bioRxiv preprint doi: https://doi.org/10.1101/2020.08.05.237560; this version posted August 5, 2020. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

Imputation of Spatially-resolved Transcriptomes by Graph-regularized Tensor Completion

Zhuliu Li[1], Tianci Song[1], Jeongsik Yong[2] and Rui Kuang[1*]

[1] Department of Computer Science and Engineering, University of Minnesota Twin Cities, Minneapolis, Minnesota, United States of America
[2] Department of Biochemistry, Molecular Biology, and Biophysics, University of Minnesota Twin Cities, Minneapolis, Minnesota, United States of America

* kuang@umn.edu

Abstract

High-throughput spatial-transcriptomics RNA sequencing (sptRNA-seq) based on in-situ capturing technologies has recently been developed to spatially resolve transcriptome-wide mRNA expressions mapped to the captured locations in a tissue sample. One major limitation of in-situ capturing is the high dropout rate of mRNAs that fail the capture or the amplification, which leads to incomplete profiling of the gene expressions. In this paper, we introduce a graph-regularized tensor completion model for imputing the missing mRNA expressions in sptRNA-seq data, namely FIST, Fast Imputation of Spatially-resolved transcriptomes by graph-regularized Tensor completion. We first model sptRNA-seq data as a 3-way sparse tensor in genes (p-mode) and the (x, y) spatial coordinates (x-mode and y-mode) of the observed gene expressions, and then consider the imputation of the unobserved entries as a tensor completion problem in Canonical Polyadic Decomposition (CPD) form. To improve the imputation of highly sparse sptRNA-seq data, we also introduce a protein-protein interaction network to add prior knowledge of gene functions, and a spatial graph to capture the the spatial relations among the capture spots. The tensor completion model is then regularized by a Cartesian product graph of protein-protein interaction network and the spatial graph to capture the high-order relations in the tensor. In the experiments, FIST was tested on ten 10x Genomics Visium spatial transcriptomic datasets of different tissue sections with cross-validation among the known entries in the imputation. FIST significantly outperformed several best performing single-cell RNAseq data imputation methods. We also demonstrate that both the spatial graph and PPI network play an important role in improving the imputation. In a case study, we further analyzed the gene clusters obtained from the imputed gene expressions to show that the imputations by FIST indeed capture the spatial characteristics in the gene expressions and reveal functions that are highly relevant to three different kinds of tissues in mouse kidney. The source code and data are available at https://github.com/kuanglab/FIST.

Author summary

Biological tissues are composed of different types of structurally organized cell units playing distinct functional roles. The exciting new spatial gene expression profiling methods have enabled the analysis of spatially resolved transcriptomes to understand the spatial and functional characteristics of these cells in the context of eco-environment

bioRxiv preprint doi: https://doi.org/10.1101/2020.08.05.237560; this version posted August 5, 2020. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY 4.0 International license.

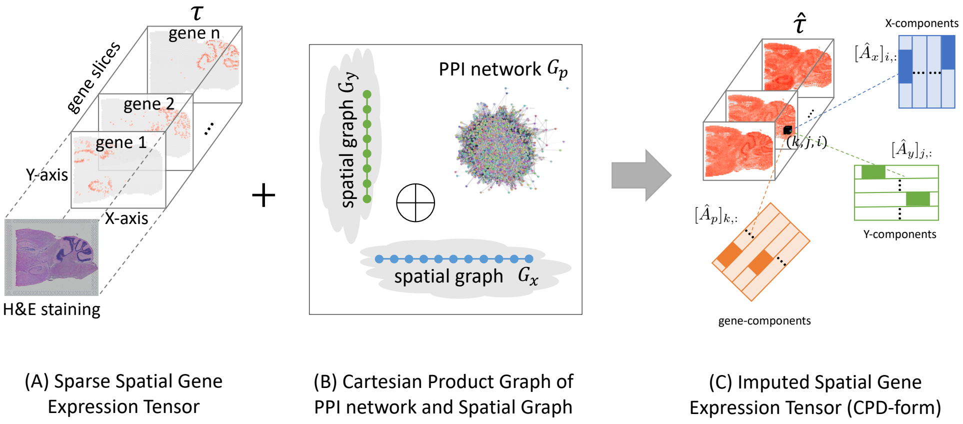
of tissue. Similar to single-cell RNA sequencing data, spatial transcriptomics data also suffers from a high dropout rate of mRNAs in in-situ capture. Our method, FIST (Fast Imputation of Spatially-resolved transcriptomes by graph-regularized Tensor completion), focuses on the spatial and high-sparsity nature of spatial transcriptomics data by modeling the data as a 3-way gene-by-(x, y)-location tensor and a product graph of a spatial graph and a protein-protein interaction network. Our comprehensive evaluation of FIST on ten 10x Genomics Visium spatial genomics datasets and comparison with the methods for single-cell RNA sequencing data imputation demonstrate that FIST is a better method more suitable for spatial gene expression imputation. Overall, we found FIST a useful new method for analyzing spatially resolved gene expressions based on novel modeling of spatial and functional information.

Introduction

Dissection of complex genomic architectures of heterogeneous cells and how they are organized spatially in tissue are essential for understanding the molecular and cellular mechanisms underlying important phenotypes. For example, each tumor is a mixture of different types of proliferating cancerous cells with changing genetic materials [1]. The cancer cell sub-populations co-evolve in the micro-environment formed around their spatial locations. It is important to understand the cell-cell interactions and signaling as well as the functioning of each individual cell to develop effective cancer treatment to eradicate all cancer clones at their locations [2]. Conventional gene expression analyses have been limited to low-resolution bulk profiling that measures the average transcription levels in a population of cells. With single-cell RNA sequencing (scRNA-seq) [3–5], single cells are isolated with a capture method such as fluorescence-activated cell sorting (FACS), Fluidigm C1 or microdroplet microfluidics and then the RNAs are captured, reverse transcribed and amplified for sequencing the RNAs barcoded for the individual origin cells [6,7]. While scRNA-seq is useful for detecting the cell heterogeneity in a tissue sample, it does not provide the spatial information of the isolated cells. To map cell localization, earlier in-situ hybridization methods such as FISH [8], FISSEQ [9], smFISH [10] and MERFISH [11] were developed to profile up to a thousand targeted genes in pre-constructed references with single-molecule RNA imaging. Based on in-situ capturing technologies, more recent spatial transcriptomics RNA sequencing (sptRNA-seq) [12–15] combines positional barcoded arrays and RNA sequencing with single-cell imaging to spatially resolve RNA expressions in each measured spot in the array [12,16–18]. These new technologies have transformed the transcriptome analysis into a new paradigm for connecting single-cell molecular profiling to tissue micro-environment and the dynamics of a tissue region [19–21].

With in-situ capturing technology, mRNAs are captured and sequenced in the spots on the spatial genomic array aligned to the locations on the tissue. For example, spatial transcriptome techniques based on 10x Genomics Visium kit report the counts of mRNAs by unique molecular identifiers (UMIs) in the read-pairs mapped to each gene [22]. Very similar to scRNA-seq data, a significant technical limitation of sptRNA-seq data is known as *dropout*, where dropout events refer to the false quantification of a gene as unexpressed due to the failure in amplifying the transcripts during reverse-transcription [23]. For example, spatial transcriptomic technology's detection efficiency is as low as 6.9% while 10x Genomics Visum has a slightly higher efficiency [24]. It has been shown in previous studies on scRNA-seq data that normalizations will not address the dropout effects [22,25]. In the literature, many imputation methods such as Zero-inflated factor analysis (ZIFA) [26], Zero-Inflated Negative Binomial-based Wanted Variation Extraction (ZINB-WaVE) [27] and

BISCUIT [25] have been developed to impute scRNA-seq. While these methods are also applicable to impute the spatial gene expressions, they ignore a unique characteristic of sptRNA-seq data, which is the spatial information among the gene expressions in the spatial array, and do not fully take advantage of the functional relations among genes for more reliable joint imputation.



(A) Sparse Spatial Gene Expression Tensor

(B) Cartesian Product Graph of PPI network and Spatial Graph

(C) Imputed Spatial Gene Expression Tensor (CPD-form)

Fig 1. Imputation of spatial transcriptomes by graph-regularized tensor completion. **(A)** The input sptRNA-seq data is modeled by a 3-way sparse tensor in genes (p-mode) and the (x, y) spatial coordinates (x-mode and y-mode) of the observed gene expressions. H&E image is also shown to visualize the cell morphologies aligned to the spots. **(B)** A protein-protein interaction network and a spatial graph are integrated as a product graph for tensor completion. The spatial graph is also a product graph of two chain graph for columns (x-mode) and rows (y-mode) in the grid.**(C)** After the imputation, the CPD form of the complete tensor can be used to impute any missing gene expressions, e.g. the entry (k, j, i) can be reconstructed as the sum of the element-wise multiplications of the three components $[\hat{A}_p]_{k,:}$, $[\hat{A}_y]_{j,:}$ and $[\hat{A}_x]_{i,:}$.

To provide a more suitable method for imputation of spatially-resolved gene expressions, we introduce FIST, Fast Imputation of Spatially-resolved transcriptomes by graph-regularized Tensor completion. FIST is a tensor completion model regularized by a product graph as illustrated in Figure 1. FIST models sptRNA-seq data as a 3-way sparse tensor in genes (p-mode) and the (x, y) spatial coordinates (x-mode and y-mode) of the observed gene expressions (Figure 1(A)). As shown in Figure 1(B), a protein-protein interaction network models the interactions between pairs of genes in the gene mode, and the spatial graph is modeled by a product graph of two chain graph for columns (x-mode) and rows (y-mode) in the grid to capture the spatial relations among the (x, y) spots. The tensor product of the two graphs with prior knowledge of gene functions and the spatial relations among the capture spots are then introduced as a regularization of tensor completion to obtain the Canonical Polyadic Decomposition (CPD) of the tensor. The imputation of the unobserved entries can then be derived by reconstructing the entries in the completed tensor shown in Figure 1(C). In the experiments, we comprehensively evaluated FIST on ten 10x Genomics Visium spatial genomics datasets by comparison with widely used methods for single-cell RNA sequencing data imputation. We also analyzed a mouse kidney dataset with more functional interpretation of the gene clusters obtained by the imputed gene expressions

Table 1. Notations

Notation	Definition
G_x, G_y	Spatial graph of (x, y) coordinates
G_p	Protein-protein interaction (PPI) network
n_x, n_y, n_p	Number of vertices in G_x, G_y and G_p
$W_x \in \mathbb{R}_{[0,1]}^{n_x \times n_x}, W_y \in \mathbb{R}_{[0,1]}^{n_y \times n_y}, W_p \in \mathbb{R}_{[0,1]}^{n_p \times n_p}$	Adjacency matrix of G_x, G_y, G_p
$L_x \in \mathbb{R}^{n_x \times n_x}, L_y \in \mathbb{R}^{n_y \times n_y}, L_p \in \mathbb{R}^{n_p \times n_p}$	Graph Laplacian of G_x, G_y, G_p
$\mathfrak{G}(x, y, z)$	Cartesian product of G_x, G_y and G_z
$\mathfrak{W}(x, y, z) \in \mathbb{R}_{[0,1]}^{n_x n_y n_p \times n_x n_y n_p}$	Adjacency matrix of $\mathfrak{G}(x, y, z)$
$\mathfrak{L}(x, y, z) \in \mathbb{R}^{n_x n_y n_p \times n_x n_y n_p}$	Graph Laplacian of $\mathfrak{G}(x, y, z)$
$\mathcal{T} \in \mathbb{R}_+^{n_p \times n_y \times n_x}$	Incomplete spatial gene expression tensor
$\hat{\mathcal{T}} \in \mathbb{R}_+^{n_p \times n_y \times n_x}$	Complete spatial gene expression tensor
$\mathcal{M} \in \mathbb{R}_{[0,1]}^{n_p \times n_y \times n_x}$	Binary mask tensor
$\hat{A}_x \in \mathbb{R}_+^{n_x \times r}$, $\hat{A}_y \in \mathbb{R}_+^{n_y \times r}$, $\hat{A}_p \in \mathbb{R}_+^{n_p \times r}$	CPD component matrices of $\hat{\mathcal{T}}$
$\mathbf{vec}(\mathcal{T}) \in \mathbb{R}^{n_x n_y n_p \times 1}$	Convert $\mathcal{T}$ to be a vector

to detect highly relevant functions in the clusters expressed in three kidney tissue regions, corex, outer stripe of the outer medulla (OSOM) and inner stripe of the outer medulla (ISOM).

