FIST-nD: A tool for n-dimensional spatial transcriptomics data imputation via graph-regularized tensor completion

Thomas Karl Atkins[1], Tianci Song[1], Rui Kuang[*1]

[1]Department of Computer Science and Engineering, University of Minnesota Twin Cities

Abstract

Functional interpretation of spatial transcriptomics data usually requires non-trivial preprocessing steps and other supporting data in the analysis due to the high sparsity and incompleteness of spatial RNA profiling, especially in 3D constructions. As a solution, we present a new software tool FIST-nD, Fast Imputation of Spatially-resolved transcriptomes by graph-regularized Tensor completion in n-Dimensions for imputing 3D as well as 2D spatial transcriptomics data. FIST-nD is implemented based on a novel graph-regularized tensor decomposition method, which imputes spatial gene expression data using 4-way high-order tensor structure and relations in spatial and gene functional graphs. The implementation, accelerated by GPU or multicore parallel computing, can efficiently impute high-resolution 3D spatial transcriptomics data within a few minutes. The experiments on three 3D Spatial Transcriptomics datasets and one 3D high-resolution Stereo-seq dataset confirm the high accuracy of the imputation by FIST-nD and demonstrate that the imputed spatial transcriptomes provide a more complete gene expression landscape for downstream analyses such as spatial gene expression clustering and visualizations.

Background

To understand how the composition and arrangement of cells in tissues contribute to their functional role has long been one of the primary goals in biological research (Vesalius, 1543). While RNA sequencing (RNA-seq) and lately developed single-cell RNA sequencing (scRNA-seq) allow researchers to quantify the transcriptome in single cells as well as bulk tissue samples, the spatial variation in expression patterns is still needed for a complete mapping of the cells' organization in a tissue (Miller et al., 2021). More recently developed in-situ

*Email: kuang@umn.edu

capturing (ISC) methods perform RNA sequencing of the whole transcriptome with positional barcodes in a spatial genomic array aligned to locations on the tissue such as Spatial Transcriptomics (ST) (Ståhl et al., 2016) (commercialized as 10x Genomics Visium (10x Genomics, 2019)), higher resolution Slide-seq (Rodriques et al., 2019), high-definition spatial transcriptomics (HDST) (Vickovic et al., 2019) and Spatio-temporal enhanced REsolution omics-sequencing (Stereo-seq) (Chen et al., 2022). These ISC-based methods capture two dimensional gene expression profiles from a tissue section attached to a slide, where RNA is hybridized to primers that contain both a spatial barcode and unique molecular identifiers (UMIs) that, when prepared into a DNA library, allows researchers to determine both the location and the transcript where a read originated. From this, a "map" of RNA expression across a tissue slice can be constructed to give the levels of RNA expressions at every spot on the slide.

Though current spatial transcriptomics platforms collect data from a two-dimensional tissue slice, parallel adjacent tissue slices can be combined to create a three dimensional gene expression dataset (Asp et al., 2019; Ortiz et al., 2020; Vickovic et al., 2022; Wang et al., 2022). Several methods have been developed to align these parallel slices in 3D space, either using histological imaging or similarity of expression profiles (Bergenstråhle et al., 2020; Zeira et al., 2022). The 3D construction is critical for visualizing and understanding gene expression patterns in complete spatial arrangement in 3D such as those in Drosophila embryos (Wang et al., 2022). Due to the technical limitations of ISC such as low RNA capture efficiency (Asp et al., 2020) and failure in RNA fixation and permeabilization in some tissue regions due to various possible issues in preparing tissue sections and the array (Li et al., 2021), integrative analysis of the 2D profiling of multiple adjacent slices of a tissue to reconstruct the 3D gene expressions in the complete real 3D space is still a new challenge. To facilitate the analysis of the 3D gene expressions, we present FIST-nD (Fast Imputation of Spatially-resolved transcriptomes by Graph-regularized Tensor decomposition in n-Dimensions), a novel method of imputing the 3D data that incorporates spatial information and gene functional relations in a protein-protein interaction (PPI) network. FIST-nD is an extension of our earlier ST 2D imputation method, FIST (Li et al., 2021) with an extended 3D formulation and implementation to be applied on stacked slice data. While original FIST completes a 3-way sparse tensor in genes (p-mode) and the (x, y) spatial coordinates (x-mode and y-mode). FIST-nD is a generalization to include the 3D spatial coordinates (x, y, z) and the genes in a 4-way sparse tensor. FIST-nD discretizes the input spatial gene expression data with continuous spatial coordinates into a tensor by a voxelization process. Then, graph-regularized tensor completion is applied to the 4D sparse tensor regularized by a spatial graph and protein-protein interaction network (PPI). The optimization problem is solved by a Python script (optimized for GPU computing) or a MATLAB script (optimized for multicore performance). Finally, the output tensor decomposition can be directly analyzed for understanding the 3D gene expression patterns or alternatively, converted to original spatial expressions through a process of multilinear interpolation. Experiments on four publicly available datasets finds that FIST-nD is more accurate than other methods, and that the imputation with FIST-nD leads to increased performance on downstream analysis tasks. We also provide an Python command line implementation of the method that is both easy-to-use and compares favorably to competing methods in terms of running time.

Many software pipelines have been developed for analysis of spatial transcriptomic data.

STUtility is a pipeline that takes 10X Visium data and standardizes alignment of sections with annotation and visualization features (Bergenstråhle et al., 2020). Squidpy integrates multiple types of spatial data to compute a spatial graph for analysis of spatial patterns (Palla et al., 2022). Giotto is a tool that clusters spatial data and performs enrichment analyses (Dries et al., 2020). However, none of these tools adopts the imputation strategy prior to their functional analysis, limiting the information gained from the spatial data. Because modern high-throughput and transcriptome-wide spatial and single-cell sequencing technologies generate zero-inflated data, many of the zero values in the final count matrices are not reflective of actual biological zeros, but rather arise as a result of sampling biases (Jiang et al., 2022). Imputation is a strategy that has the potential to improve the functional analysis of the data with more truthful RNA expressions (Li et al., 2021), which is more eminent in 3D spatial gene expressions. So far, most existing imputation algorithms have primarily focused on imputation and dimension reduction of single-cell RNA-seq (scRNA-seq) data, rather than spatial transcriptome data. One such method, DeepImpute, imputes scRNA-seq data by splitting genes into related blocks for imputation with a sub-neural network (Arisdakessian et al., 2019). Another popular imputation technique for scRNA-seq data, MAGIC constructs a Markov distance matrix between cells, which can be used to impute the original data with a smoothing step (van Dijk et al., 2018). However, these scRNA-seq methods do not utilize information about spatial relationships between cells, and as such ignore the spatial context beneficial to the imputation. While the previous FIST program has been developed to perform imputation limited 2D ST data (Li et al., 2021). Some more recent deep learning methods focus on learning the embedding of spatial gene expressions rather than being optimized for the imputation task (Dong and Zhang, 2022).

Results

Overview of FIST-nD

FIST-nD (Fast Imputation of Spatially-resolved transcriptomes by graph-regularized Tensor completion in n-Dimensions) is an extension of our earlier ST imputation method, FIST, which minimizes an objective function of graph-regularized tensor completion over the gene expression tensor and a tensor product graph of the spatial chain graphs of each spatial axis and the PPI network (Li et al., 2021). The complete mathematical definitions and the multiplicative updating algorithm are reviewed in the **Methods** section. FIST-nD generalizes FIST for n-dimensional tensor and the matched higher-order graph such that stacked 2D slice spatial gene expression data can be analyzed in 3D. From here on out, we refer to FIST-nD as FIST as well, as FIST-nD applied to two-dimensional data is identical to the original method. The FIST-nD pipeline is detailed in Figure 1. First, FIST takes in continuous spatial gene expression data as input and performs a voxelization process to convert the data into a discrete tensor where the first dimension corresponds to genes and all subsequent dimensions correspond to the spatial dimensions (x,y,z) of the data. In the tensor completion step, the objective function is constructed to minimize the distance between the original and imputed tensors. The objective function is also regularized by a product graph of the spatial chain graphs and protein-protein interaction network (PPI). The optimiza-

