bulk expression matrix with zero mean and small standard deviation: for the i-th batch the standard deviation is $i/10{,}000$.

2.4. Unsupervised training on synthetic data

In the unsupervised training, we trained the DECODE model (M_I) to minimize the reconstruction error $\|V - SF_{\text{model}}\|_2$ for I iterations. Before we trained the model, we computed the NNLS supervised loss on the synthetic data and denote its loss by e_{nnls}. For each unsupervised iteration of DECODE, if the supervised loss exceeds e_{nnls} we stopped at this iteration (and not after I).

2.5. Hyperparameter tuning

DECODE has several hyperparameters that need tuning. Due to the small size and number of data sets we opted for using an independent data for tuning the hyperparameters. For number of layers we used 4 for speed considerations, as the model's accuracy is robust to the specific number used, and we used learning rate of 0.001 [14]. Since the problem we are trying to tackle is unsupervised in nature, we followed [14] and did not use regularization. In order to set the number of training iterations (I and I_r) we performed a grid search using an independent stromal dataset from [2] (see next section for detailed description). Specifically, we tried values from 40,000 to 80,000 for I and 50 to 200 for I_r, arriving at the best combination of I = 60,000 and I_r = 100. When applied to the stromal dataset, it can be seen that each of the algorithmic steps aids in improving DECODE's performance (Figure 2).

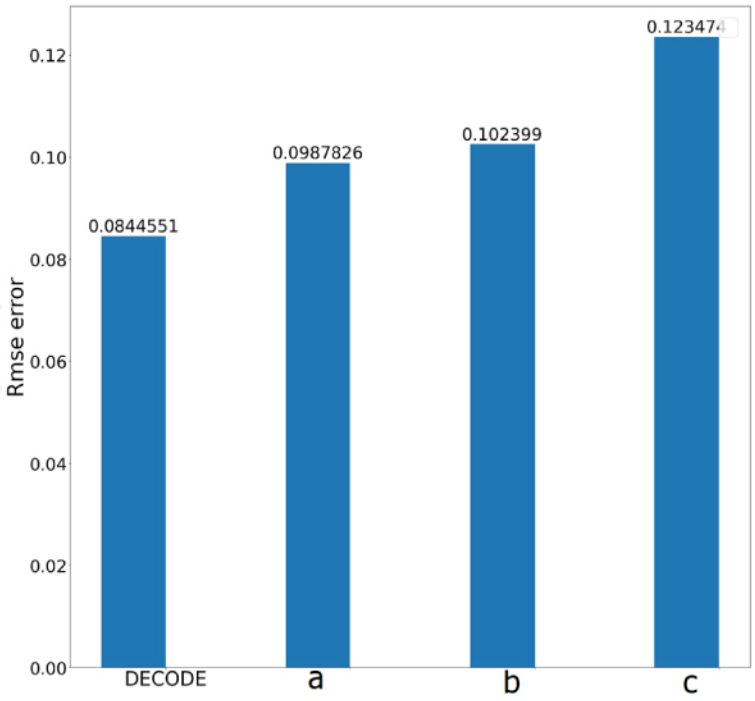

Figure 2. Contribution of different algorithmic stages to the final result (stromal data) compared to the full algorithm: (a) no supervised training on synthetic data; (b) no unsupervised training on synthetic data; and (c) unsupervised training on real data only.

2.6. Data and preprocessing

Good training data for the deconvolution problem is scarce. Our main data source is a recent benchmark paper [2] which has three available datasets, all are results of *in-silico* simulations. Two of the datasets, PBMC1 and PBMC2, represent 200 simulated mixtures of single-cell peripheral blood mononuclear cell (PBMC) expression profiles (100 mixtures in each dataset). For each mixture, individual cells were randomly chosen and their expression profiles summed. Both contain five common PBMC cell types (B, CD4 T, CD8 K, NK and monocytes). A third independent dataset, STROMAL, contains 100 simulated mixtures of stromal cell types (B, CD4 T, CD8 T, macrophage, mast, endothelial and fibroblast cells). We use the first two for testing and the third for hyperparameter tuning.

In addition, we retrieved a real, GSE65133, dataset from [12]. This dataset contains 20 samples of real PBMC cell fractions that were measured by flow cytometry.

To complement the expression datasets, we used known signature matrices from [2]. This study contains a comparative analysis of 9 deconvolution methods with respect to 10 signature matrices. Among the top performing methods were CIBERSORT, NNLS and GEDIT which we use for comparison. We averaged the

accuracy values reported in Figure 1 of [2] for each of the signature matrices and selected the four signatures with the highest accuracy value: Lm22, Skin Signatures, HPCA-Blood, and BlueCode, which are focused in our study. In detail, LM22 [12] contains 22 cell types and 547 genes; Human Primary Cell Atlas (HPCA-Blood) [17] contains 7 cells and 19,715 genes; Blue-Code [18] contains 34 cells and 13,299 genes; and Skin Signatures [19] contains 21 cells and 20,307 genes.