Materials and methods

In this section, we first describe the task of spatial gene expression imputation, and next introduce the mathematical model for graph-regularized tensor completion problem. We then present a fast iterative algorithm FIST to solve the optimization problem defined to optimize the model. We also provide the convergence analysis of proposed algorithm in Appendix. Finally, we provide a review of several state-of-the-art methods for scRNA-seq data imputation, which are also compared in the experiments later. The notations which will be used for the derivations in the forthcoming sections are summarized in Table 1.

Imputation of spatial gene expressions by tensor modeling

Let $\mathcal{T} \in \mathbb{R}_+^{n_p \times n_y \times n_x}$ be the 3-way sparse tensor of the observed spatial gene expression data as show in Figure 1(A), with its zero entries representing the missing gene expressions, where n_p denote the total number of genes, n_x and n_y denote the dimensions of the x and y spatial coordinates of the spatial transcriptomics array. Our goal is to learn a complete spatial gene expression tensor $\hat{\mathcal{T}} \in \mathbb{R}_+^{n_p \times n_y \times n_x}$ from $\mathcal{T}$ as illustrated in Figure 1(C). Apparently, it becomes computationally expensive and often infeasible to compute or store a dense tensor $\hat{\mathcal{T}}$, especially in high spatial resolutions with millions of spots. Therefore, we propose to compute an economy-size representation of $\hat{\mathcal{T}}$ via an equality constraint $\hat{\mathcal{T}} = [\![\hat{A}_p, \hat{A}_y, \hat{A}_x]\!]$, which is called *Canonical Polyadic Decomposition (CPD)* [28] of $\hat{\mathcal{T}}$ defined below

$$\hat{\mathcal{T}} = [\![\hat{A}_p, \hat{A}_y, \hat{A}_x]\!] = \sum_{i=1}^{r} [\hat{A}_p]_{:,i} \circ [\hat{A}_y]_{:,i} \circ [\hat{A}_x]_{:,i},$$

where r is the rank of $\hat{\mathcal{T}}$, and $\circ$ denotes the vector outer product. Here, $[\hat{A}_x]_{:,i}$ is the i-th column of the low-rank matrix $\hat{A}_x \in \mathbb{R}^{n_x \times r}$, which can be similarly defined for $[\hat{A}_y]_{:,i}$ and $[\hat{A}_p]_{:,i}$. By utilizing the tensor CPD form, we replaced the optimization variables from $\hat{\mathcal{T}}$ to $\hat{A}_p$, $\hat{A}_y$ and $\hat{A}_x$, reducing the number of parameters from $n_p n_y n_x$ to

$r(n_p + n_y + n_x)$. The advantage of the tensor representation is to incorporate the 2-D spatial x-mode and y-mode such that the grid structure is preserved within the columns and the rows of the spatial array in the tensor, which contains useful spatial information. Next, we introduce the tensor completion model over $\hat{A}_p$, $\hat{A}_y$ and $\hat{A}_x$.

Graph regularized tensor completion model

The key ideas of modeling the task of spatial gene expression imputation are i) the inferred complete spatial gene expression tensor $\hat{\mathcal{T}}$ is regularized to integrate the spatial arrangements of the spot in the tissue array and the functional relations among the genes; ii) the observed part in $\mathcal{T}$ is also required to be preserved in $\hat{\mathcal{T}}$ as the completion task requires; and iii) the inferred tensor $\mathcal{T}$ is compressed as the economy-size representation $\hat{\mathcal{T}} = [\![\hat{A}_p, \hat{A}_y, \hat{A}_x]\!]$ for scalable space and time efficiencies. The novel optimization formulation is shown below in Proposition 1,

Proposition 1. *The complete spatial gene expression tensor $\hat{\mathcal{T}} \in \mathbb{R}^{n_p \times n_y \times n_x}$ can be obtained by solving the following optimization problem:*

$$\begin{aligned} \underset{\{\hat{A}_p, \hat{A}_y, \hat{A}_x\}}{\textit{minimize}} \quad & \frac{1}{2}||\mathcal{M} \circledast (\mathcal{T} - \hat{\mathcal{T}})||_{\mathcal{F}}^2 + \frac{\lambda}{2}\mathbf{vec}(\hat{\mathcal{T}})^T \mathfrak{L}(x, y, p)\mathbf{vec}(\hat{\mathcal{T}}) \\ \textit{subject to} \quad & \hat{\mathcal{T}} = [\![\hat{A}_p, \hat{A}_y, \hat{A}_x]\!] \\ & \hat{A}_p \geq 0, \hat{A}_x \geq 0, \hat{A}_y \geq 0. \end{aligned} \tag{1}$$

where $\lambda \in [0, 1]$ is a model hyperparameter; $\circledast$ denotes the Hadamard product; and $||.||_{\mathcal{F}}$ denotes the Frobenius norm of a tensor.

There are two optimization terms in the model defined in equation (1), consistency with the observations (the first term) and Cartesian product graph regularization (the second term), which are explained below,

- **Consistency with the observations**

  We introduce a binary mask tensor $\mathcal{M}$ to indicate the indices of the observed entries in $\mathcal{T}$. The (i, j, k)-th entry $\mathcal{M}_{i,j,k}$ which is defined below, represents whether the (i, j, k)-th element in $\mathcal{T}$ is observed or not.

  $$\mathcal{M}_{i,j,k} = \begin{cases} 1 & \text{if } \mathcal{T}_{i,j,k} \text{ is observed,} \\ 0 & \text{otherwise.} \end{cases}$$

  By introducing the squared-error in $\mathcal{F}$-norm $||\mathcal{M} \circledast (\mathcal{T} - \hat{\mathcal{T}})||_{\mathcal{F}}^2$ in the model, we ensure the inferred spatial gene expression tensor $\hat{\mathcal{T}}$ is consistent with its observed counterparts in $\mathcal{T}$.

- **Cartesian product graph regularization**

  Two useful assumptions to introduce prior knowledge for inferring the tensor are *1) the spatially adjacent spots should share similar gene expressions, and 2) the expression of two genes are likely highly correlated if they share similar gene functions [29, 30]*. We introduce a spatial graph and a protein-protein interaction (PPI) network into the model. We first encode the spatial information in two undirected unweighted graph $G_x = (V_x, E_x)$ and $G_y = (V_y, E_y)$, where V_x and V_y are vertex sets and E_x and E_y are edge sets. There are $|V_x| = n_x$ vertices in G_x where n_x is the number of the spatial coordinates along the x-axis of the spatial array. Two vertices in G_x are connected by an edge if they are adjacent along the

x-axis. The connections in G_y can be similarly defined to encode the y-coordinates of the tissue. We also incorporate the topological information of a PPI network download from BioGRID 3.5 [31] to use the functional modules in the PPI network. We denote the PPI network as $G_p = (V_p, E_p)$ which contains $|E_p|$ experimentally documented physical interactions among the $|V_p| = n_p$ proteins.

We then use the Cartesian product [32] $\mathfrak{G}(x, y, z) = (V, E)$ of the three individual graphs G_x, G_y and G_p to regularize the elements in $\hat{\mathcal{T}}$, where $|V| = n_x n_y n_p$. The $(v_x v_y v_p)$-th vertex in V represents a triple of vertices $\{v_x \in V_x, v_y \in V_y, v_p \in V_p\}$ from each of the three graphs. The (a_x, a_y, a_p)-th and (b_x, b_y, b_p)-th vertices in V are connected by an edge iff for any $i, j \in \{x, y, p\}$, $(a_i, b_i) \in E_i$ and $a_j = b_j \in V_j$ for all $j \neq i$. Denoting the adjacency, degree and graph Laplacian matrices of graph G_i as W_i, D_i and $L_i = D_i - W_i$ for $i \in \{x, y, p\}$, the adjacency and graph Laplacian matrices of $\mathfrak{G}(x, y, p)$ are obtained as $\mathfrak{W}(x, y, p) = W_x \oplus W_y \oplus W_p$ and $\mathfrak{L}(x, y, p) = L_x \oplus L_y \oplus L_p$ respectively, where $\oplus$ denotes the Kronecker sum [33].

By introducing the term $\mathbf{vec}(\hat{\mathcal{T}})^T \mathfrak{L}(x, y, p) \mathbf{vec}(\hat{\mathcal{T}})$ in equation (1), the inferred gene expression values in $\hat{\mathcal{T}}$ are ensured to be smooth over the manifolds of the product graph $\mathfrak{G}(x, y, p)$, such that a pair of tensor entries $\hat{\mathcal{T}}_{a_p, a_y, a_x}$ and $\hat{\mathcal{T}}_{b_p, b_y, b_x}$ share similar values if the (a_x, a_y, a_p)-th and (b_x, b_y, b_p)-th vertices are connected in $\mathfrak{G}(x, y, p)$. A connection implies that the x-coordinate a_x and b_x is adjacent or y-coordinate a_y and b_y is adjacent or gene a_p and gene b_p are connected in the PPI, with the two other dimensions fixed. Using Cartesian product graph is a more conservative strategy to connect multi-relations in a high-order graph as we have shown in [34] since only replacing one of the dimensions by the immediate neighbors is allowed to create connections. Note that it also possible to use tensor product graph or strong product graph [34] but there could be too many connections to provide meaningful connectivity in the product graph for helpful regularization.

FIST Algorithm

The model introduced in equation (1) is non-convex on variables $\{\hat{A}_p, \hat{A}_y, \hat{A}_x\}$ jointly, thus finding its global minimum is difficult. In this section, we propose an efficient iterative algorithm **F**ast **I**mputation of **Sp**atially-resolved Gene Expression **T**ensor (FIST) to find its local optimal solution using the multiplicative updating rule [35], based on derivatives of $\hat{A}_p$, $\hat{A}_y$ and $\hat{A}_x$. Without loss of generality, we only show the derivations with respect to $\hat{A}_p$, and provide the FIST algorithm in Algorithm 1.

We first bring the equality constraint $\hat{\mathcal{T}} = \llbracket \hat{A}_p, \hat{A}_y, \hat{A}_x \rrbracket$ in Model (1) into the objective function, and rewrite the objective function as

$$
\begin{aligned}
\mathcal{J} &= \mathcal{J}_1 + \lambda \mathcal{J}_2 \qquad (2)\\
\mathcal{J}_1 &= \frac{1}{2} || \mathcal{M} \circledast (\mathcal{T} - \llbracket \hat{A}_p, \hat{A}_y, \hat{A}_x \rrbracket ||_{\mathcal{F}}^2 \\
\mathcal{J}_2 &= \frac{1}{2} \mathbf{vec}(\llbracket \hat{A}_p, \hat{A}_y, \hat{A}_x \rrbracket)^T \mathfrak{L}(x, y, p) \mathbf{vec}(\llbracket \hat{A}_p, \hat{A}_y, \hat{A}_x \rrbracket)
\end{aligned}
$$

The partial derivative of $\mathcal{J}_1$ with respect to $\hat{A}_p$ can be computed as

$$
\frac{\partial \mathcal{J}_1}{\partial \hat{A}_p} = (\mathcal{M}_{(1)} \circledast \hat{\mathcal{T}}_{(1)} - \mathcal{M}_{(1)} \circledast \mathcal{T}_{(1)})(\hat{A}_x \odot \hat{A}_y), \qquad (3)
$$

where $\mathcal{T}_{(1)} \in \mathbb{R}^{n_p \times n_x n_y}$ denotes the matrix flattened from tensor $\mathcal{T}$; $\odot$ denotes the Khatri–Rao product [28]. Note that the term $\mathcal{M}_{(1)} \circledast \hat{\mathcal{T}}_{(1)}$ in Equation (3) implies we only need to compute the entries in $\hat{\mathcal{T}}$ which correspond to the non-zero entries (indices of the observed gene expression) in $\mathcal{M}$. The rest of the computation in Equation (3) involves the well-known MTTKRP (matricized tensor times Khatri-Rao product) [36] operation, which is in the form of $\mathcal{X}_{(1)}(\hat{A}_x \odot \hat{A}_y)$, and can be computed in $O(r|\mathcal{X}|)$ if $\mathcal{X}$ is a sparse tensor with $|\mathcal{X}|$ non-zeros, and $\hat{A}_x$ and $\hat{A}_y$ have r columns. Thus, the overall time complexity of computing Equation (3) is $O(r|\mathcal{M}|)$.