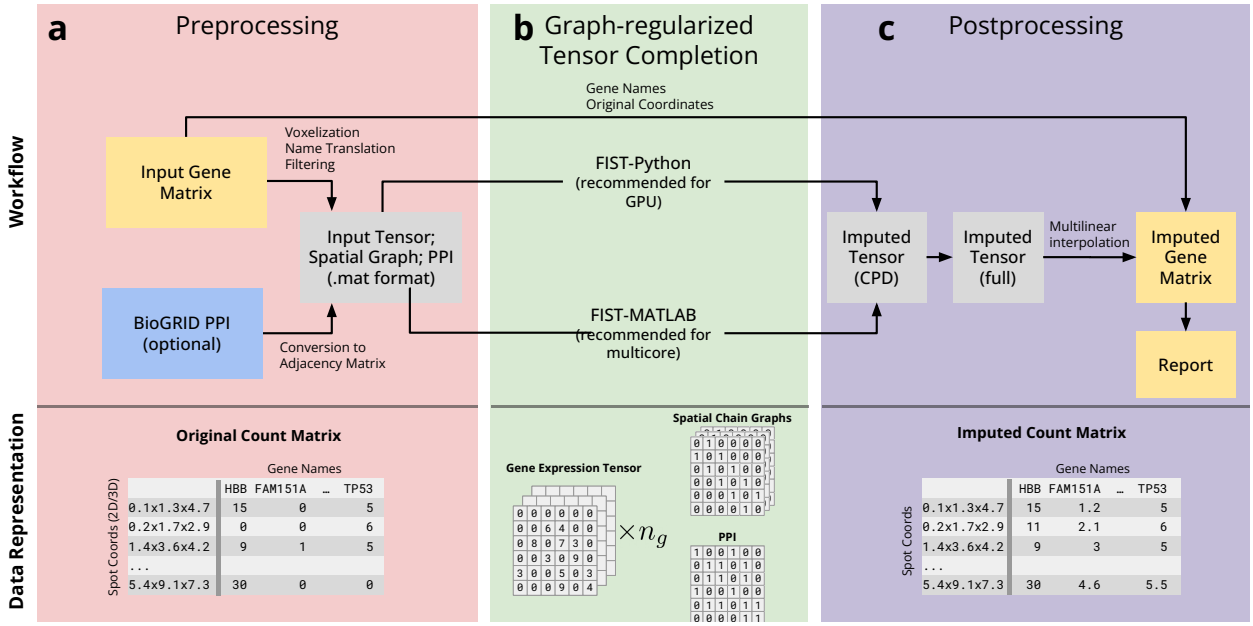

Figure 1: **Overview of FIST-nD.** (a) The input data is a matrix of multi-dimensional array of gene expression read counts indexed by the spatial coordinates of the spots. The preprocessing step filters and voxelizes the input data, converting it to a tensor representation by aggregating the spots. This step also constructs the spatial chain graph and a PPI graph for the next step. (b) The tensor and the graphs generated in the preprocessing step are then fed into the FIST algorithm for graph-regularized tensor completion. FIST-nD provides two potions, a Python implementation (for use with a GPU system) and a MATLAB implementation (for use with a multicore system). (c) The postprocessing step converts the CPD from of the tensor to the original data format through multilinear interpolation or a full tensor representation of the imputed gene expressions.

tion algorithm is implemented by either a Python script (optimized for GPU computing) or a MATLAB script (optimized for multicore performance). Finally, the learned canonical polyadic decomposition (CPD) form of the tensor can be used to construct the full tensor for conversion to the original data format through a process of multilinear interpolation.

Summary of Datasets

In the experiments, we focus on testing FIST-nD on four publicly available 3D spatial gene expression datasets. The first dataset is an expression atlas of the adult mouse brain (AMB) (Ortiz et al., 2020). The dataset was created by registering tissue slices sequenced using ST into a tissue atlas. The second dataset measures gene expression in the developing human heart at 6.5 PCW (DHH)(Asp et al., 2019). The third dataset measures gene expression in the human rheumatoid arthritis synovium, prepared in the same way (RAS) (Vickovic et al., 2022). The three datasets were similarly prepared through ST sequencing of parallel 2-dimensional slices, assembled to form three-dimensional data. The fourth dataset measures gene expression in a *Drosophila* embryo 16-18 hours after egg-laying (Wang et al., 2022). This dataset was created using slices of the embryo measured by high-resolution spatial sequencing technology, which was binned before being similarly arranged in the stacked slice format. The detail of the four datasets are described in **Methods** section and Table 3.

FIST Outperforms Competing Methods under Cross-validation

In the five-fold cross-validation, we held out a subset of the test entries in the input matrix and measure how accurately FIST with a PPI network or a diagonal gene graph reconstructs the missing entries. This cross-validation technique roughly approximates the performance of predicting unseen data to evaluation the generalization. We compared FIST with the other imputation algorithms, including spatial nearest-neighbor (SNN), STAGATE (Dong and Zhang, 2022), DeepImpute (DI) (Arisdakessian et al., 2019) and Markov Affinity-based Graph Imputation of Cells (MAGIC) (van Dijk et al., 2018) as outlined in the **Methods** section. The SNN imputation algorithm was implemented by the authors, and we used the provided Python packages to experiment with the other methods. As a major component of the graph regularization is the PPI network, we performed an ablation experiment to test whether the inclusion of the network significantly contributed to the accuracy of our results. To do this, in addition to running FIST with an organism-specific PPI downloaded from BioGrid, we also run FIST with a 'diagonal' gene graph, represented in adjacency matrix form as the identity matrix. For each method, we measure error by absolute error (MAE), symmetric mean absolute percentage error (SMAPE), and root mean squared error (RMSE), averaged across all genes defined in the **Methods** section.

Figure 2 shows the performance of FIST versus other imputation methods under the 5-fold entry-wise cross-validation. It is clear that FIST with a PPI network or a diagonal gene graph has a lower error than every other method tested, under all the three metrics. In particular, incorporation of a PPI network in the regularization also significantly improved the imputation performance by the introduction of the gene-gene relations. To further establish the significance, we also computed 95% highest-density intervals (HDIs) for the posterior distribution of the difference between the mean MAE, SMAPE, and RMSE of FIST between

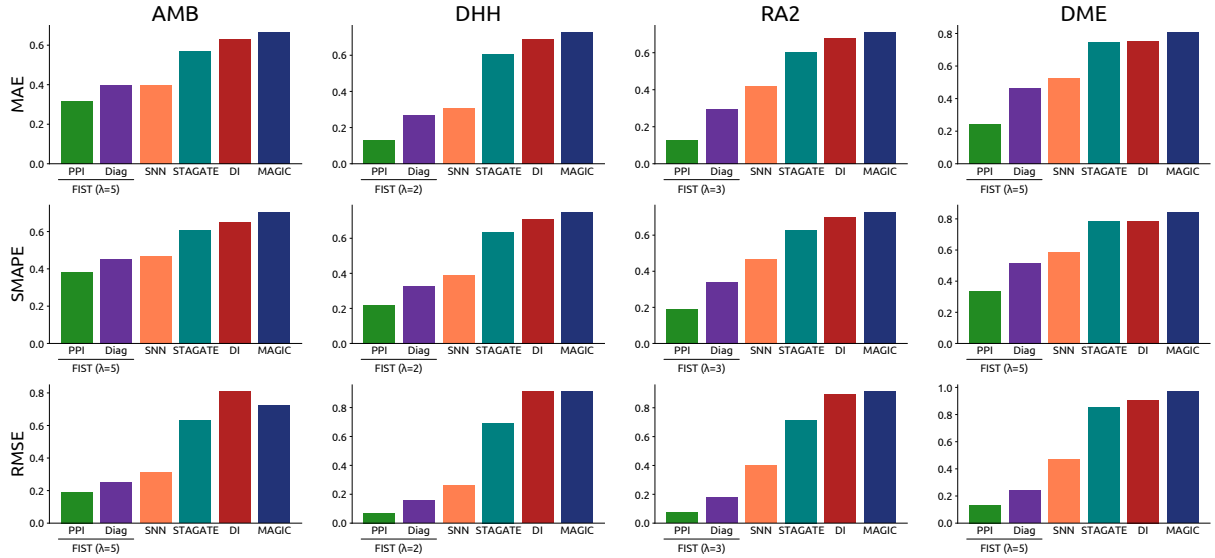

Figure 2: **Imputation performance by five-fold cross-validation.** The imputation performance was measured for each method in the four datasets. In the plots, the height of each bar corresponds to the mean value of the metric for that method, averaged over all genes. More detailed pairwise comparisons between FIST and the other methods are given in Supplementary Tables S1-S3, and Supplementary Figure S4.

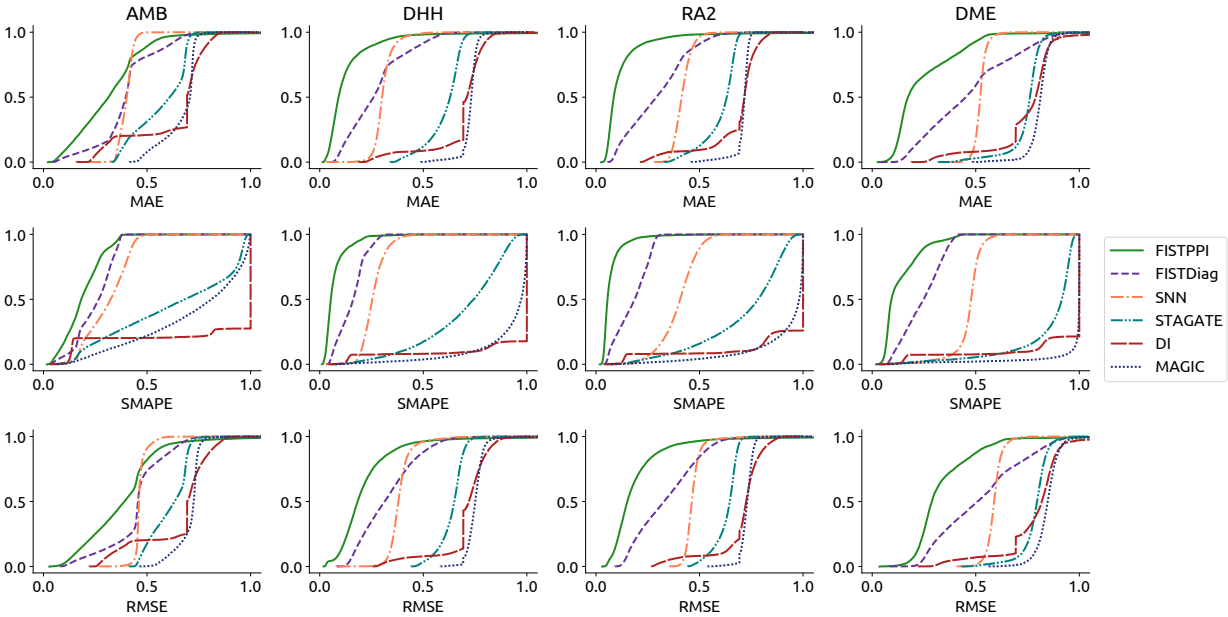

Figure 3: **CDF of gene-wise error metrics for all the methods tested.** In the plots, the x-axis is a threshold of the error measure and the y-axis is the percentage of the genes with an error lower than the threshold. Thus, the height of the distribution indicate a better performance by having a larger fraction of errors below the threhold.