We preprocess the data using GEDIT's approach [13] which removes cell types that are not present in either the input expression or signature matrix, runs quantile normalization on both matrices—such that each column follows the same distribution as every other, removes genes that are missing from either matrix, and selects a subset of 50 genes with lowest entropy for each cell to focus on (for each cell we want the genes that are expressed in a cell type-specific manner. Entropy is minimized when expression is detected only in a single cell type).

2.7. Algorithm comparison

We used several performance measures to compare DECODE to four existing cell deconvolution algorithms: CIBERSORTX, NNLS, GEDIT and SCADEN. We ran GEDIT with its R source code. We ran CIBERSORTX from its official website (https://cibersortx.stanford.edu/). We ran NNLS with its R function. We ran Scaden with its Python source code and kept its default training datasets (as it does not train with a signature matrix). To compare the performance of the five deconvolution algorithms, we measured both RMSE (root mean squared error) and Pearson correlation coefficient, comparing real and predicted cell fractions estimates.

3. Results

We designed a novel algorithm for cell-type deconvolution, DECODE, which is based on DNMF method [14] and a novel learning pipeline in which supervised and unsupervised versions of the method are first applied to synthetic data to enhance the learning process. A high level description of the algorithm is shown in Figure 1. A detailed description of the algorithm and its hyperparameter tuning is elucidated in Methods.

To evaluate DECODE, we applied it to three independent test datasets and compared its performance to those of four state-of-the-art approaches: NNLS [11], CIBERSORTX (an updated version of CIBERSORT) [10], GEDIT [13] and SCADEN [9]. As a first test case, we tested DECODE on two simulated datasets of PBMC cells from [2]. The results are summarized in Figure 3 and show the superiority of our approach compared to previous methods with respect to the two most common evaluation metrics—RMSE and Pearson correlation.

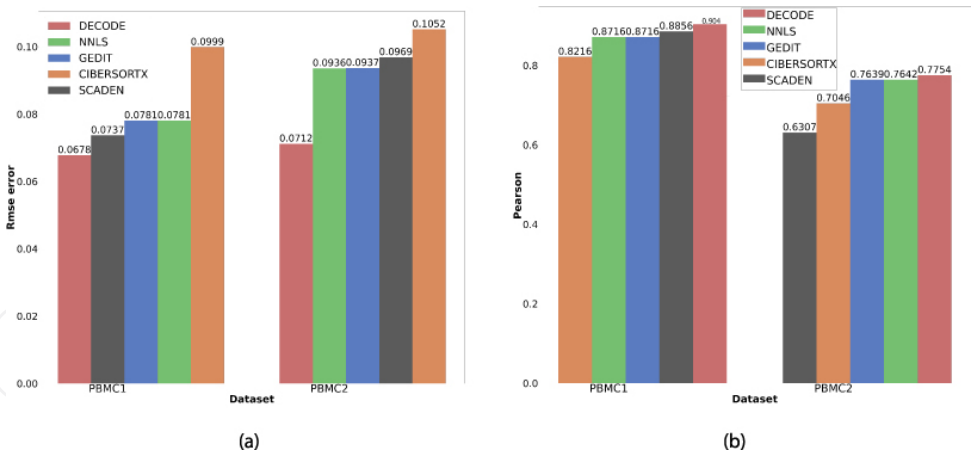

Figure 3. Performance evaluation on simulated data. (a) RMSE performance. (b) Pearson correlation performance.

As a second test, we applied DECODE to a real dataset of PBMCs from [12], again obtaining favorable results (Figures 4 and 5).