Following the derivations in [34], we obtain the partial derivatives of the second term $\mathcal{J}_2$ as

$$\frac{\partial \mathcal{J}_2}{\partial \hat{A}_p} = \hat{A}_p(\Phi_x \circledast \Theta_y + \Phi_y \circledast \Theta_x) + L_p\hat{A}_p(\Phi_x \circledast \Phi_y), \tag{4}$$

where $\Phi_i = \hat{A}_i^T\hat{A}_i$, and $\Theta_i = \hat{A}_i^T L_i \hat{A}_i$, for all $i \in \{x, y, p\}$. It is not hard to show that the complexity of computing the Equation (4) is $O(\sum_{i\in\{x,y,p\}}(r^2 n_i + r n_i^2))$.

Next, we combine $\frac{\partial \mathcal{J}_1}{\partial \hat{A}_p}$ and $\frac{\partial \mathcal{J}_2}{\partial \hat{A}_p}$ to obtain the overall derivative as

$$\begin{aligned}\frac{\partial \mathcal{J}}{\partial \hat{A}_p} &= \frac{\partial \mathcal{J}_1}{\partial \hat{A}_p} + \lambda(\frac{\partial \mathcal{J}_2}{\partial \hat{A}_p}) \\ &= [\frac{\partial \mathcal{J}_1}{\partial \hat{A}_p}]^+ - [\frac{\partial \mathcal{J}_1}{\partial \hat{A}_p}]^- + \lambda([\frac{\partial \mathcal{J}_2}{\partial \hat{A}_p}]^+ - [\frac{\partial \mathcal{J}_2}{\partial \hat{A}_p}]^-),\end{aligned} \tag{5}$$

where $[\frac{\mathcal{J}_i}{\partial \hat{A}_p}]^+$ and $[\frac{\mathcal{J}_i}{\partial \hat{A}_p}]^-$ are non-negative components in $\frac{\mathcal{J}_i}{\partial \hat{A}_p}$, which are defined below as

$$\begin{aligned}[\frac{\partial \mathcal{J}_1}{\partial \hat{A}_p}]^+ &= (\mathcal{M}_{(1)} \circledast \hat{\mathcal{T}}_{(1)})(\hat{A}_x \odot \hat{A}_y), \\ [\frac{\partial \mathcal{J}_1}{\partial \hat{A}_p}]^- &= (\mathcal{M}_{(1)} \circledast \mathcal{T}_{(1)})(\hat{A}_x \odot \hat{A}_y), \\ [\frac{\partial \mathcal{J}_2}{\partial \hat{A}_p}]^+ &= \hat{A}_p(\Phi_x \circledast \Theta_y^D + \Phi_y \circledast \Theta_x^D) + D_p\hat{A}_p(\Phi_x \circledast \Phi_y), \\ [\frac{\partial \mathcal{J}_2}{\partial \hat{A}_p}]^- &= \hat{A}_p(\Phi_x \circledast \Theta_y^W + \Phi_y \circledast \Theta_x^W) + W_p\hat{A}_p(\Phi_x \circledast \Phi_y),\end{aligned}$$

where $\Theta_i^D = \hat{A}_i^T D_i \hat{A}_i$ and $\Theta_i^W = \hat{A}_i^T W_i \hat{A}_i$, for all $i \in \{x, y, p\}$. According to Equation (5), the objective function $\mathcal{J}$ objective will monotonically decrease under the following multiplicative updating rule,

$$[\hat{A}_p]_{a,b} \leftarrow [\hat{A}_p]_{a,b}\left(\frac{[\frac{\partial \mathcal{J}_1}{\partial \hat{A}_p}]^-_{a,b} + \lambda[\frac{\partial \mathcal{J}_2}{\partial \hat{A}_p}]^-_{a,b}}{[\frac{\partial \mathcal{J}_1}{\partial \hat{A}_p}]^+_{a,b} + \lambda[\frac{\partial \mathcal{J}_2}{\partial \hat{A}_p}]^+_{a,b}}\right), \tag{6}$$

where $[\hat{A}_p]_{a,b}$ denotes the (a, b)-th element in matrix $\hat{A}_p$. Similarly, we can derive the update rule for $[\hat{A}_x]_{a,b}$ and $[\hat{A}_p]_{a,b}$ as follows,

$$[\hat{A}_y]_{a,b} \leftarrow [\hat{A}_y]_{a,b}\left(\frac{[\frac{\partial \mathcal{J}_1}{\partial \hat{A}_y}]^-_{a,b} + \lambda[\frac{\partial \mathcal{J}_2}{\partial \hat{A}_y}]^-_{a,b}}{[\frac{\partial \mathcal{J}_1}{\partial \hat{A}_y}]^+_{a,b} + \lambda[\frac{\partial \mathcal{J}_2}{\partial \hat{A}_y}]^+_{a,b}}\right), \tag{7}$$

$$[\hat{A}_x]_{a,b} \leftarrow [\hat{A}_x]_{a,b}\left(\frac{[\frac{\partial \mathcal{J}_1}{\partial \hat{A}_x}]^-_{a,b} + \lambda[\frac{\partial \mathcal{J}_2}{\partial \hat{A}_x}]^-_{a,b}}{[\frac{\partial \mathcal{J}_1}{\partial \hat{A}_x}]^+_{a,b} + \lambda[\frac{\partial \mathcal{J}_2}{\partial \hat{A}_x}]^+_{a,b}}\right). \tag{8}$$

We then propose an iterative algorithm FIST in Algorithm to find the local optimum of the proposed graph regularized tensor completion problem with time complexity $O(r|\mathcal{M}| + \sum_{i\in\{x,y,p\}}(r^2 n_i + r n_i^2))$. The theoretical convergence analysis of FIST is given in Appendix.

Algorithm 1 FIST: **F**ast **I**mputation of **Sp**atially-resolved Gene Expression **T**ensor

```
Input: 1) spatial gene expression tensor $\mathcal{T} \in \mathbb{R}_+^{n_p\times n_y\times n_x}$, 2) binary mask tensor $\mathcal{M} \in \mathbb{R}_{[0,1]}^{n_p\times n_y\times n_x}$ which indicates the observed part in $\mathcal{T}$ , 3) protein-protein interaction (PPI) network $G_p$ and 4) hyper parameter $\lambda$.
Construct the spatial chain graphs $G_x$ and $G_y$ as described in the text.
while not converge do
    update $\hat{A}_p$ by Equation (6).
    update $\hat{A}_y$ by Equation (7).
    update $\hat{A}_x$ by Equation (8).
end while
Output: the low-rank matrices $\hat{A}_p, \hat{A}_y$ and $\hat{A}_x$, which forms the CPD representation of the inferred spatial gene expression tensor $\hat{\mathcal{T}} = [\![\hat{A}_p, \hat{A}_y, \hat{A}_x]\!] \in \mathbb{R}_+^{n_p\times n_y\times n_x}$.
```

Related methods for comparison

To benchmark the performance of FIST, we compared it with three matrix factorization (MF)-based methods (with graph regularizations) and a nearest neighbors (NN)-based method, which have been applied to impute various types of biological data including the imputation of dropouts in single-cell RNA sequencing (scRNA-seq) data. Note that NMF-based methods have been shown to be effective for learning latent features and clustering of high-dimension sparse genomic data [37].

- **ZIFA**: Zero-inflated factor analysis (ZIFA) [26] factorizes the single cell expression data $Y \in \mathbb{R}^{N\times D}$ where N and D denote the number of single cells and genes respectively, into a factor loading matrix $A \in \mathbb{R}^{K\times D}$ and a matrix $Z \in \mathbb{R}^{N\times K}$ which spans the latent low-dimensional space where dropouts can happen with a probability specified by an exponential decay associated with the expression levels. The imputed matrix can be computed as $\hat{Y} = ZA + \mu$, where $\mu \in \mathbb{R}^{1\times D}$ is the latent mean vector.

- **REMAP**: Since ZIFA is a probabilistic MF model which does not utilize the spatial and gene networks, we therefore also compare with REMAP [38], which was developed to impute the missing chemical-protein associations for the identification of the genome-wide off-targets of chemical compounds. REMAP factorizes the incomplete chemical-protein interactions matrix into the chemical and protein low-rank matrices, which are regularized by the compound similarity graph and protein sequence similarity (NCBI BLAST [39]) graph respectively.

- **GWNMF**: Both ZIFA and REMAP are only applicable to the spot-by-gene matrix which is a flatten of a input tensor $\mathcal{T}$. Such flattening process assumes the spots are independent from each other, and thus loses the spatial information. To keep the spatial arrangements, we also apply MF to each $n_x \times n_y$ slice in tensor $\mathcal{T}$. Specially, we adopt the graph regularized weighted NMF (GWNMF) [40] method to impute each $n_x \times n_y$ gene slice. We let GWNMF use the same x-axis and y-axis graphs G_x and G_y as described in the previous section to regularize the MF.

Table 2. 10x Genomics spatial transcriptome data from 10 tissue sections.

Dataset	Tissue section	Tensor dimensions	Density
HBA1	Human Breast Cancer (Block A Section 1)	$13,426 \times 60 \times 77$	0.093
HBA2	Human Breast Cancer (Block A Section 2)	$13,470 \times 58 \times 75$	0.100
HH	Human Heart	$7,487 \times 63 \times 70$	0.049
HLN	Human Lymph Node	$12,368 \times 61 \times 78$	0.088
MKC	Mouse Kidney Section (Coronal)	$12,264 \times 41 \times 56$	0.103
MBC	Mouse Brain Section (Coronal)	$13,570 \times 49 \times 74$	0.110
MB1P	Mouse Brain Serial Section 1 (Sagittal-Posterior)	$15,404 \times 62 \times 67$	0.115
MB2P	Mouse Brain Serial Section 2 (Sagittal-Posterior)	$12,497 \times 63 \times 65$	0.077
MB1A	Mouse Brain Serial Section 1 (Sagittal-Anterior)	$12,658 \times 59 \times 66$	0.105
MB2A	Mouse Brain Serial Section 2 (Sagittal-Anterior)	$12,295 \times 63 \times 66$	0.082

- **Spatial-NN**: It has been observed that in sparse high-dimensional scRNA-seq data, constructing a nearest neighbor (NN) graph among cells can produce more robust clusters in the presence of dropouts because of taking into account the surrounding neighbor cells [41]. Such rationale has be considered in the clustering methods such as Seurat [42] and shared nearest neighbors (SNN)-Cliq [41], and can also be adopted to impute the spatial gene expression data. We introduce a SNN-based baseline Spatial-NN using neighbor averaging to compare with FIST. Specifically, to impute the missing expression of a target spot, Spatial-NN first searches its spatially nearest spots with observed gene expressions, then assign their average gene expression to the target spot.