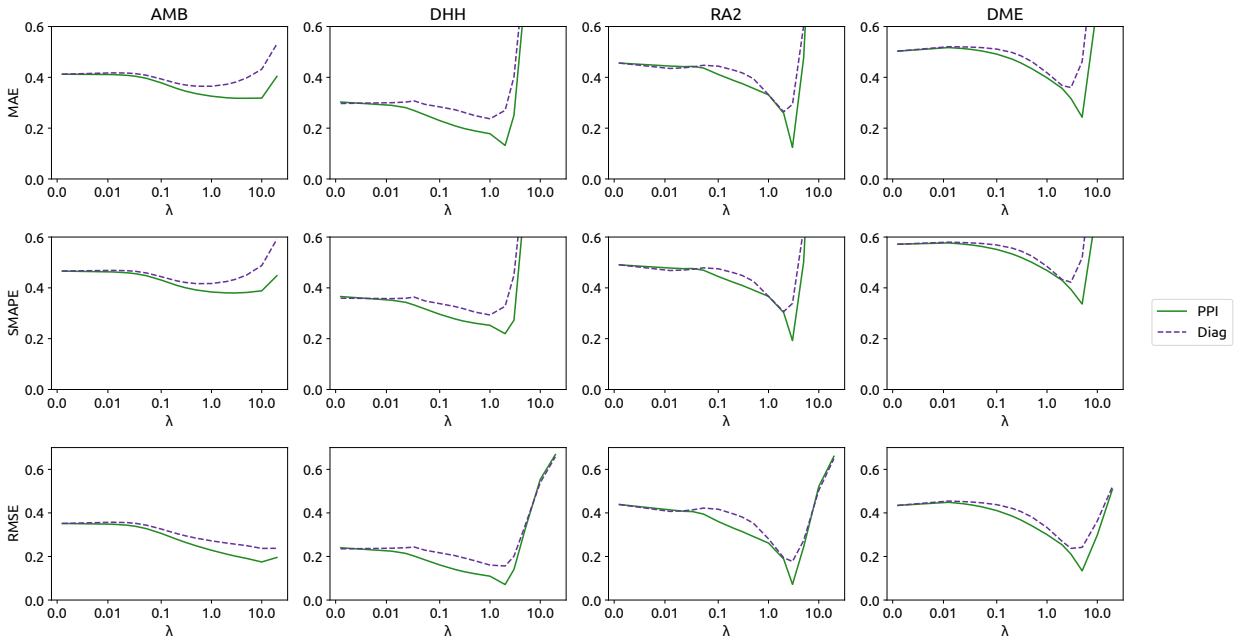

Figure 4: **Tuning hyperparameter** λ. The plot shows the error of FIST on each dataset under cross-validation over a set of λ parameters.

FIST-nD with PPI against FIST-nD with diagonal gene graph and the other methods using the the BEST (Bayesian Estimation Supersedes the t-Test) method (Kruschke, 2013). The results can be found in Supplementary Tables S1-S3 with plots of the posterior distributions in Supplementary Figure S4.

To further characterize the error profiles of the imputation methods, Figure 3 shows a cumulative density function for the error metrics for each gene for each dataset. The errors of FIST are consistently smaller than the other imputation method. In particular, we see that FIST does not have the concentration of high SMAPEs that we see in STAGATE, DeepImpute, and MAGIC. To examine comparisons between the datasets, we plot these data in two dimensional histograms in Supplementary Figures S1-S3, where we compare FIST with diagonal gene graph to all other methods. From these data, we conclude the FIST is the most accurate algorithm tested for imputation of stacked ST data.

FIST Hyperparameter Tuning

To choose a hyperparameter λ for the regularization term of the FIST objective function, we ran a search over the set $\{0, 0.0.01, 0.02, 0.03, 0.05, 0.1, 0.2, 0.3, 0.5, 1, 2, 3, 5, 10, 20\}$. All searches were run with a rank of 150, which is close the optimal rank reported in (Li et al., 2021). Plots of the error metrics of FIST models for each λ are shown in Figure 4. On all four datasets, there exists a global minimum for all metrics somewhere between $\lambda = 1$ and $\lambda = 10$, suggesting that choosing a lambda between those two values is optimal for a variety of 3D spatial transcriptomics datasets.

Furthermore, because we have the additional voxelization step of converting the continu-

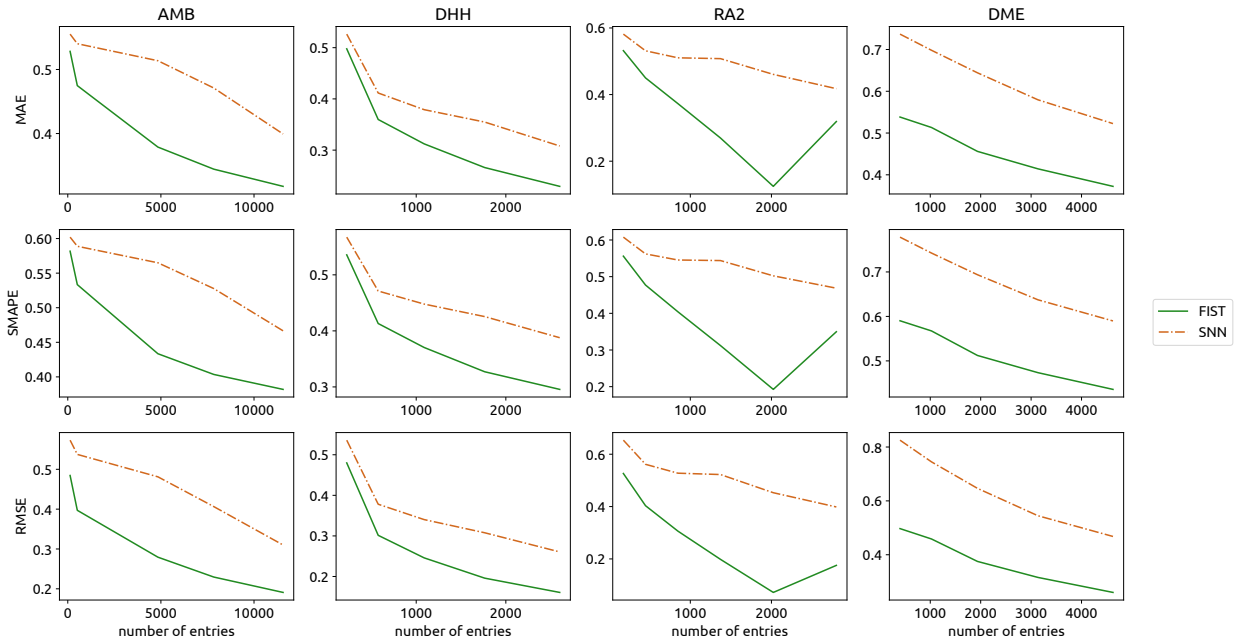

Figure 5: **Variable binning size.** The plots show the error of FIST and spatial nearest-neighbor (SNN) on each dataset for different sized input tensors.

ous data to a discrete tensor, we test the effect of the tensor size, which can be chosen by the user, on the output quality. Figure 5 shows the performance of FIST and SNN on tensors of different sizes, as measured by the product of the spatial dimensions of the tensor. For all datasets, we see consistent improvements in the performance of FIST as the size of the tensor increases. Because the resolution of the RA2 slices is significantly lower than the resolution of the slices of the other datasets, when the tensor size exceeds 2000 entries, SNN performs worse imputation than smaller tensor constructions. This observation suggests that increasing the tensor size beyond the resolution of the slices is potentially detrimental. However, on the vast majority of metrics, we see improvement that suggests that users should choose as large a tensor size that their computational setup allows.

Clustering Performance on GO Terms

While the previous results demonstrate that FIST accurately imputes ST data, we must also validate the utility of FIST on downstream analysis tasks. Here, we consider the task of clustering genes based on their spatial profile. We performed a clustering and enrichment analysis on the data before and after imputation with and without a PPI network. For each dataset, we employ k-means clustering 10 times on gene profiles (with spatial features (spots) reduced to 1000 through PCA). We then compute the fraction of clusters enriched to some cutoff. We repeat this process twice, once with all genes, and once with only the genes that have at least one interaction in the PPI with more functional relevance.