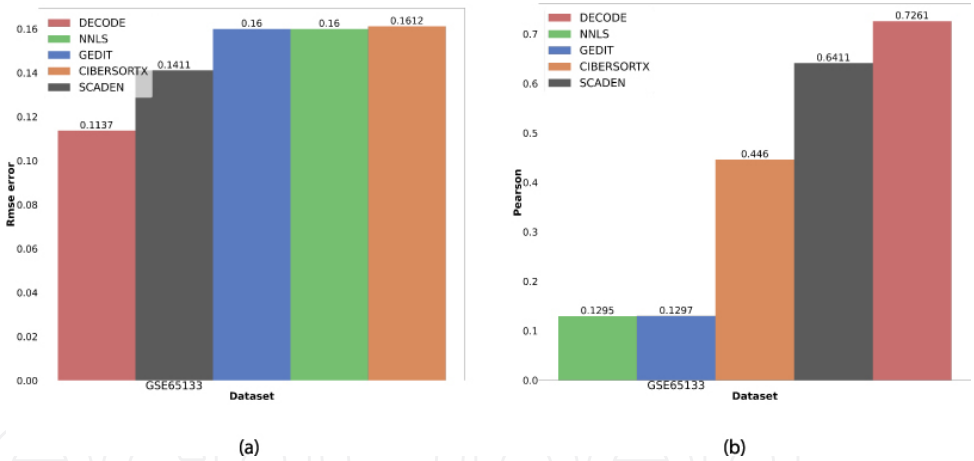

Figure 4. Performance evaluation on real data. (a) RMSE performance. (b) Pearson correlation performance.

In summary, DECODE significantly improved the results of the former methods. Table 1 shows that DECODE produces much lower RMSE errors than the previous best methods.

4. Conclusions

We provided a deep learning framework for deconvolution of bulk gene expression to its cell fractions. Its main innovations include the generation of labeled training

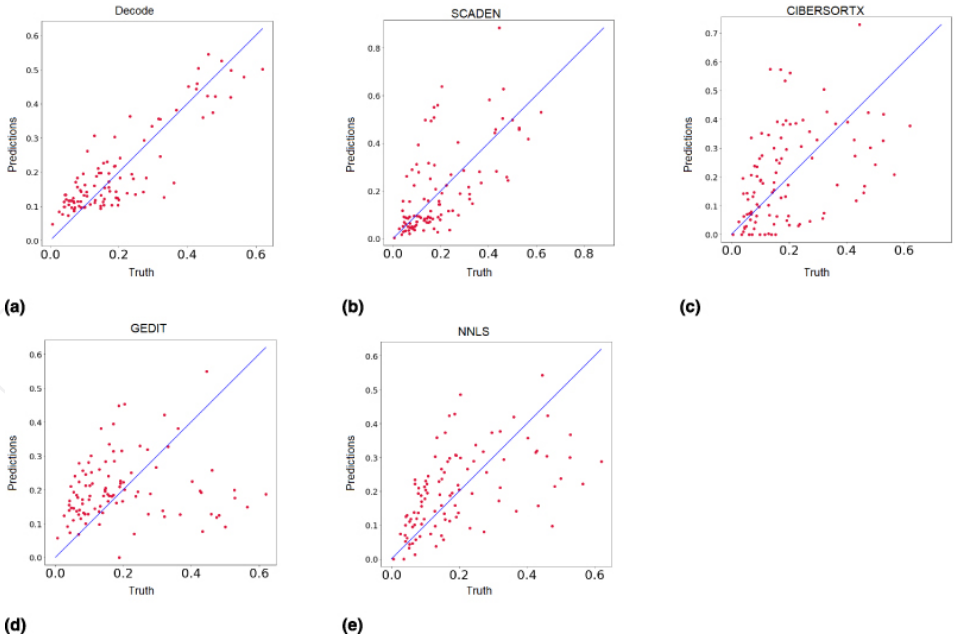

Figure 5. Scatter plots of ground-truth (*x* axis) and predicted values (*y* axis) for DECODE (a), SCADEN (b), CIBERSORTX (c), GEDIT (d) and NNLS (e) on real data.

Table 1. Comparison of DECODE with previous best methods.

Dataset	DECODE RMSE	Previous best result	DECODE improvement (%)
PBMC1	0.0678	0.0737 (by SCADEN)	8
PBMC2	0.0712	0.0936 (by NNLS)	23
Real GSE65133	0.1137	0.1411 (by SCADEN)	20

data and the combination of supervised and unsupervised learning in the training process, as well as the use of DNMF method which does not explicitly code the cell signatures, allowing data-driven behavior. We demonstrated the utility of our framework in deconvolution of simulated and real data.

While DECODE's methodology does not depend on a signature matrix, such a matrix is used in the initialization of the neural network. Future work includes the inference of the signature matrix as part of the learning process so as not to depend on receiving it as input. Another limitation of DECODE is the use of synthetic data for training due to the scarcity of real data. With the accumulation of single cell expression data, a potential way forward is to use these data to simulate deconvolution scenarios and thus improve the training process.

Acknowledgements

RS was supported by a joint program grant from the Cancer Biology Research Center (CBRC), Djerassi Oncology Center, Edmond J. Safra Center for Bioinformatics and Tel Aviv University Center for AI and Data Science (TAD).