We used the provided Python package[1] to experiment with ZIFA, and the provided MATLAB package[2] to experiment with REMAP. To apply both methods, we rearranged the data tensor $\mathcal{T} \in \mathbb{R}^{n_p \times n_y \times n_x}$ to a matrix $T \in \mathbb{R}^{N \times n_p}$, where $N = n_x n_y$ denotes the total number of spots. The spatial graph of REMAP is constructed by connecting two spots if they are spatially adjacent. REMAP adopts the same PPI network as the gene graph G_p as used by FIST. We used MATLAB to implement GWNMF and Spatial-NN ourselves to impute each gene slice $T_i \in \mathbb{R}^{n_x \times n_y}$ in $\mathcal{T}$. In the comparisons, the graph hyperparameter λ of FIST is only selected from $\{0, 0.01, 0.1, 1\}$. The graph hyperparameters of REMAP and GWNMF are set by searching the grids from $\{0.1, 0.5, 0.9, 1\}$ and $\{0, 0.1, 1, 10, 100\}$ respectively as suggested in the original studies. Note that different methods use different scales of graph hyperparameters since the gradients of their variables with respective to the regularization terms are in different scales. The optimal hyper-parameters are selected by the validation set for FIST. For FIST, REMAP and GWNMF, we applied PCA on matrix $T \in \mathbb{R}^{N \times n_p}$ to determine the rank $r \in [200, 300]$ of the low-rank factor matrices, such that at least 60% of the variance is accounted for by the top-r PCA components of T. The latent dimension K of ZIFA is set to 10 since it is time consuming to run ZIFA with a larger K. We also observed that increasing K from 10 to 50 does not show clear improvement on the imputation accuracy.

Results

In this section, we first describe data preparation and performance measure and then show the results of spatial gene expression imputation. We also analyzed the results by the gene-wise density of the gene expressions and regularization by permuted graphs. Finally, we analyzed the imputed spatial gene expressions in the Mouse Kidney Section

[1] https://github.com/epierson9/ZIFA

[2] https://github.com/hansaimlim/REMAP

dataset to show several interesting gene clusters revealing functional characteristics of the three tissue regions, corex, OSOM and ISOM.

Preparation of spatial gene expression datasets

We downloaded the spatial transcriptomic datasets from 10x Genomics[3], which is a collection of spatial gene expressions in 10 different tissue sections from mouse brain, mouse kidney, human breast cancer, human heart and human lymph node as listed in Table 2. All the sptRNA-seq datasets were collected with 10x Genomics Visium Spatial protocol (v1 chemistry) [14] to profiles each tissue section with a high density hexagonal array with 4,992 spots to achieve a resolution of 55 µm (1-10 cells per spot). To fit a tensor model on the spatial gene expression datasets, we organized each of the 10 datasets into a 3-way tensor $\mathcal{T} \in \mathbb{R}^{n_p \times n_x \times n_y}$, where the (i, j, k)-th entry in $\mathcal{T}$ is the UMI count of the i-th gene at the (k, j)-th coordinate in the array. We set the entries in $\mathcal{T}$ to zeros if their UMI counts is lower than 3. We then removed the genes with no UMI counts across the spots, and removed the empty spots in the boundaries of the four sides in the H&E staining from $\mathcal{T}$. The sizes and densities of the 10 different spatial gene expression tensors after prepossessing are also summarized in Table 2. The log-transformation is finally applied to every entry of $\mathcal{T}$ as $\mathcal{T}_{i,j,k} \leftarrow \log(1 + \mathcal{T}_{i,j,k})$. Finally, we downloaded the Homo sapiens protein-protein interactions (PPI) network from BioGRID 3.5 [31] as the gene network G_p to match with the genes in each dataset.

Performance measures

We applied 5-fold cross-validation to evaluate the performance of imputing spatial gene expressions. Specifically, for each gene, we chose 4-fold of its observed expression values (non-zeros in $\mathcal{T}$) for training and validation, and hold out the rest 1-fold observed expression values as test data. The hyper-parameter λ is optimized by the validation set for FIST. Denoting vectors $\mathbf{t} \in \mathbb{R}^{n \times 1}$ and $\hat{\mathbf{t}} \in \mathbb{R}^{n \times 1}$ as the true and predicted expressions of a target gene on the n hold-out test spots, the prediction performance of the target gene is evaluated by the following three metrics,

- MAE (mean absolute error) $= \frac{1}{n}(\sum_{i=1}^{n} |\mathbf{t}_i - \hat{\mathbf{t}}_i|)$,
- MAPE (mean absolute percentage error) $= \frac{1}{n}(\sum_{i=1}^{n} |\frac{\mathbf{t}_i - \hat{\mathbf{t}}_i}{\mathbf{t}_i}|)$,
- R^2 (coefficient of determination) $= 1 - (\sum_{i=1}^{n}(\mathbf{t}_i - \hat{\mathbf{t}}_i)^2)(\sum_{i=1}^{n}(\mathbf{t}_i - \frac{\sum_{i=1}^{n}\mathbf{t}_i}{n})^2)^{-1}$.

We expect a method to achieve smaller MAE and MAPE and larger R^2 for better performance.

FIST signficantly improves the accuracy of imputing spatial gene expressions.

The performance of gene expression imputation in the five fold cross-validation on the ten sptRNA-seq datasets are shown in Figure 2. The average and standard deviation of the prediction performances across all the genes are shown as error bar plots in Figure 2. FIST consistently outperforms all the baselines with significantly lower MAE and MAPE, and larger R^2 in all the 10 datasets. FIST also shows a more robust performance across all the genes as the variances in all the three evaluation metrics are also lower than the other compared methods. To examine the prediction performance more closely, Figure 3 shows the distributions of MAE of individual genes in the 10

[3] https://support.10xgenomics.com/spatial-gene-expression/datasets/

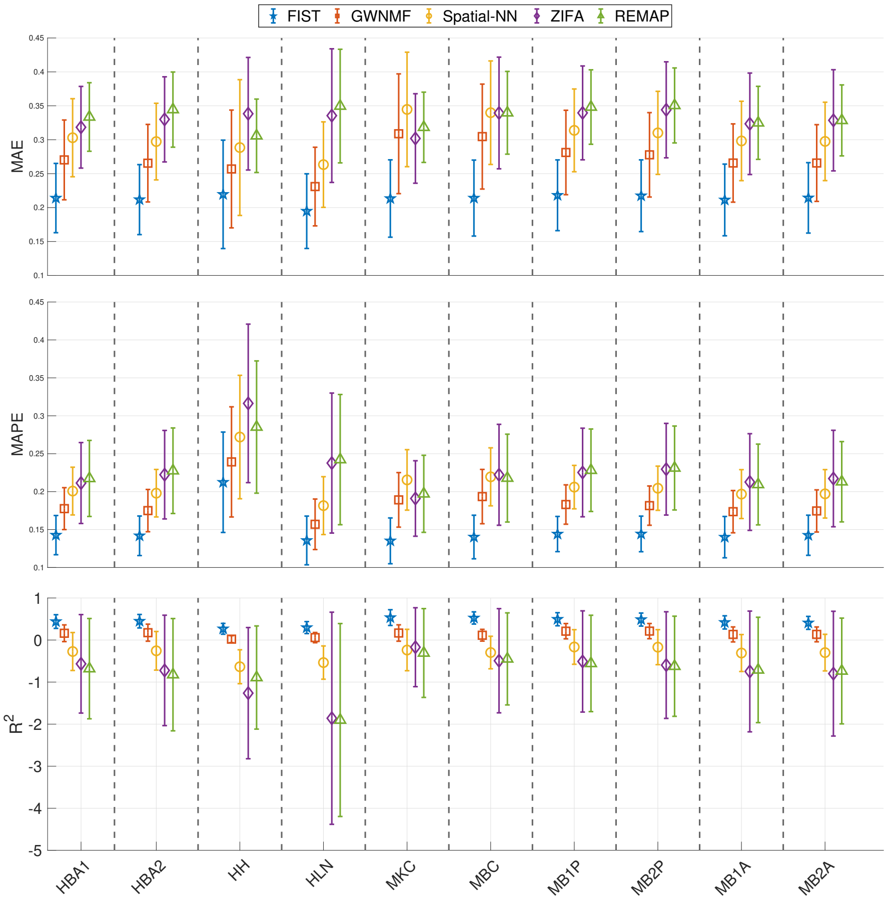


Fig 2. Cross-validation results on gene imputation. The performances of the five compared methods on the 10 tissue sections are measured by 5-fold cross-validation. The error bars (mean and variance) in different shapes and colors show the imputation performance of the methods on all the genes. The result on each of the 10 datasets is shown in one vertical column separated by dashed lines.

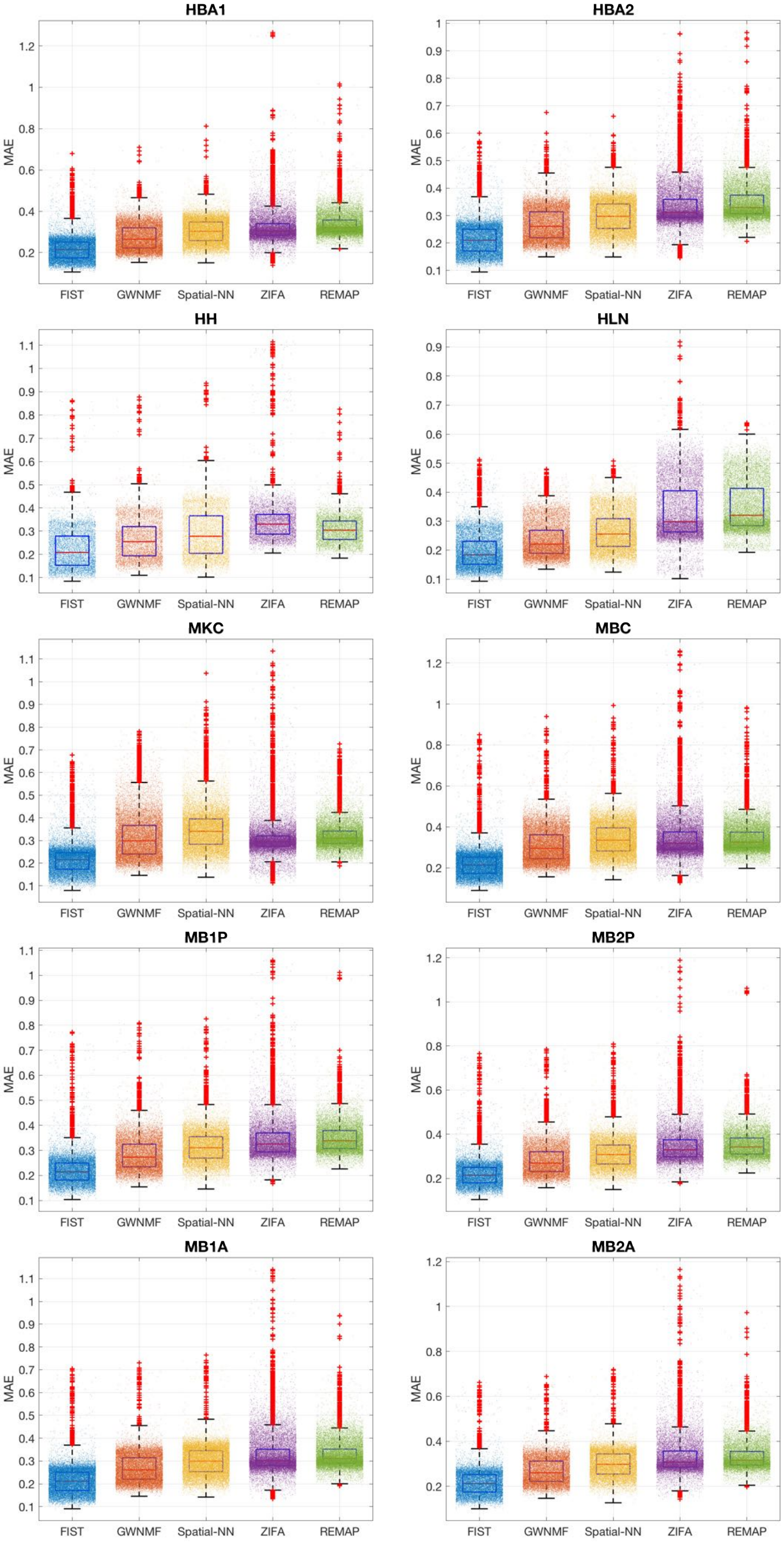
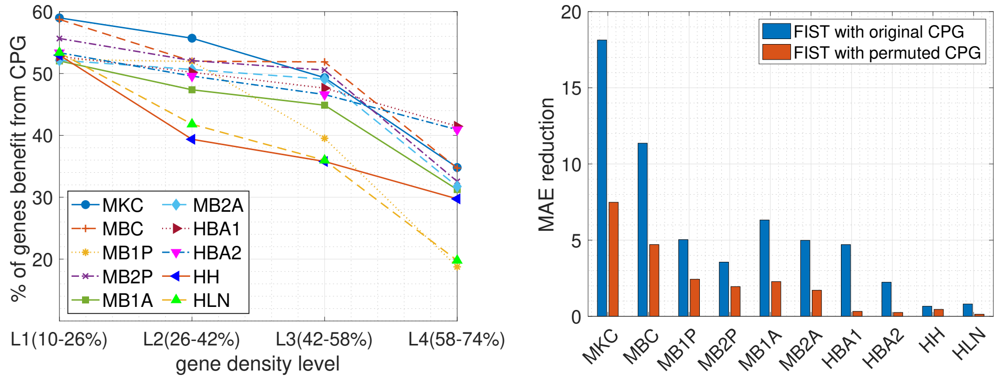