The results of the clustering analysis are shown in Figure 6. The plots show the fraction of enriched clusters as the provided p-value cutoff. For the human and mouse datasets,

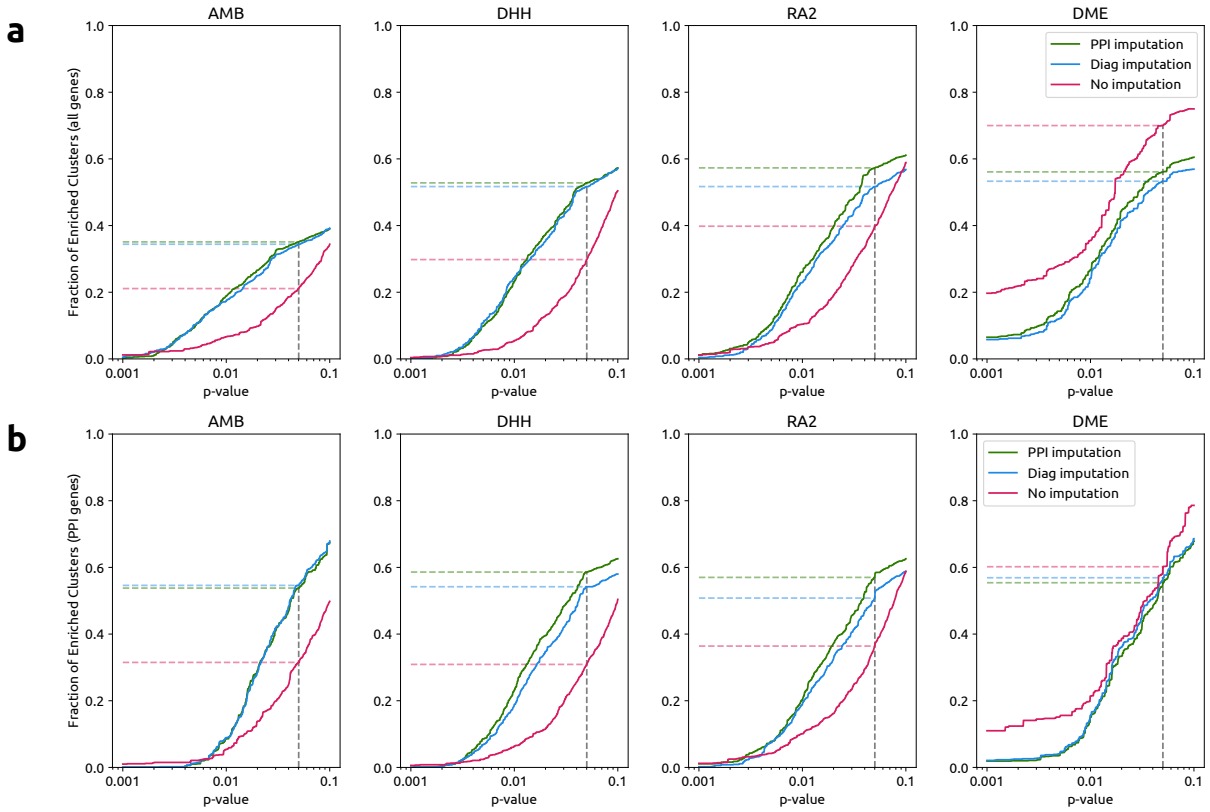

Figure 6: **Gene ontology enrichment analysis.** For each dataset, the plot shows the fraction of enriched clusters on pre- and post-imputed data. For a given p-value, the plots correspond to the number of clusters that had an enriched term with a smaller p-value. The fraction of enriched clusters at $p = 0.05$ is annotated by the dashed lines. (a) Clustering performed on all genes in the dataset. (b) Clustering performed only on genes containing at least one connection in the PPI network.

the data post-imputation has more significantly enriched clusters at almost every p-value between $p = 0.001$ and $p = 0.1$, suggesting that FIST model imputes data with consistence with the biological functions. Interestingly, a lower fraction of clusters enriched is observed in clustering on all genes on the *Drosophila* dataset (Figure 6a). However, this difference significantly shrinks when we restrict our clustering to genes with a connection in the PPI network (Figure 6b). This suggests that the low performance may be indicative of lesser annotation quality in non-human organisms such as Drosophila, rather than the overall assumption.

One possible concern is the model's ability to recover functional relevant signals without the functional information by inclusion of the PPI network, which carries functional information. To mitigate the concern in the evaluation, we also tested the model's imputation performance when the PPI was removed. It is observed that while including a PPI does increase the fraction of enriched clusters, only including a diagonal PPI still leads to increased performance over the raw data in three of the four cases. Therefore, it is evident that FIST recovers gene expressions for more effective functional analysis, and with a well annotated PPI network such as human PPI network and mouse PPI network, both the accuracy of imputation and functional analysis can be improved.

Timing

Finally, we compared the runtime of FIST and the other tested algorithms. For each method with a canonical implementation (FIST, DeepImpute, and MAGIC), we time the method on the Minnesota Supercomputing Institute hardware, using 8 cores, 100g memory, and an NVIDIA A100 GPU. STAGATE was not tested because it requires a version of Tensorflow which was incompatible with installed GPU drivers, leading to unfair comparisons. Based on our empirical observation, STATGATE is significantly smaller than all the other methods, and thus a controlled comparison is unnecessary.

The results of the timing are shown in Figure 7, with dataset size plotted on the x-axis, and running time plotted on the y-axis (both axes are log-scaled). We see that while there is significant variation of the sizes of the different datasets, FIST has running time comparable to the other two methods that also scale well with the dataset size. Notably, on the largest dataset AMB, FIST pipeline completes the imputation process within a few minutes. The results suggest the high computational efficiency of the FIST algorithm even if it operates on a 4-way tensor.

Discussion

The newly developing "stacked slice" 3D spatial transcriptomics dataset is a promising way through which we can understand the spatial organization of tissues. Yet because these data are so sparse, we propose an imputation method, FIST, to improve downstream analyses.

Here, we provide compelling evidence that FIST-nD is the most accurate method for imputing 3D spatial trancriptomic data created using the "stacked slices" experimental technique. We found that FIST has significantly lower MAE, SMAPE, and RMSE when compared to other spatial and single-cell imputation methods. Furthermore, we demonstrated

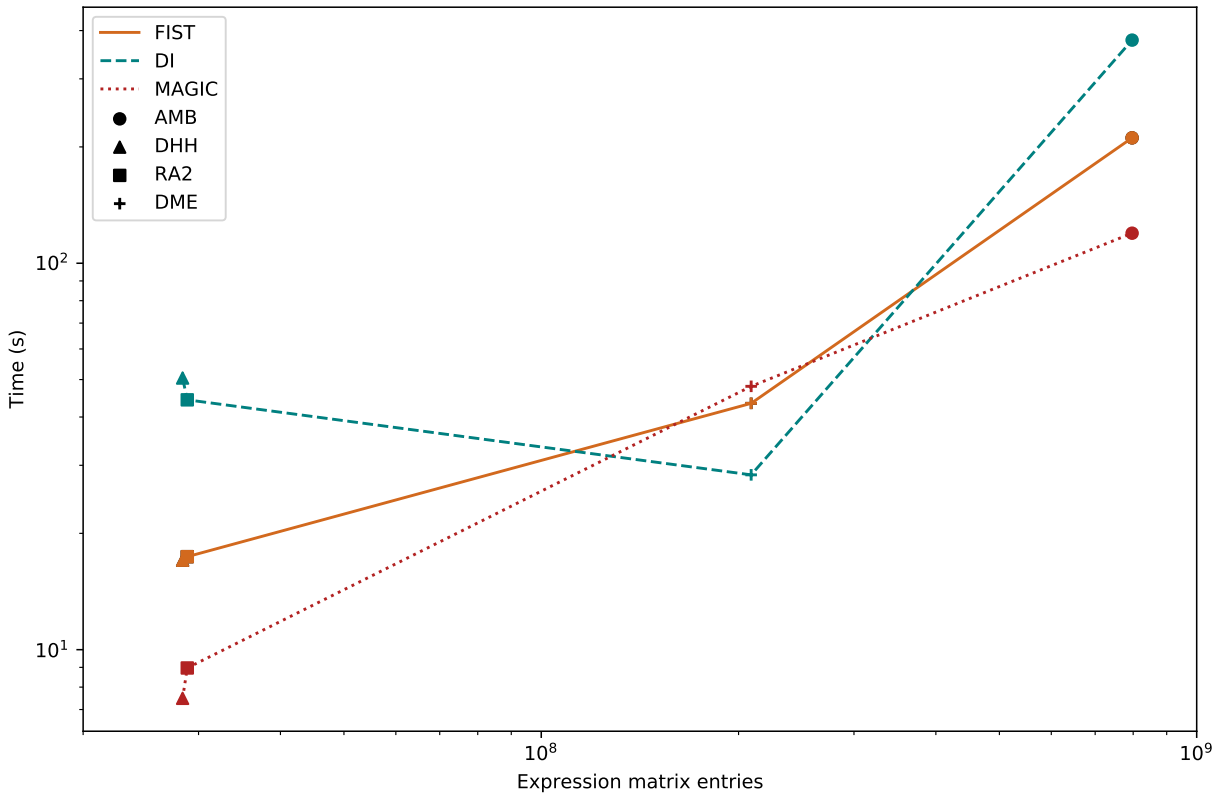

Figure 7: **Runtime results of FIST, DeepImpute, and MAGIC.** The x-axis plots the input dataset size (measured as number of spots times number of genes). The y-axis shows time (in seconds) for the method to run using 8 cores, 100g memory, and an NVIDIA A100 GPU. Color and line type correspond to the methods, while marker shape corresponds to the datasets.