Conflict of interest

The authors declare no conflict of interest.

References

1 Şenbabaoğlu Y, Gejman RS, Winer AG, Liu M, Van Allen EM, de Velasco G, et al. Tumor immune microenvironment characterization in clear cell renal cell carcinoma identifies prognostic and immunotherapeutically relevant messenger rna signatures. *Genome Biol*. 2016;**17**(1):1–25.

2 Nadel BB, Oliva M, Shou BL, Mitchell K, Ma F, Montoya DJ, et al. Systematic evaluation of transcriptomics-based deconvolution methods and references using thousands of clinical samples. *Brief Bioinform*. 2021;**22**(6):bbab265. arXiv:https://academic.oup.com/bib/article-pdf/22/6/bbab265/42242154/bbab265.pdf, doi:10.1093/bib/bbab265.

3 Cobos FA, Alquicira-Hernandez J, Powell JE, Mestdagh P, De Preter K. Benchmarking of cell type deconvolution pipelines for transcriptomics data. *Nat Commun*. 2020;**11**(1):1–14.

4 Wang X, Park J, Susztak K, Zhang NR, Li M. Bulk tissue cell type deconvolution with multi-subject single-cell expression reference. *Nat Commun*. 2019;**10**(1):1–9.

5 Patrick E, Taga M, Ergun A, Ng B, Casazza W, Cimpean M, et al. Deconvolving the contributions of cell-type heterogeneity on cortical gene expression. *PLOS Comput Biol*. 2020;**16**(8):e1008120.

6 Cai M, Yue M, Chen T, Liu J, Forno E, Lu X, et al. Robust and accurate estimation of cellular fraction from tissue omics data via ensemble deconvolution. *Bioinformatics*. 2022;**38**(11):3004–3010.

7 Jin H, Liu Z. A benchmark for rna-seq deconvolution analysis under dynamic testing environments. *Genome Biol*. 2021;**22**(1):1–23.

8 Sutton GJ, Poppe D, Simmons RK, Walsh K, Nawaz U, Lister R, et al. Comprehensive evaluation of deconvolution methods for human brain gene expression. *Nat Commun*. 2022;**13**(1):1–18.

9 Menden K, Marouf M, Oller S, Dalmia A, Magruder DS, Kloiber K, et al. Deep learning–based cell composition analysis from tissue expression profiles. *Sci Adv*. 2020;**6**(30):eaba2619.

10 Newman AM, Steen CB, Liu CL, Gentles AJ, Chaudhuri AA, Scherer F, et al. Determining cell type abundance and expression from bulk tissues with digital cytometry. *Nat Biotechnol*. 2019;**37**(7):773–782.

11 Lawson CL, Hanson RJ. *Solving least squares problems*. Philadelphia, PA: SIAM; 1995.

12 Newman AM, Liu CL, Green MR, Gentles AJ, Feng W, Xu Y, et al. Robust enumeration of cell subsets from tissue expression profiles. *Nat Meth*. 2015;**12**(5):453–457.

13 Nadel BB, Lopez D, Montoya DJ, Ma F, Waddel H, Khan MM, et al. The Gene Expression Deconvolution Interactive Tool (GEDIT): accurate cell type quantification from gene expression data. *GigaScience*. 2021;**10**(2):giab002. arXiv:https://academic.oup.com/gigascience/article-pdf/10/2/giab002/36332229/giab002.pdf, doi:10.1093/gigascience/giab002.

14 Nasser R, Eldar YC, Sharan R. Deep unfolding for non-negative matrix factorization with application to mutational signature analysis. *J Comput Biol*. 2022;**29**(1):45–55.

15 Lee D, Seung HS. Algorithms for non-negative matrix factorization. In: *Advances in neural information processing systems*, vol. 13, Cambridge, MA: MIT Press; 2000. p. 535–541.

16 Blood fractions. MACS Handbook [internet]; URL: https://www.miltenyibiotec.com/US-en/resources/macs-handbook/human-cells-and-organs/human-cell-sources/blood-human.html.

17 Mabbott NA, Baillie JK, Brown H, Freeman TC, Hume DA. An expression atlas of human primary cells: inference of gene function from coexpression networks. *BMC Genom*. 2013;**14**(1):1–13.

18 Martens JHA, Stunnenberg HG. Blueprint: mapping human blood cell epigenomes. *Haematologica*. 2013;**98**(10):1487.

19 Swindell WR, Johnston A, Voorhees JJ, Elder JT, Gudjonsson JE. Dissecting the psoriasis transcriptome: inflammatory-and cytokine-driven gene expression in lesions from 163 patients. *BMC Genom*. 2013;**14**(1):1–20.