Fig 3. Gene-wise imputation performance by MAE. The performances on the imputations of each gene are shown as box plots. The MAE of every gene slice is denoted by one dot. The performance of each method is shown in each colored box plot.



(A) Imputation performance by density (B) Graph permutation test

Fig 4. Analysis of Cartesian product graph regularization. (A) The percentages of genes benefit from the CPG are plotted by their densities in four different ranges. Each colored line represents one of the 10 datasets. (B) The total reduction of MAE using the original and permuted graphs are compared across the 10 tissue sections.

datasets (the box plots of MAPE and R^2 are given in S1 Figure and S2 Figure). The result is consistent with the overall performance in Figure 2. The observations suggest that FIST indeed performs better than the other methods in the imputation accuracy informed by the spatial information in the tensor model. It is also noteworthy that GNMF, the MF method regularized by the spatial graph applied to each individual gene slice in tensor $\mathcal{T}$, outperforms the other baselines in almost all the datasets. This observation further confirms that the spatial patterns maintained in each gene slice is informative for the imputation task. It is clear that FIST outperformed GNMF with better use of the spatial information coupled with the functional modules of the PPI network G_p and the joint imputation of all the genes in the tensor $\mathcal{T}$.

Cartesian product graph regularization plays a significant role

To understand the role of the regularization by the Cartesian product graph, we further analyzed the genes that are benefitting most from the regularization by the Cartesian product graph in the cross-validation experiments. In particular, in the grid search of the optimal λ weight on the CPG regularization term by the validation set, we count the percentage of the genes with optimal $\lambda = 0.01$ rather than 0, which means completely ignore the regularization. To correlate the improved imputations with the sparsity of the gene expressions, we divided all the genes into 4 equally partitioned groups (L1-4) ordered by their densities in the sptRNA-seq data, where L1 and L4 contain the sparsest and the densest gene slices, respectively. For each of the four density levels, we count the percentage of gene slices that benefit from the CPG regularization and plot the results in Figure 4(A). In the plots, there is a clear trend that the sparser a gene slice, the more likely it benefits from the CPG regularization in all the ten datasets. In the densest L4 group, as low as 20% of the genes can benefit from the CPG regularization versus more than 50% in the sparest L1 group. This is understandable that there is less training information available for sparsely expressed genes (with more dropouts) and the spatial and functional information in the CPG can play a more important role in the imputation by seeking information from the gene's spatial neighbors or the functional neighbors in the PPI network. This observation is also consistent with the fact that the performance of tensor completion tends to degrade severely when only a very small fraction of entries are observed [43,44], and therefore those sparser gene slices tend to benefit more from the side information carried in the CPG.

To further confirm the role of the Cartesian product graph, we also compared the

Table 3. Functional terms enriched by spatial gene clusters (most significant relevant functions)

Region	Cluster	Significantly Enriched GO terms
Cortex	**Cluster 9**	GO:0003073 - regulation of systemic arterial blood pressure (p $= 9.1 \times 10^{-6}$) GO:0008217 - regulation of blood pressure (p $= 1.0 \times 10^{-4}$) GO:0055067 - monovalent inorganic cation homeostasis (p $= 4.3 \times 10^{-4}$) GO:0008015 - blood circulation (p $= 5.3 \times 10^{-3}$) GO:0045777 - positive regulation of blood pressure (p $= 5.8 \times 10^{-3}$) GO:0015672 - monovalent inorganic cation transport (p $= 2.3 \times 10^{-2}$)
	Cluster 23	GO:0086011 - membrane repolarization during action potential (p $= 2.2 \times 10^{-3}$) GO:0034763 - negative regulation of transmembrane transport (p $= 2.2 \times 10^{-3}$) GO:1901017 - negative regulation of potassium ion transmembrane transporter activity (p $= 2.4 \times 10^{-3}$) GO:0032413 - negative regulation of ion transmembrane transporter activity (p $= 2.7 \times 10^{-3}$) GO:1901380 - negative regulation of potassium ion transmembrane transport (p $= 3.4 \times 10^{-3}$)
	Cluster 28	GO:1901605 - alpha-amino acid metabolic process (p $= 4.8 \times 10^{-10}$) GO:0006520 - cellular amino acid metabolic process (p $= 6.4 \times 10^{-9}$) GO:0006790 - sulfur compound metabolic process (p $= 3.1 \times 10^{-6}$) GO:0043648 - dicarboxylic acid metabolic process (p $= 8.4 \times 10^{-6}$)
OSOM	**Cluster 4**	GO:0015711 - organic anion transport (p $= 7.7 \times 10^{-7}$) GO:0046942 - carboxylic acid transport (p $= 1.1 \times 10^{-4}$) GO:0015849 - organic acid transport (p $= 1.1 \times 10^{-4}$) GO:0015718 - monocarboxylic acid transport (p $= 5.0 \times 10^{-3}$)
	Cluster 8	GO:0010498 - proteasomal protein catabolic process (p $= 1.3 \times 10^{-3}$) GO:0006497 - protein lipidation (p $= 1.3 \times 10^{-3}$) GO:0042158 - lipoprotein biosynthetic process (p $= 1.3 \times 10^{-3}$) GO:0043161 - proteasome-mediated ubiquitin-dependent protein catabolic process (p $= 1.3 \times 10^{-3}$)
	Cluster 25	GO:0044282 - small molecule catabolic process (p $= 5.5 \times 10^{-19}$) GO:0016054 - organic acid catabolic process (p $= 1.0 \times 10^{-18}$) GO:0046395 - carboxylic acid catabolic process (p $= 1.0 \times 10^{-18}$) GO:0006631 - fatty acid metabolic process (p $= 2.9 \times 10^{-16}$) GO:0072329 - monocarboxylic acid catabolic process (p $= 9.6 \times 10^{-14}$) GO:0009062 - fatty acid catabolic process (p $= 1.0 \times 10^{-13}$) GO:0044242 - cellular lipid catabolic process (p $= 4.7 \times 10^{-11}$)
	Cluster 52	GO:0006732 - coenzyme metabolic process (p $= 1.2 \times 10^{-10}$) GO:0006520 - cellular amino acid metabolic process (p $= 1.6 \times 10^{-10}$) GO:1901605 - alpha-amino acid metabolic process (p $= 2.3 \times 10^{-9}$) GO:0044282 - small molecule catabolic process (p $= 2.1 \times 10^{-8}$) GO:0000096 - sulfur amino acid metabolic process (p $= 2.3 \times 10^{-7}$)
ISOM	**Cluster 3**	GO:0009150 - purine ribonucleotide metabolic process (p $= 7.4 \times 10^{-5}$) GO:0009259 - ribonucleotide metabolic process (p $= 7.4 \times 10^{-5}$) GO:0006163 - purine nucleotide metabolic process (p $= 7.4 \times 10^{-5}$) GO:0019693 - ribose phosphate metabolic process (p $= 7.4 \times 10^{-5}$) GO:0072521 - purine-containing compound metabolic process (p $= 7.4 \times 10^{-5}$)
	Cluster 5	GO:0048872 - omeostasis of number of cells (p $= 4.5 \times 10^{-5}$) GO:0030218 - erythrocyte differentiation (p $= 3.2 \times 10^{-3}$) GO:0034101 - erythrocyte homeostasis (p $= 3.2 \times 10^{-3}$) GO:0003094 - glomerular filtration (p $= 3.2 \times 10^{-3}$) GO:0097205 - renal filtration (p $= 3.2 \times 10^{-3}$)

performance of FIST using the CPG of G_x, G_y and G_p with the one using a randomly permuted graph from the CPG. To generate the random graph, we first generated three random graphs by permute G_x, G_y and G_p individually which also preserves the degree distributions of the original graphs, by randomly swapping the edges in each graph while keeping the degree of each node. Then we measured the performances of FIST using the original CPG and the CPG obtained from the permuted graphs by MAE reduction, which is the total reduction of MAE on all the genes by varying hyperparameter λ from 0 to 0.01 meaning not using the graph versus using the graph. The comparisons across all the 10 datasets are shown in Figure 4(B). We observe that the FIST using the original graphs receives much higher MAE reduction than the FIST using the permuted graphs. This observation suggests the topology in the original graph topology carry rich information that is helpful for the imputation task beyond just the degree distributions preserved in the random graphs.