FIST	FIST-nD	Definition
2	N	Number of spatial dimensions
$\{x, y\}$	$\{x_1, ..., x_N\}$	Spatial dimensions
n_i	n_i	Size of spatial dimension x_i
n_p	n_p	Size of gene dimension
$\mathcal{T} \in \mathbb{R}_+^{n_p \times n_x \times n_y}$	$\mathcal{T} \in \mathbb{R}_+^{n_p \times n_1 \times ... \times n_N}$	Original spatial gene expression tensor
$\hat{\mathcal{T}} \in \mathbb{R}_+^{n_p \times n_x \times n_y}$	$\hat{\mathcal{T}} \in \mathbb{R}_+^{n_p \times n_1 \times ... \times n_N}$	Imputed spatial gene expression tensor
$\mathcal{M} \in \mathbb{R}_+^{n_p \times n_x \times n_y}$	$\mathcal{M} \in \mathbb{R}_+^{n_p \times n_1 \times ... \times n_N}$	Binary mask tensor
$\hat{A}_i \in \mathbb{R}^{n_i \times r}$	$\hat{A}_i \in \mathbb{R}^{n_i \times r}$	CPD component matrix i of $\hat{\mathcal{T}}$
$\mathfrak{L}(p, x, y)$	$\mathfrak{L}(p, x_1, ..., x_N)$	Laplacian of PPI and spatial chain graphs
$\mathcal{M}$	$\mathcal{M}$	mask matrix selecting none-zero entries in $\mathcal{T}$
λ	λ	Regularization hyperparameter

Table 1: Notations

the utility of incorporating a PPI into the model by performing an ablation experiment.

The method not only accurately imputes 3D spatial transcriptomics data, but also improves the quality of downstream analyses. By analyzing the fraction of enriched clusters before and after imputation, we see that the imputation process leads to higher enrichment for a large range of p-values for the majority of the datasets tested. Furthermore, to demonstrate that this is not the result of the inclusion of PPI, we show that the result still holds when the PPI network is replaced by a diagonal gene graph in the model. Thus, we have shown that FIST is both accurate and useful.

In addition to a compelling model, we also present an implementation that is fast and easy-to-use. Timing results using an NVIDIA A100 GPU show that FIST is comparable to other popular scRNA-seq methods in terms of running time, while delivering significantly more accurate imputations. As FIST is implemented as a command-line tool, it also is easy to use.

While we validated the method for downstream analysis using enrichment tests, an experiment comparing spatial clustering performance on labelled datasets could further validate the utility of the model. Finally, the current iteration of FIST utilizes a voxelization step to take advantage of the tensor decomposition model. Future versions that impute on the raw data instead of the voxelized data could potentially see better performance. As long as the stacked slice 3D spatial transcriptomic technique that help us understand spatial arrangement of tissues produces sparse data, imputation methods such as FIST cannot fail to be of use.

Methods

Mathematical Formulation

We begin with a review of the FIST (Fast Imputation of Spatially-Resolved transcriptomes by graph-regularized Tensor completion) method for imputation of two-dimensional ST data (Li et al., 2021), before introducing the generalization of the algorithm to FIST-nD. Notations

are provided in Table 1.

Review of FIST Algorithm

Denote the spots, which are arranged on a grid of size (n_x, n_y), and their associated spatial gene expressions as a tensor $\mathcal{T} \in \mathbb{R}_+^{n_p \times n_x \times n_y}$ where n_p denotes the number of genes, so that $\mathcal{T}_{ijk}$ is the number of reads of gene i at spot (j, k) on the slide. The learning task is to approximate $\mathcal{T}$ with $\hat{\mathcal{T}}$ represented as a sum of outer products of vectors in the Canonical Polyadic Decomposition (CPD) form (Kolda and Bader, 2009) as follows,

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

where r is the rank of the CPD form of $\hat{\mathcal{T}}$ and $\otimes$ denotes the vector outer product. The objective function minimizes the difference between the observed and the imputed tensor under a smoothness constraint defined on the graph Laplacian of a Cartesian product of a protein-protein interaction network (PPI) with the spatial chain graphs in the x and y dimensions, denoted by $\mathfrak{L}(p, x, y)$ as follows, where $\circledast$ represents the Hadamard product,

$$\min_{\hat{A}_i, i \in \{x,y,p\}} \frac{1}{2} ||\mathcal{M} \circledast (\mathcal{T} - \hat{\mathcal{T}})||_{\mathcal{F}}^2 + \frac{\lambda}{2} \mathbf{vec}(\hat{\mathcal{T}})^T \mathfrak{L}(p, x, y) \mathbf{vec}(\hat{\mathcal{T}}) \tag{2}$$

To minimize the objective function, we utilize the following multiplicative update rule, denoting the Khatri-Rao product as $\odot$:

$$\left[\hat{A}_i\right]_{a,b} \leftarrow \left[\hat{A}_i\right]_{a,b} \left(\frac{\left[\frac{\partial \mathcal{J}_1}{\partial \hat{A}_i}\right]_{a,b}^{-} + \lambda \left[\frac{\partial \mathcal{J}_2}{\partial \hat{A}_i}\right]_{a,b}^{-}}{\left[\frac{\partial \mathcal{J}_1}{\partial \hat{A}_i}\right]_{a,b}^{+} + \lambda \left[\frac{\partial \mathcal{J}_2}{\partial \hat{A}_i}\right]_{a,b}^{+}} \right). \tag{3}$$

where

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

Each of the above terms can be similarly defined for update of $\hat{A}_x$ and $\hat{A}_y$.

Generalization to FIST-nD

The equations from the FIST algorithm to any number of spatial dimensions can be generalized following the derivations and the formulation in Li et al. (2019, 2021). Equation 1 becomes

$$\hat{\mathcal{T}} = \sum_{i=1}^{r} [\hat{A}_p]_{:,i} \otimes \left(\bigotimes_{j=1}^{N} [\hat{A}_j]_{:,i} \right). \tag{4}$$

Furthermore, the objective function (Equation 2) becomes generalized as follows,

$$\min_{\hat{A}_i, i \in \{x_1, ..., x_n, p\}} \frac{1}{2} ||\mathcal{M} \circledast (\mathcal{T} - \hat{\mathcal{T}})||_{\mathcal{F}}^2 + \frac{\lambda}{2} \mathbf{vec}(\hat{\mathcal{T}})^T \mathfrak{L}(x_1, ..., x_n, p) \mathbf{vec}(\hat{\mathcal{T}}) \tag{5}$$

Extending the update rule is less straightforward, but the overall form factor remains the same. The generalized update rule is

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

where

$$\begin{aligned}
\left[\frac{\partial \mathcal{J}_1}{\partial \hat{A}_i}\right]^{+} &= (\mathcal{M} \circledast \hat{\mathcal{T}}) \left(\bigodot_{j \neq i} \hat{A}_j \right) \\
\left[\frac{\partial \mathcal{J}_1}{\partial \hat{A}_i}\right]^{-} &= (\mathcal{M} \circledast \mathcal{T}) \left(\bigodot_{j \neq i} \hat{A}_j \right) \\
\left[\frac{\partial \mathcal{J}_2}{\partial \hat{A}_i}\right]^{+} &= \hat{A}_i \left[\sum_{j \neq i} \Theta_j^D \circledast \left(\bigcirc\!\!\!\!\!\circledast_{k \neq i,j} \Phi_k \right) \right] + D_i \hat{A}_i \left(\bigcirc\!\!\!\!\!\circledast_{j \neq i} \Phi_j \right) \\
\left[\frac{\partial \mathcal{J}_2}{\partial \hat{A}_i}\right]^{-} &= \hat{A}_i \left[\sum_{j \neq i} \Theta_j^W \circledast \left(\bigcirc\!\!\!\!\!\circledast_{k \neq i,j} \Phi_k \right) \right] + W_i \hat{A}_i \left(\bigcirc\!\!\!\!\!\circledast_{j \neq i} \Phi_j \right).
\end{aligned}$$

With these modifications, the FIST-nD algorithm is generalized to any n-dimensional tensor completion in theory and can be used to perform imputation on 3D spatial RNA-seq data in practice.

FIST-nD Pipeline

The overall pipeline of FIST-nD is given in 1. Each of the steps and the components are described in detail below.

Use of FIST-nD Pipeline

FIST-nD is implemented as a Python package, available through the PyPI package manager, or from source at `https://github.com/kuanglab/FIST-nD` pip. Installation and running instructions and dependencies are provided on the GitHub page. The package is implemented as a command-line tool with options for inputs, outputs, PPI, hyperparameters, etc.

Input format

The pipeline starts with a multidimensional array of gene expression indexed by the spatial coordinates of the spots after mapping as input, where the first three (or two) columns contain the spatial coordinates and the remaining columns contain expression values of the genes. The format of the input is the same as the output provided by many existing spatial RNA-seq data processing software tools (Navarro et al., 2017) for easy integration with other pipelines.

Preprocessing

Before converting the data to a tensor format, a few preprocessing steps need to be applied. FIST-nD attempts to determine if one of the axes (regards as z mode) is composed of discrete slices so that these slices are represented as consecutive tensor indices in that dimension. The tool optionally allows the user to rotate the data using a PCA transformation in order to have the dimensions with the highest variance line up with the tensor axes for better performance after voxelization.

Voxelization

If the user is working with 2-dimensional 10X Visium data, the data is already arranged in a spatially uniform grid for tensor completion. However, if we have data with spots in continuous space, we must represent these data as a N-way tensor for operating on a tensor in the CPD form. The process of converting points in continuous space to a tensor (akin to voxelization) for the FIST input is as follows, shown graphically in Supplementary Figure S5. Voxel boundaries are determined by taking evenly spaced quantiles of the distribution of spots along each spatial dimension. In the binning process, let s index the spots as (x_s, y_s, z_s) with expression vectors g_i, we calculate $\mathcal{T}_{:,ijk}$, each (i, j, k)-th gene fiber in the tensor by the following metric:

$$\mathcal{T}_{:,ijk} = \frac{1}{\sum_s I_{ijk}(s)} \sum_s g_s I_{ijk}(s) \tag{7}$$

$$I_{ijk}(s) = \begin{cases} 1, & q_x(\frac{i}{n_x}) \leq d_x \leq q_x(\frac{i+1}{n_x}), q_y(\frac{j}{n_y}) \leq d_y \leq q_y(\frac{j+1}{n_y}), q_z(\frac{k}{n_z}) \leq d_z \leq q_z(\frac{k+1}{n_z}) \\ 0, & \text{else,} \end{cases} \tag{8}$$

where $q_d(.)$ is the quantile function for dimension $d \in \{x, y, z\}$. Here, $I_{ijk}(s)$ is an indicator if spot s is a member of voxel (i, j, k). After the voexlization, any voxel with less than three total read counts is set to zero and all the genes with 8 or less total counts after this step

Organism	Unique Genes	Total Interactions	Mean Interactions per Gene
H. sapiens	25744	783618	30.439
M. musculus	16531	88295	5.341
D. melanogaster	10372	83763	8.076

Table 2: Statistics for the PPI networks used for the model. Total interactions is calculated as the size of the set of all interacting pairs of genes.

are also removed from the tensor. Finally, a log-transform is then applied to the voxelized tensor.

Processing PPI Network

We optionally allow the user to specify a PPI network as input for the graph regularization, which is downloaded from BioGRID (`https://downloads.thebiogrid.org/BioGRID/Release-Archive/BIOGRID-4.4.201/`)(Oughtred et al., 2021). BioGRID contains PPI networks for a variety of organisms, represented as a list of pairs of interacting genes. Our software converts this representation to an adjacency-matrix format, using the MyGene annotation service to match the gene names provided by the user in the original count matrix to their official gene symbols (Wu et al., 2013). Summary statistics for all the PPI networks used in the corss-validation experiments are provided in Table 2.

If no PPI is provided by the user, FIST-nD by default uses a dummy diagonal graph, where every gene interacts only with itself. Product with the default diagonal graph leads to identical product graph for every gene for utilizing the spatial information only. This default construction of FIST-nD is still significantly better than other competing imputation methods in the comparisons in the Results section, although the imputation result is much better when a real PPI is used.

Stopping Criteria

The optimization algorithm stops when one of the three criteria is reached. First, the user can specify a maximum number of iterations for the algorithm to run. Second, the user can specify a stopping criteria such that if the square root of the ratio of percent squared difference between the CPD forms between iterations is less than this value, the algorithm will stop. Finally, the user is able to specify that a set of indices should be held out as a validation set, in which case the algorithm will stop if MAE and MAPE are increasing, and R^2 is decreasing.

Multilinear Interpolation

After the completion of the optimization, the CPD form of the imputed tensor can be used to construct the full tensor or utilized for conversion back to the original format, a matrix of gene expressions by spots. A multilinear interpolation algorithm is used for the conversion. If (x, y, z) is in voxel (i, j, k) with corner (v_i, v_j, v_k), then

$$f(x,y,z) = \sum_{i' \in \{i,i+1\}} \sum_{j' \in \{j,j+1\}} \sum_{k' \in \{k,k+1\}} |x - v_{i'}| \cdot |y - v_{j'}| \cdot |z - v_{k'}| \cdot \mathcal{T}(i',j',k'). \quad (9)$$

This function considers all neighboring voxels of the spot to allows further variation in read counts to occur within voxels, reducing the importance of binning size as a hyperparameter.

Outputs and Visualizations

The final output of the imputed data is in the same format as the original input data, along with an optional report of the imputation. This report includes parameter details, plots of validation error, a list of the genes with the highest variance across all spots in the post-imputation data, and side-by-side plots of selected genes pre-imputation and post-imputation. Furthermore, we provide a separate script for the side-by-side visualization of genes pre- and post-imputation in an interactive 3D plot.

Evaluation and Cross-validation

To test the methods, we performed 5-fold entry-wise crossvalidation held out a subset of the entries (20%) in the original gene expression matrix. This modified input matrix with 80% training entries would be run through our pipeline, and the imputed entries were compared to the original held-out test entries. The read count data were log-transformed. For each gene g, we evaluated the following three metrics:

- Mean absolute error (MAE): $\frac{1}{n_g}\sum_{i=0}^{n_g} |\mathcal{T}_{g,i} - \hat{\mathcal{T}}_{g,i}|$.
- Symmetric mean absolute percentage error (SMAPE): $\frac{1}{n_g}\sum_{i=0}^{n_g} \frac{|\mathcal{T}_{g,i} - \hat{\mathcal{T}}_{g,i}|}{|\mathcal{T}_{g,i}| + |\hat{\mathcal{T}}_{g,i}|}$.
- Root mean squared error (RMSE): $\sqrt{\frac{1}{n_g}\sum_{i=0}^{n_g} (\mathcal{T}_{g,i} - \hat{\mathcal{T}}_{g,i})^2}$.

Then, the overall performance is evaluated by the mean and variance of these metrics over the genes. Note that for all the metrics, nonzero entries in the original tensor $\mathcal{T}$ were considered as unknowns and not used in the calculating the errors.

Spatial Gene Expression Clustering and Enrichment Analysis

The post imputation matrix was defined as $\bar{T} = T \odot M + \hat{T} \odot (1 - M)$, where M is the binary mask matrix, the pre-imputation matrix was given by T and the output of FIST is given by $\hat{T}$. Clustering was done by first reducing the number of features (spots) for each gene down to 1000 with PCA and then applying k-means with $k = 100$ using the scikit-learn package (Pedregosa et al., 2011). This was repeated 10 times with different random seed initializations to reduce variability by local minima (Macqueen, 1967). Enrichment analysis on each cluster was performed with the enrichGO function in the R clusterProfiler package with FDR correction applied (Wu et al., 2021). The size of 15 by 15 by 15 and the value of λ found to be optimal in cross-validation for each specific dataset was used for the imputation by FIST-nD models.

Dataset	# of Genes	# of Spots	Slices	Density
AMB	23371	34103	40	0.1892
DHH	19336	1467	9	0.1092
RAS	12333	2338	7	0.1456
DME	13668	15295	16	0.0464

Table 3: Summary of 3D spatial transcriptomics datasets. Density is calculated as number of nonzero entries in the gene expression matrix divided by the total number of entries in the gene expression matrix.

Related Methods for Comparison

FIST-nD was compared with several existing imputation methods by the same five cross-validation procedure described below,

- **SpatialNN:** For a baseline method that also incorporates spatial information, we implemented a spatial single nearest-neighbor algorithm that takes the voxelized tensor as input and imputes missing tensor entries based on the closest tensor entry (as measured by Euclidian distance between tensor indices).
- **STAGATE (Spatially resolved Transcriptomics with an Adaptive Graph ATtention auto-Encoder):** STAGATE implements a graph attention-based autoencoder for spatial transcriptomic embedding and clustering (Dong and Zhang, 2022). From the associated decoder, we recover the model's imputations of 3D stacked slice ST data.
- **DeepImpute:** DeepImpute imputes scRNA-seq data by dividing genes into correlated classes, each of which are imputed by a sub-neural network (Arisdakessian et al., 2019). However, this model does not take into account spatial information, being originally developed for single-cell data.
- **MAGIC (Markov Affinity-based Graph Imputation of Cells):** Another scRNA-seq imputation technique, MAGIC imputes single cell RNA-seq data by constructing a Markov distance matrix for random walk (van Dijk et al., 2018). This method has consistently ranked among the top single cell RNA-seq data imputation methods (Hou et al., 2020).

By comparing FIST-nD to scRNA-seq imputation techniques that do not utilize spatial information and the recent spatial gene expression embedding learning method, we demonstrate the utility of FIST as a tool for accurate imputation of spatial transcriptomics data.

3D Spatial Transcriptomics Datasets

Four 3D spatial gene expression datasets including three ST datasets and one high-resolution Stereo-seq dataset are analyzed in the experiments.