FIST imputations recover spatial patterns enriched by highly relevant functional terms

To demonstrate that imputations by FIST can reveal spatial gene expression patterns with highly relevant functional characteristics among the genes in the spatial region, we

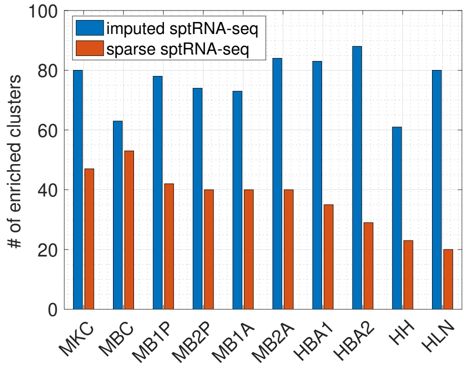
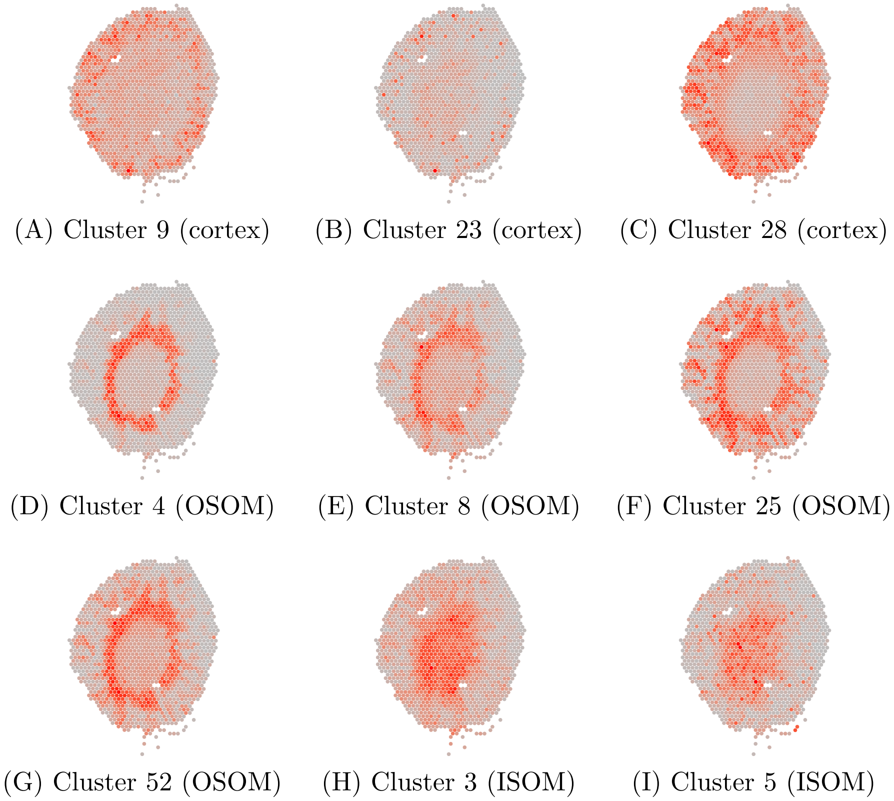


Fig 5. Enrichment analysis on the sparse and imputed sptRNA-seq data. The total number of significantly enriched clusters (with at least one enriched GO term with FDR adjusted p-value < 0.05) in the 10 tissue sections are shown.



(A) Cluster 9 (cortex) (B) Cluster 23 (cortex) (C) Cluster 28 (cortex)

(D) Cluster 4 (OSOM) (E) Cluster 8 (OSOM) (F) Cluster 25 (OSOM)

(G) Cluster 52 (OSOM) (H) Cluster 3 (ISOM) (I) Cluster 5 (ISOM)

Fig 6. FIST recovers spatial gene expression patterns on Mouse Kidney Section. (A)-(C) Gene expression patterns active in the cortex region, (D)-(G) Gene expression patterns active in the outer stripe of the outer medulla (OSOM) region, (H)-(I): Gene expression patterns active in the inner stripe of the outer medulla (ISOM) region.

performed comparative GO enrichment analysis of gene clusters detected with the imputed gene expressions. We conducted a case study on the Mouse Kidney Section data to further analyze the associations between the spatial gene clusters and the relevance between their functional characteristics and three kidney tissue regions, corex, outer stripe of the outer medulla (OSOM) and inner stripe of the outer medulla (ISOM).

To validate the hypothesis that the imputed sptRNA-seq tensor $\widetilde{\mathcal{T}}$ given below

$$\widetilde{\mathcal{T}} = (1 - \mathcal{M}) \circledast \hat{\mathcal{T}} + \mathcal{T}$$

can better capture gene functional modules than the sparse sptRNA-seq tensor $\mathcal{T}$ does, we first rearranged both sptRNA-seq tensors into matrices $\widetilde{T} \in \mathbb{R}^{N \times n_p}$ and $T \in \mathbb{R}^{N \times n_p}$, where $N = n_x n_y$ denotes the total number of spots. We then applied K-means on each matrix to partition the genes into 100 clusters. Next, we used the enrichGO function in the R package clusterProfiler [45] to perform the GO enrichment analysis of the gene clusters. The total number of significantly enriched gene clusters (FDR adjusted p-value < 0.05) in each of the 10 tissue sections are shown in Figure 5, which clearly tells that K-means on the imputed sptRNA-seq data produces much more significantly enriched clusters across all the 10 tissue sections than the sparse sptRNA-seq data without imputation.

Finally, we conducted a case study on the Mouse Kidney Section and present the highly relevant functional characteristics in different tissues in mouse kidney detected with the imputations by FIST. For each of the 100 gene clusters generated by K-means as described above, we collapsed the corresponding gene slices in $\widetilde{\mathcal{T}}$ into a $n_x \times n_y$ matrix by averaging the slices to visualize the center of the gene cluster. The visualizations and enrichment results of all the 100 clusters are given in Table S1. We focus on 3 kinds of representative clusters in Figure 6 which match well with three distinct mouse kidney tissue regions: cortex, ISOM (inner stripe of outer medulla and OSOM (outer stripe of outer medulla). By investigating the enriched GO terms by the clusters (p-values shown in Table 3), we found their functional relevance to cortex, ISOM and OSOM regions. We found that the spatial gene cluster 9 which is highly expressed in cortex specifically enriched biological processes for the regulation of blood pressure (GO:0008217, GO:0003073, GO:0008015 and GO:0045777) and transport/homeostasis of inorganic molecules (GO:0055067 and GO:0015672). The spatial gene cluster 23 and 28 which are also highly expressed in cortex enriched cellular pathways that are critical for the polarity of cellular membranes (GO:0086011, GO:0034763, GO:1901017, GO:0032413 and GO:1901380) and the transport of cellular metabolites (GO:1901605, GO:0006520, GO:0006790 and GO:0043648), respectively. These observations are consistent with previous studies reporting the regulation of kidney function by above listed biological processes in cortex [46–49]. In contrast, the pattern analysis of spatial gene expression in cluster 4, 8, 25 and 52 which are highly expressed in OSOM in kidney showed that catabolic processes of organic and inorganic molecules are specifically enriched such as GO:0015711, GO:0046942, GO:0015849, GO:0015718, GO:0010498, GO:0043161, GO:0044282, GO:0016054, GO:0046395, GO:0006631, GO:0072329, GO:0009062 and GO:0044242. These cellular processes are known to be active in renal proximal tubule which exists across cortex and OSOM [50–55]. Distinctively, the spatial gene clusters highly expressed in ISOM enriched pathways for nucleotide metabolisms (GO:0009150, GO:0009259 and GO:0006163) in cluster 3 and renal filtration (GO:0097205 and GO:0003094) in cluster 5. Collectively, these observations demonstrate that FIST could identify physiologically relevant distinctive spatial gene expression patterns in the mouse kidney dataset. Further, it suggests that FIST can provide a high-resolution anatomical analysis of organ functions in sptRNA-seq data.

Discussions

In this study, we proved that tensor is a natural representation of the multidimensional structure in spatially-resolved gene expression data mapped by the 2D spatial array. To the best of our knowledge, this is the first work to model the imputation of spatially-resolved transcriptomes as a tensor completion problem. Our key observations in the experiments with the ten 10x Genomics Visium spatial transcriptomic datasets are that 1) the imputation accuracy is significantly improved by leveraging the tensor representation of the sptRNA-seq data, and 2) by incorporating the spatial graph and PPI network, the accuracy the imputation and the content of the functional information in the imputed spatial gene expressions can be further improved significantly.

We observed that the genes that are more sparsely expressed can benefit more from the adjacency information in the spatial graph and the functional information in the PPI network. These genes can be empirically detected with a validation set to tune the only hyper-parameter λ for deciding if the regularization by the product graph is needed for the imputation of a gene. Thus, we expect a low risk of overfitting in applying FIST to other datasets. In addition, the functional analysis of the spatial gene clusters detected on the Mouse Kidney Section data further confirms that FIST detects gene clusters with more spatial characteristics that are consistent with the physiological features of the tissue.

Although our experiments focused on medium density 10x Genomics Visium kit array (5000 spots), we also further tested that FIST is also applicable and scalable to high-resolution spatial transcriptomics datasets with millions of spots in the preliminary work in a follow-up study. We tested the high-definition spatial transcriptomics (HDST) datasets generated from [14]. The HDST datasets includes 3 mouse tissue sections from olfactory bulb and 3 human tissue sections from breast cancer using hexagonal array to profile tissue with a high density (1,467,270 spots in total) to achieve a resolution of 2 µm. Our preliminary result suggest that FIST can finish imputing each of the the 6 HDST datasets in $\backsim$ 1hr.

Overall, we concluded that FIST is an effective and easy-to-use approach for reliable imputation of spatially-resolved gene expressions by modeling the spatial relation among the spots in the spatial array and the functional relation among the genes. The imputation results by FIST is both more accurate and functionally interpretable. FIST is also highly generalizable to other spatial transcriotomics datasets with high scalability and only one hyper-parameter needed to tune.

Supporting information

S1 Figure Gene-wise imputation performance by MAPE. The performances on the imputations of each gene are shown as box plots. The MAPE of every gene slice is denoted by one dot. The performance of each method is shown in each colored box plot.

S2 Figure Gene-wise imputation performance by R^2. The performances on the imputations of each gene are shown as box plots. The R^2 of every gene slice is denoted by one dot. The performance of each method is shown in each colored box plot.

S1 Table Enriched GO terms of spatial gene clusters. The GO terms significantly enriched by the genes in each spatial gene cluster (FDR adjusted p-value < 0.05) are shown in the spreadsheet tables.

S1 Appendix. Convergence of FIST.

We follow the convergence analysis in our previous work [34] to show that FIST can converge under the updating rules in Equation (6)-(8).

As the objective function $\mathcal{J}$ in Equation (2) is bounded from below by zero, we can prove the convergence of FIST by showing that $\mathcal{J}$ is non-increasing under each of the updating rules in Equations (6)-(8). Here, we only show that $\mathcal{J}$ is non-increasing under Equations (6). The proof is directly applicable to Equations (7) and (8).

We first expand the derivative in Equation (5) as

$$\frac{\partial \mathcal{J}}{\partial \hat{A}_p} = -X_1 - \hat{A}_p X_2 - W_p \hat{A}_p X_3 + X_4 + \hat{A}_p X_5 + D_p \hat{A}_p X_3,$$

where $X_1 = (\mathcal{M}_{(1)} \circledast \mathcal{T}_{(1)})(\hat{A}_x \odot \hat{A}_y)$, $X_2 = \lambda(\Phi_x \circledast \Theta_y^W + \Phi_y \circledast \Theta_x^W)$, $X_3 = \lambda(\Phi_x \circledast \Phi_y)$, $X_4 = (\mathcal{M}_{(1)} \circledast \hat{\mathcal{T}}_{(1)})(\hat{A}_x \odot \hat{A}_y)$, and $X_5 = \lambda(\Phi_x \circledast \Theta_y^D + \Phi_y \circledast \Theta_x^D)$.

Theorem 1. *Lee and Seung [35]: A function $\mathcal{J}(h)$ is non-increasing under the update $h^* \leftarrow arg\min_h G(h, \tilde{h})$ if $G(h, \tilde{h})$ is an auxiliary function for $\mathcal{J}(h)$, such that the following conditions are satisfied:*

$$G(h, \tilde{h}) \geq \mathcal{J}(h), \quad G(h, h) = \mathcal{J}(h).$$

Based on Theorem 1, $\mathcal{J}$ is non-increasing under the update in Equation (6) if it is an update of one proper *auxiliary function* of $\mathcal{J}(\hat{A}_p)$, which is defined in Theorem 2.

Theorem 2. *The following function*

$$G([\hat{A}_p]_{a,b}, [\tilde{A}_p]_{a,b}) = \mathcal{J}([\tilde{A}_p]_{a,b}) + \mathcal{J}'([\tilde{A}_p]_{a,b})([\hat{A}_p - \tilde{A}_p]_{a,b}) + \frac{[X_4 + \tilde{A}_p X_5 + D_p \tilde{A}_p X_3]_{a,b}}{2[\tilde{A}_p]_{a,b}}([\hat{A}_p - \tilde{A}_p]_{a,b})^2 \tag{9}$$

is an auxiliary function of $\mathcal{J}([\hat{A}_p]_{a,b})$ and has its global minimum.

Proof: First, it is obvious that $G([\hat{A}_p]_{a,b}, [\hat{A}_p]_{a,b}) = \mathcal{J}([\hat{A}_p]_{a,b})$. To show $G([\hat{A}_p]_{a,b}, [\tilde{A}_p]_{a,b}) \geq \mathcal{J}([\hat{A}_p]_{a,b})$ we obtain the second-order Taylor expansion of $\mathcal{J}([\hat{A}_p]_{a,b})$ at the point $[\tilde{A}_p]_{a,b}$ as

$$\begin{aligned}\mathcal{J}([\hat{A}_p]_{a,b}) &= \mathcal{J}([\tilde{A}_p]_{a,b}) + \mathcal{J}'([\tilde{A}_p]_{a,b})([\hat{A}_p]_{a,b} - [\tilde{A}_p]_{a,b}) \\ &\quad + \frac{1}{2}\mathcal{J}''([\tilde{A}_p]_{a,b})([\hat{A}_p]_{a,b} - [\tilde{A}_p]_{a,b})^2,\end{aligned}$$

with the second-order derivative given below:

$$\mathcal{J}''([\tilde{A}_p]_{a,b}) = -[X_2]_{b,b} - [W_p]_{a,a}[X_3]_{b,b} + [X_5]_{b,b} + [D_p]_{a,a}[X_3]_{b,b}$$

Thus, the inequality $G([\hat{A}_p]_{a,b}, [\tilde{A}_p]_{a,b}) \geq \mathcal{J}([\hat{A}_p]_{a,b})$ holds if

$$\frac{[X_4 + \tilde{A}_p X_5 + D_p \tilde{A}_p X_3]_{a,b}}{[\tilde{A}_p]_{a,b}} \geq \mathcal{J}''([\tilde{A}_p]_{a,b}),$$

which can be demonstrated by the facts that

$$[D_p \tilde{A}_p X_3]_{a,b} = \sum_{l,m} [D_p]_{a,l}[\tilde{A}_p]_{l,m}[X_3]_{m,b} \geq [D_p]_{a,a}[X_3]_{b,b}[\tilde{A}_p]_{ab},$$

$$\text{and } [\tilde{A}_p X_5]_{a,b} = \sum_{l} [\tilde{A}_p]_{a,l}[X_5]_{l,b} \geq [X_5]_{b,b}[\tilde{A}_p]_{a,b}. \text{ (End of Proof)}$$

As the *auxiliary function* $G([\hat{A}_p]_{a,b}, [\tilde{A}_p]_{a,b})$ in Equation (9) is a quadratic function on variable $[\hat{A}_p]_{a,b}$, its minimum can be easily obtained in a closed-form as

$$\begin{aligned}[\hat{A}_p]^*_{a,b} &= \underset{[\hat{A}_p]_{a,b}}{\arg\min}\, G([\hat{A}_p]_{a,b}, [\tilde{A}_p]_{a,b}) \\ &= \frac{[\tilde{A}_p]_{a,b}[X_1 + \tilde{A}_p X_2 + W_p \tilde{A}_p X_3]_{a,b}}{[X_4 + \tilde{A}_p X_5 + D_p \tilde{A}_p X_3]_{a,b}},\end{aligned}$$

which leads to the updating rule in Equation (6).

To analyze the optimality of the fixed point after convergence, we first define $\{\Lambda_p \in \mathbb{R}^{n_p \times r}, \Lambda_x \in \mathbb{R}^{n_x \times r}, \Lambda_y \in \mathbb{R}^{n_y \times r}\}$ to be the matrices of Lagrange multipliers with the Lagrange function

$$\mathcal{L} = \mathcal{J} - \sum_{i \in \{p,x,y\}} \mathrm{tr}(\Lambda_i \hat{A}_i^T).$$

Setting $\frac{\partial \mathcal{L}}{\partial \hat{A}_p}$ to be zero, we obtain $\Lambda_p = \frac{\partial \mathcal{J}}{\partial \hat{A}_p}$. Furthermore, when $A^{(i)}$ is a fixed point under the updating in Equation (6) we have

$$[-X_1 - \hat{A}_p X_2 - W_p \hat{A}_p X_3 + X_4 + \hat{A}_p X_5 + D_p \hat{A}_p X_3]_{a,b} [\hat{A}_p]_{a,b} = 0,$$

which implies the KKT complementary slackness condition $[\Lambda_p]_{a,b}[A_p]_{ab} = 0$ is satisfied.

Acknowledgments

This research work was supported by University of Minnesota - Minnesota Robotics Institute Seed Grants and NIH grant R01GM113952-01A1. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

References

1. Heppner GH. Tumor heterogeneity. Cancer research. 1984;44(6):2259–2265.

2. Schmidt F, Efferth T. Tumor heterogeneity, single-cell sequencing, and drug resistance. Pharmaceuticals. 2016;9(2):33.

3. Jaitin DA, Kenigsberg E, Keren-Shaul H, Elefant N, Paul F, Zaretsky I, et al. Massively parallel single-cell RNA-seq for marker-free decomposition of tissues into cell types. Science. 2014;343(6172):776–779.

4. Macosko EZ, Basu A, Satija R, Nemesh J, Shekhar K, Goldman M, et al. Highly parallel genome-wide expression profiling of individual cells using nanoliter droplets. Cell. 2015;161(5):1202–1214.

5. Klein AM, Mazutis L, Akartuna I, Tallapragada N, Veres A, Li V, et al. Droplet barcoding for single-cell transcriptomics applied to embryonic stem cells. Cell. 2015;161(5):1187–1201.

6. Hebenstreit D. Methods, challenges and potentials of single cell RNA-seq. Biology. 2012;1(3):658–667.

7. Liu S, Trapnell C. Single-cell transcriptome sequencing: recent advances and remaining challenges. F1000Research. 2016;5.

8. Lubeck E, Coskun AF, Zhiyentayev T, Ahmad M, Cai L. Single-cell in situ RNA profiling by sequential hybridization. Nature methods. 2014;11(4):360.

9. Lee JH, Daugharthy ER, Scheiman J, Kalhor R, Yang JL, Ferrante TC, et al. Highly multiplexed subcellular RNA sequencing in situ. Science. 2014;343(6177):1360–1363.

10. Shah S, Lubeck E, Schwarzkopf M, He TF, Greenbaum A, Sohn CH, et al. Single-molecule RNA detection at depth by hybridization chain reaction and tissue hydrogel embedding and clearing. Development. 2016;143(15):2862–2867.

11. Chen KH, Boettiger AN, Moffitt JR, Wang S, Zhuang X. Spatially resolved, highly multiplexed RNA profiling in single cells. Science. 2015;348(6233).

12. Vickovic S, Ståhl PL, Salmén F, Giatrellis S, Westholm JO, Mollbrink A, et al. Massive and parallel expression profiling using microarrayed single-cell sequencing. Nature communications. 2016;7(1):1–9.

13. Nawy T. Spatial transcriptomics. Nature Methods. 2018;15(1):30–30.

14. Vickovic S, Eraslan G, Salmén F, Klughammer J, Stenbeck L, Schapiro D, et al. High-definition spatial transcriptomics for in situ tissue profiling. Nature methods. 2019;16(10):987–990.

15. Rodriques SG, Stickels RR, Goeva A, Martin CA, Murray E, Vanderburg CR, et al. Slide-seq: A scalable technology for measuring genome-wide expression at high spatial resolution. Science. 2019;363(6434):1463–1467.

16. Lignell A, Kerosuo L, Streichan SJ, Cai L, Bronner ME. Identification of a neural crest stem cell niche by Spatial Genomic Analysis. Nature communications. 2017;8(1):1–11.

17. Giacomello S, Salmén F, Terebieniec BK, Vickovic S, Navarro JF, Alexeyenko A, et al. Spatially resolved transcriptome profiling in model plant species. Nature Plants. 2017;3(6):17061.

18. Berglund E, Maaskola J, Schultz N, Friedrich S, Marklund M, Bergenstråhle J, et al. Spatial maps of prostate cancer transcriptomes reveal an unexplored landscape of heterogeneity. Nature communications. 2018;9(1):1–13.

19. Smith EA, Hodges HC. The spatial and genomic hierarchy of tumor ecosystems revealed by single-cell technologies. Trends in cancer. 2019;5(7):411–425.

20. Liang SB, Fu LW. Application of single-cell technology in cancer research. Biotechnology advances. 2017;35(4):443–449.

21. Maniatis S, Äijö T, Vickovic S, Braine C, Kang K, Mollbrink A, et al. Spatiotemporal dynamics of molecular pathology in amyotrophic lateral sclerosis. Science. 2019;364(6435):89–93.

22. Vallejos CA, Risso D, Scialdone A, Dudoit S, Marioni JC. Normalizing single-cell RNA sequencing data: challenges and opportunities. Nature methods. 2017;14(6):565.

23. Stegle O, Teichmann SA, Marioni JC. Computational and analytical challenges in single-cell transcriptomics. Nature Reviews Genetics. 2015;16(3):133–145.

24. Asp M, Bergenstråhle J, Lundeberg J. Spatially Resolved Transcriptomes — Next Generation Tools for Tissue Exploration. BioEssays. 2020; p. 1900221.

25. Prabhakaran S, Azizi E, Carr A, et al. Dirichlet process mixture model for correcting technical variation in single-cell gene expression data. In: International Conference on Machine Learning; 2016. p. 1070–1079.

26. Pierson E, Yau C. ZIFA: Dimensionality reduction for zero-inflated single-cell gene expression analysis. Genome biology. 2015;16(1):241.

27. Risso D, Perraudeau F, Gribkova S, et al. A general and flexible method for signal extraction from single-cell RNA-seq data. Nature communications. 2018;9(1):284.

28. Kolda TG, Bader BW. Tensor decompositions and applications. SIAM review. 2009;51(3):455–500.

29. Hwang T, Tian Z, Kuang R, Kocher JP. Learning on weighted hypergraphs to integrate protein interactions and gene expressions for cancer outcome prediction. In: 2008 Eighth IEEE International Conference on Data Mining. IEEE; 2008. p. 293–302. Available from: http://compbio.cs.umn.edu/wp-content/uploads/2017/10/HyperGene.pdf.

30. Tian Z, Hwang T, Kuang R. A hypergraph-based learning algorithm for classifying gene expression and arrayCGH data with prior knowledge. Bioinformatics. 2009;25(21):2831–2838. doi:10.1093/bioinformatics/btp467.

31. Chatr-Aryamontri A, Oughtred R, Boucher L, Rust J, Chang C, Kolas NK, et al. The BioGRID interaction database: 2017 update. Nucleic acids research. 2017;45(D1):D369–D379.

32. Sayama H. Estimation of Laplacian spectra of direct and strong product graphs. Discrete Applied Mathematics. 2016;205:160–170.

33. Horn RA, Horn RA, Johnson CR. Topics in matrix analysis. Cambridge university press; 1994.

34. Li Z, Zhang W, Huang RS, Kuang R. Learning a Low-Rank Tensor of Pharmacogenomic Multi-relations from Biomedical Networks. In: 2019 IEEE International Conference on Data Mining (ICDM). IEEE; 2019. p. 409–418.

35. Lee DD, Seung HS. Algorithms for non-negative matrix factorization. In: Advances in neural information processing systems; 2001. p. 556–562.

36. Smith S, Ravindran N, Sidiropoulos ND, Karypis G. SPLATT: Efficient and parallel sparse tensor-matrix multiplication. In: 2015 IEEE International Parallel and Distributed Processing Symposium. IEEE; 2015. p. 61–70.

37. Kim H, Park H. Sparse non-negative matrix factorizations via alternating non-negativity-constrained least squares for microarray data analysis. Bioinformatics. 2007;23(12):1495–1502.