- Adult Mouse Brain (AMB): This dataset was collected from a sample of adult mouse brain (AMB) tissue (Ortiz et al., 2020). The authors took 40 parallel slices along the anteroposterior axis of the tissue, which were then registered to produce three dimensional coordinates for each spot. The data were downloaded from the authors' website (`https://www.molecularatlas.org/download-data`). The meta table and the raw expression matrix were downloaded from the processed data section, which were combined to create the input file.

- Developing Human Heart (DHH): This dataset was collected from a sample of developing human heart (DHH) tissue taken 6.5 weeks post-conception (Asp et al., 2019). The authors obtained 9 parallel slices of the tissue sample with 105-219 spots per slice covered by the tissue (of 1,007 maximum possible). The data were downloaded from the authors' GitHub repo (`https://github.com/MickanAsp/Developmental_heart`). The data were then processed to combine the atlas and gene tables into one table in the format described, using the `spot.pos` column as an index.

- Rheumatoid Arthritis Synovium (RA2): This dataset was collected from a rheumatoid arthritis synovium tissue sample (Vickovic et al., 2022). From the six patient samples presented in the original study, we took the dataset from the second patient (RA2), as it contains 7 slices, with the most number of slices among all six samples. The data were downloaded from the repository Single Cell Portal (`https://singlecell.broadinstitute.org/single_cell/study/SCP1414/3dst-ra`). Because the separate slices were provided as separate files, the location and position information for each slice was combined, with all the slices being concatenated into a single dataset.

- Drosophila Melanogaster Embryo (DME): This dataset was collected from a Drosophila embryo at 16-18 hours after egg-laying and contains 16 slices (Wang et al., 2022). The data were collected via Stereo-seq, and binned into 20x20 bins of DNBs. The data are available at `https://db.cngb.org/stomics/flysta3d/download.html`.The dataset was rotated using a PCA transfomation of the x and y axes so that the major axis of the ellipsoid was parallel to the x axis, so that the shape fit more efficiently for binning.

Availability and Requirements

The data used for the project are publicly available datasets published elsewhere. They can be found at the links in the **3D Spatial Transcriptomics Datasets** section.

The tool is available from PyPI at `https://pypi.org/project/fistnd/` or from source at `https://github.com/kuanglab/FIST-nD`.

Competing interests

The authors declare that they have no competing interests.

Funding

This research work is supported by a grant from the National Science Foundations, USA (NSF BIO DBI-IIBR 2042159).

Acknowledgements

The authors acknowledge the Minnesota Supercomputing Institute (MSI) at the University of Minnesota for providing resources that contributed to the research results reported within this paper (URL: `http://www.msi.umn.edu`).

References

Python package index - pypi. URL `https://pypi.org/`.

10x Genomics. Visium spatial transcriptomics, www.10xgenomics.com/products/spatial-gene-expression, 2019. URL `https://www.10xgenomics.com/products/spatial-gene-expression`.

C. Arisdakessian, O. Poirion, B. Yunits, X. Zhu, and L. X. Garmire. DeepImpute: an accurate, fast, and scalable deep neural network method to impute single-cell RNA-seq data. *Genome Biol*, 20(1):211, 10 2019.

M. Asp, S. Giacomello, L. Larsson, C. Wu, D. Fürth, X. Qian, E. Wärdell, J. Custodio, J. Reimegård, F. Salmén, C. Österholm, P. L. Ståhl, E. Sundström, E. Åkesson, O. Bergmann, M. Bienko, A. Månsson-Broberg, M. Nilsson, C. Sylvén, and J. Lundeberg. A spatiotemporal organ-wide gene expression and cell atlas of the developing human heart. *Cell*, 179(7):1647–1660.e19, Dec. 2019. doi: 10.1016/j.cell.2019.11.025.

M. Asp, J. Bergenstråhle, and J. Lundeberg. Spatially resolved transcriptomes—next generation tools for tissue exploration. *BioEssays*, 42(10):1900221, 2020. doi: https://doi.org/10.1002/bies.201900221. URL `https://onlinelibrary.wiley.com/doi/abs/10.1002/bies.201900221`.

J. Bergenstråhle, L. Larsson, and J. Lundeberg. Seamless integration of image and molecular analysis for spatial transcriptomics workflows. *BMC Genomics*, 21(1):482, 7 2020.

A. Chen, S. Liao, M. Cheng, K. Ma, L. Wu, Y. Lai, X. Qiu, J. Yang, J. Xu, S. Hao, X. Wang, H. Lu, X. Chen, X. Liu, X. Huang, Z. Li, Y. Hong, Y. Jiang, J. Peng, S. Liu, M. Shen, C. Liu, Q. Li, Y. Yuan, X. Wei, H. Zheng, W. Feng, Z. Wang, Y. Liu, Z. Wang, Y. Yang, H. Xiang, L. Han, B. Qin, P. Guo, G. Lai, P. Muñoz-Cánoves, P. H. Maxwell, J. P. Thiery, Q. F. Wu, F. Zhao, B. Chen, M. Li, X. Dai, S. Wang, H. Kuang, J. Hui, L. Wang, J. F. Fei, O. Wang, X. Wei, H. Lu, B. Wang, S. Liu, Y. Gu, M. Ni, W. Zhang, F. Mu, Y. Yin, H. Yang, M. Lisby, R. J. Cornall, J. Mulder, M. Uhlén, M. A. Esteban, Y. Li, L. Liu, X. Xu, and J. Wang. Spatiotemporal transcriptomic atlas of mouse organogenesis using DNA nanoball-patterned arrays. *Cell*, 185(10):1777–1792, 05 2022.

K. Dong and S. Zhang. Deciphering spatial domains from spatially resolved transcriptomics with an adaptive graph attention auto-encoder. *Nat Commun*, 13(1):1739, 04 2022.

R. Dries, Q. Zhu, R. Dong, C.-H. L. Eng, H. Li, K. Liu, Y. Fu, T. Zhao, A. Sarkar, R. E. George, N. Pierson, L. Cai, and G.-C. Yuan. Giotto, a toolbox for integrative analysis and visualization of spatial expression data. *bioRxiv*, 2020. doi: 10.1101/701680. URL `https://doi.org/10.1101/701680`.

W. Hou, Z. Ji, H. Ji, and S. C. Hicks. A systematic evaluation of single-cell RNA-sequencing imputation methods. *Genome Biol*, 21(1):218, 08 2020.

R. Jiang, T. Sun, D. Song, and J. J. Li. Statistics or biology: the zero-inflation controversy about scrna-seq data. *Genome Biology*, 23(1):31, 1 2022. ISSN 1474-760X. doi: 10.1186/s13059-022-02601-5. URL `https://doi.org/10.1186/s13059-022-02601-5`.

T. G. Kolda and B. W. Bader. Tensor decompositions and applications. *SIAM Review*, 51(3): 455–500, 2009. doi: 10.1137/07070111X. URL `https://doi.org/10.1137/07070111X`.

J. K. Kruschke. Bayesian estimation supersedes the t test. *J Exp Psychol Gen*, 142(2): 573–603, 05 2013.

Z. Li, W. Zhang, R. S. Huang, and R. Kuang. Learning a low-rank tensor of pharmacogenomic multi-relations from biomedical networks. In *2019 IEEE International Conference on Data Mining (ICDM)*, pages 409–418, 2019. doi: 10.1109/ICDM.2019.00051.

Z. Li, T. Song, J. Yong, and R. Kuang. Imputation of spatially-resolved transcriptomes by graph-regularized tensor completion. *PLOS Computational Biology*, 17(4):1–25, 04 2021. doi: 10.1371/journal.pcbi.1008218. URL `https://doi.org/10.1371/journal.pcbi.1008218`.

J. Macqueen. Some methods for classification and analysis of multivariate observations. In *In 5-th Berkeley Symposium on Mathematical Statistics and Probability*, pages 281–297, 1967.

B. F. Miller, D. Bambah-Mukku, C. Dulac, X. Zhuang, and J. Fan. Characterizing spatial gene expression heterogeneity in spatially resolved single-cell transcriptomic data with nonuniform cellular densities. *Genome research*, 31(10):1843–1855, 10 2021. ISSN 1549-5469. doi: 10.1101/gr.271288.120. URL `https://doi.org/10.1101/gr.271288.120`. 34035045[pmid].

J. F. Navarro, J. Sjöstrand, F. Salmén, J. Lundeberg, and P. L. Ståhl. ST Pipeline: an automated pipeline for spatial mapping of unique transcripts. *Bioinformatics*, 33(16): 2591–2593, 04 2017. ISSN 1367-4803. doi: 10.1093/bioinformatics/btx211. URL `https://doi.org/10.1093/bioinformatics/btx211`.

C. Ortiz, J. F. Navarro, A. Jurek, A. Märtin, J. Lundeberg, and K. Meletis. Molecular atlas of the adult mouse brain. *Science Advances*, 6(26), 2020. doi: 10.1126/sciadv.abb3446.

R. Oughtred, J. Rust, C. Chang, B. J. Breitkreutz, C. Stark, A. Willems, L. Boucher, G. Leung, N. Kolas, F. Zhang, S. Dolma, J. Coulombe-Huntington, A. Chatr-Aryamontri, K. Dolinski, and M. Tyers. The BioGRID database: A comprehensive biomedical resource of curated protein, genetic, and chemical interactions. *Protein Sci*, 30(1):187–200, 01 2021.