38. Lim H, Poleksic A, Yao Y, Tong H, He D, Zhuang L, et al. Large-scale off-target identification using fast and accurate dual regularized one-class collaborative filtering and its application to drug repurposing. PLoS computational biology. 2016;12(10):e1005135.

39. Camacho C, Coulouris G, Avagyan V, Ma N, Papadopoulos J, Bealer K, et al. BLAST+: architecture and applications. BMC bioinformatics. 2009;10(1):421.

40. Gu Q, Zhou J, Ding C. Collaborative filtering: Weighted nonnegative matrix factorization incorporating user and item graphs. In: Proceedings of the 2010 SIAM international conference on data mining. SIAM; 2010. p. 199–210.

41. Xu C, Su Z. Identification of cell types from single-cell transcriptomes using a novel clustering method. Bioinformatics. 2015;31(12):1974–1980.

42. Satija R, Farrell JA, Gennert D, Schier AF, Regev A. Spatial reconstruction of single-cell gene expression data. Nature biotechnology. 2015;33(5):495–502.

43. Tomioka R, Suzuki T, Hayashi K, Kashima H. Statistical performance of convex tensor decomposition. In: Advances in neural information processing systems; 2011. p. 972–980.

44. Narita A, Hayashi K, Tomioka R, Kashima H. Tensor factorization using auxiliary information. Data Mining and Knowledge Discovery. 2012;25(2):298–324.

45. Yu G, Wang LG, Han Y, He QY. clusterProfiler: an R package for comparing biological themes among gene clusters. Omics: a journal of integrative biology. 2012;16(5):284–287.

46. Crowley SD, Gurley SB, Oliverio MI, Pazmino AK, Griffiths R, Flannery PJ, et al. Distinct roles for the kidney and systemic tissues in blood pressure regulation by the renin-angiotensin system. The Journal of clinical investigation. 2005;115(4):1092–1099.

47. Coffman TM, Crowley SD. Kidney in hypertension: guyton redux. Hypertension. 2008;51(4):811–816.

48. Verouti SN, Boscardin E, Hummler E, Frateschi S. Regulation of blood pressure and renal function by NCC and ENaC: lessons from genetically engineered mice. Current opinion in pharmacology. 2015;21:60–72.

49. Brown D, Wagner CA. Molecular mechanisms of acid-base sensing by the kidney. Journal of the American Society of Nephrology. 2012;23(5):774–780.

50. Yanase H, Takebe K, Nio-Kobayashi J, Takahashi-Iwanaga H, Iwanaga T. Cellular expression of a sodium-dependent monocarboxylate transporter (Slc5a8) and the MCT family in the mouse kidney. Histochemistry and cell biology. 2008;130(5):957–966.

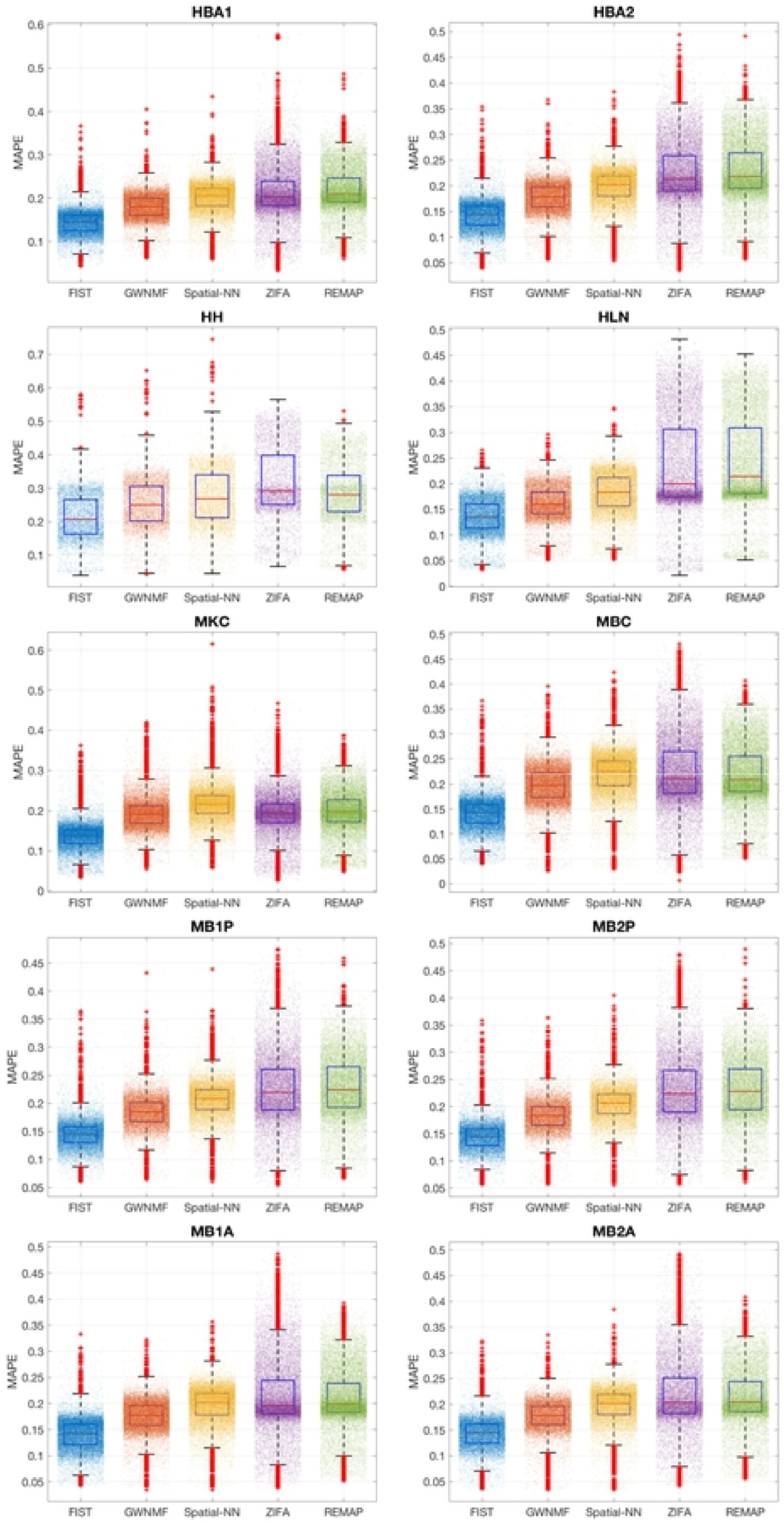
51. Nagamori S, Wiriyasermkul P, Guarch ME, Okuyama H, Nakagomi S, Tadagaki K, et al. Novel cystine transporter in renal proximal tubule identified as a missing partner of cystinuria-related plasma membrane protein rBAT/SLC3A1. Proceedings of the National Academy of Sciences. 2016;113(3):775–780.

52. Zalups RK. Organic anion transport and action of γ-glutamyl transpeptidase in kidney linked mechanistically to renal tubular uptake of inorganic mercury. Toxicology and applied pharmacology. 1995;132(2):289–298.

53. Anzai N, Jutabha P, Enomoto A, Yokoyama H, Nonoguchi H, Hirata T, et al. Functional characterization of rat organic anion transporter 5 (Slc22a19) at the apical membrane of renal proximal tubules. Journal of Pharmacology and Experimental Therapeutics. 2005;315(2):534–544.

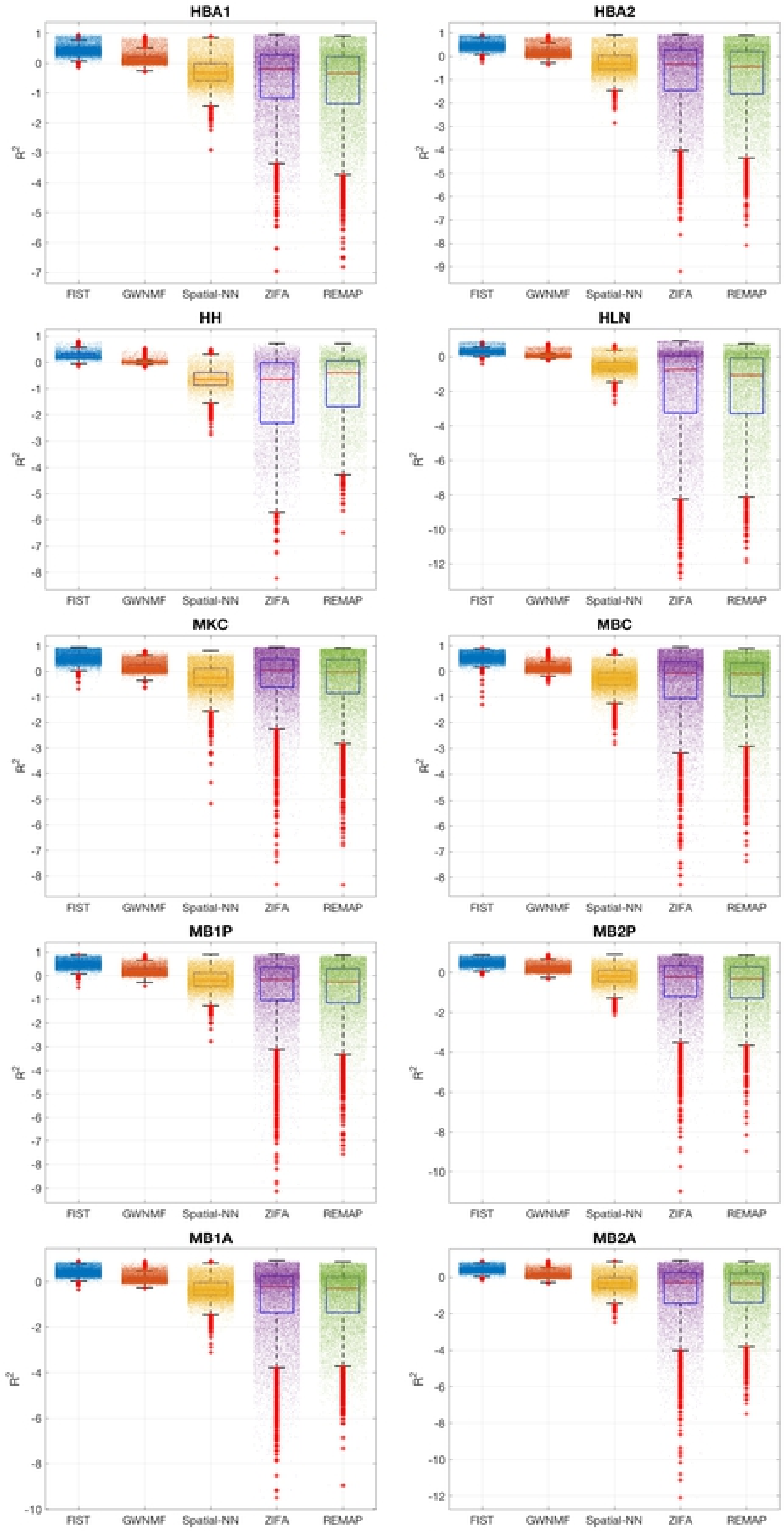
54. Tojo A, Sekine T, Nakajima N, Hosoyamada M, Kanai Y, Kimura K, et al. Immunohistochemical localization of multispecific renal organic anion transporter 1 in rat kidney. Journal of the American Society of Nephrology. 1999;10(3):464–471.

55. Hwang JS, Park EY, Kim W, Yang CW, Kim J. Expression of OAT1 and OAT3 in differentiating proximal tubules of the mouse kidney. Histology and histopathology. 2010;.



Gene-wise imputation performance by MAPE. The performances on the imputations of each gene are shown as box plots. The MAPE of every gene slice is denoted by one dot. The performance of each method is shown in each colored box plot.

Figure S1



Gene-wise imputation performance by R^2. The performances on the imputations of each gene are shown as box plots. The R^2 of every gene slice is denoted by one dot. The performance of each method is shown in each colored box plot.

Figure S2