G. Palla, H. Spitzer, M. Klein, D. Fischer, A. C. Schaar, L. B. Kuemmerle, S. Rybakov, I. L. Ibarra, O. Holmberg, I. Virshup, M. Lotfollahi, S. Richter, and F. J. Theis. Squidpy: a scalable framework for spatial omics analysis. *Nat Methods*, 19(2):171–178, 02 2022.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

S. G. Rodriques, R. R. Stickels, A. Goeva, C. A. Martin, E. Murray, C. R. Vanderburg, J. Welch, L. M. Chen, F. Chen, and E. Z. Macosko. Slide-seq: A scalable technology for measuring genome-wide expression at high spatial resolution. *Science*, 363(6434):1463–1467, 2019.

P. L. Ståhl, F. Salmén, S. Vickovic, A. Lundmark, J. F. Navarro, J. Magnusson, S. Giacomello, M. Asp, J. O. Westholm, M. Huss, A. Mollbrink, S. Linnarsson, S. Codeluppi, Åke Borg, F. Pontén, P. I. Costea, P. Sahlén, J. Mulder, O. Bergmann, J. Lundeberg, and J. Frisén. Visualization and analysis of gene expression in tissue sections by spatial transcriptomics. *Science*, 353(6294):78–82, 2016. doi: 10.1126/science.aaf2403. URL `https://www.science.org/doi/abs/10.1126/science.aaf2403`.

D. van Dijk, R. Sharma, J. Nainys, K. Yim, P. Kathail, A. J. Carr, C. Burdziak, K. R. Moon, C. L. Chaffer, D. Pattabiraman, B. Bierie, L. Mazutis, G. Wolf, S. Krishnaswamy, and D. Pe'er. Recovering Gene Interactions from Single-Cell Data Using Data Diffusion. *Cell*, 174(3):716–729, 07 2018.

A. Vesalius. *De Humani Corporis Fabrica Libri Septem.* 1543.

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

S. Vickovic, D. Schapiro, K. Carlberg, B. Lötstedt, L. Larsson, F. Hildebrandt, M. Korotkova, A. H. Hensvold, A. I. Catrina, P. K. Sorger, V. Malmström, A. Regev, and P. L. Ståhl. Three-dimensional spatial transcriptomics uncovers cell type localizations in the human rheumatoid arthritis synovium. *Commun Biol*, 5(1):129, 02 2022.

M. Wang, Q. Hu, T. Lv, Y. Wang, Q. Lan, R. Xiang, Z. Tu, Y. Wei, K. Han, C. Shi, J. Guo, C. Liu, T. Yang, W. Du, Y. An, M. Cheng, J. Xu, H. Lu, W. Li, S. Zhang, A. Chen, W. Chen, Y. Li, X. Wang, X. Xu, Y. Hu, and L. Liu. High-resolution 3D spatiotemporal transcriptomic maps of developing Drosophila embryos and larvae. *Dev Cell*, 57(10):1271–1283, 05 2022.

C. Wu, I. Macleod, and A. I. Su. BioGPS and MyGene.info: organizing online, gene-centric information. *Nucleic Acids Res*, 41(Database issue):D561–565, 01 2013.

T. Wu, E. Hu, S. Xu, M. Chen, P. Guo, Z. Dai, T. Feng, L. Zhou, W. Tang, L. Zhan, X. Fu, S. Liu, X. Bo, and G. Yu. clusterProfiler 4.0: A universal enrichment tool for interpreting omics data. *Innovation (N Y)*, 2(3):100141, 8 2021.

R. Zeira, M. Land, A. Strzalkowski, and B. J. Raphael. Alignment and integration of spatial transcriptomics data. *Nat Methods*, 19(5):567–575, 05 2022.

Supplementary Information

Dataset	FISTDiag	SNN	STAGATE	DeepImpute	MAGIC
AMB	(0.087,0.092)	(0.091,0.095)	(0.264,0.270)	(0.327,0.336)	(0.400,0.405)
DHH	(0.155,0.160)	(0.208,0.210)	(0.533,0.536)	(0.643,0.646)	(0.639,0.641)
RA2	(0.197,0.203)	(0.325,0.327)	(0.540,0.543)	(0.631,0.634)	(0.630,0.631)
DME	(0.229,0.240)	(0.325,0.332)	(0.571,0.576)	(0.605,0.611)	(0.620,0.626)

Table S1: 95% highest density intervals for posterior distribution of the difference between mean MAE for genes imputed by FIST versus competing methods.

Dataset	FISTDiag	SNN	STAGATE	DeepImpute	MAGIC
AMB	(0.057,0.061)	(0.117,0.121)	(0.436,0.445)	(0.563,0.574)	(0.525,0.535)
DHH	(0.100,0.103)	(0.199,0.201)	(0.681,0.687)	(0.946,0.947)	(0.913,0.915)
RA2	(0.123,0.127)	(0.348,0.351)	(0.714,0.720)	(0.943,0.944)	(0.918,0.920)
DME	(0.115,0.121)	(0.374,0.377)	(0.808,0.812)	(0.896,0.899)	(0.895,0.897)

Table S2: 95% highest density intervals for posterior distribution of the difference between mean SMAPE for genes imputed by FIST versus competing methods.

Dataset	FISTDiag	SNN	STAGATE	DeepImpute	MAGIC
AMB	(0.085,0.090)	(0.082,0.087)	(0.238,0.243)	(0.299,0.308)	(0.345,0.351)
DHH	(0.123,0.129)	(0.194,0.197)	(0.461,0.465)	(0.565,0.569)	(0.564,0.567)
RA2	(0.169,0.176)	(0.308,0.311)	(0.490,0.493)	(0.575,0.579)	(0.575,0.578)
DME	(0.194,0.204)	(0.289,0.294)	(0.500,0.504)	(0.533,0.537)	(0.547,0.552)

Table S3: 95% highest density intervals for posterior distribution of the difference between mean RMSE for genes imputed by FIST versus competing methods.

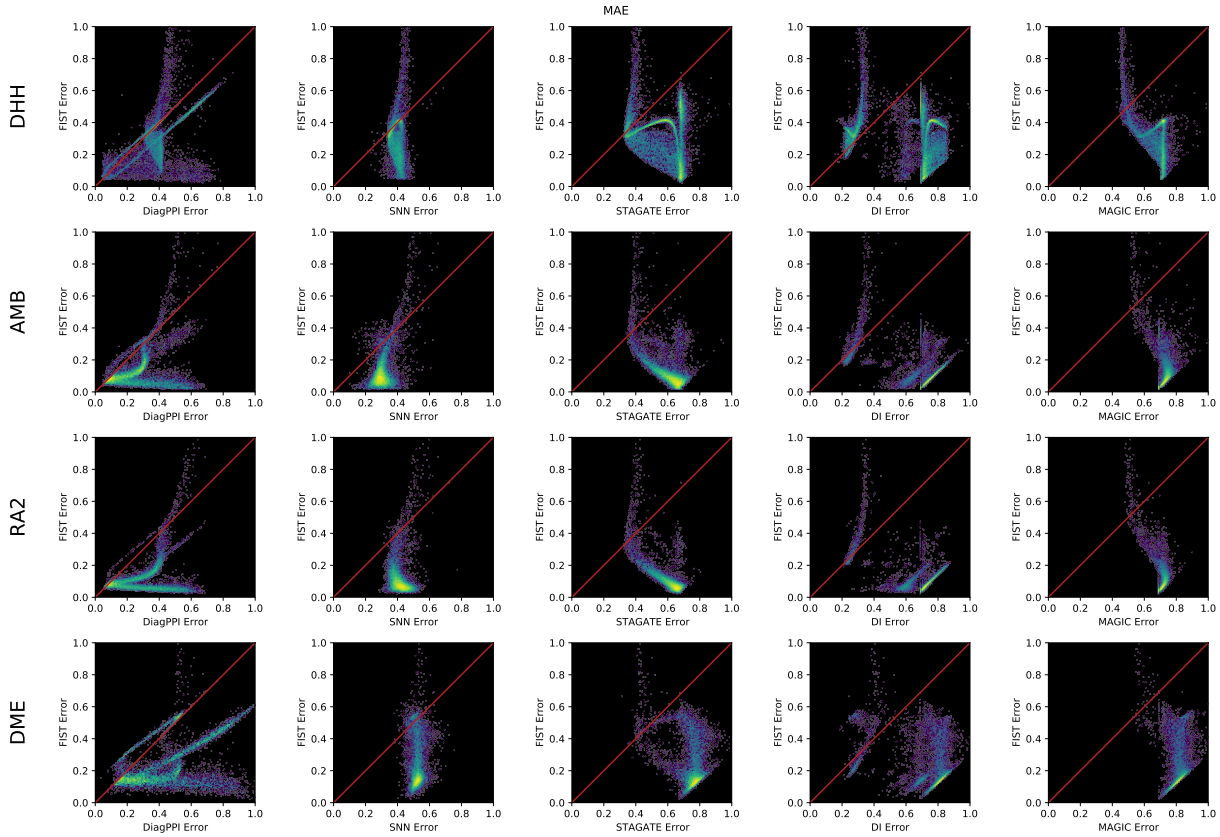

Figure S1: 2D histogram of MAEs for each gene for FIST against each method.

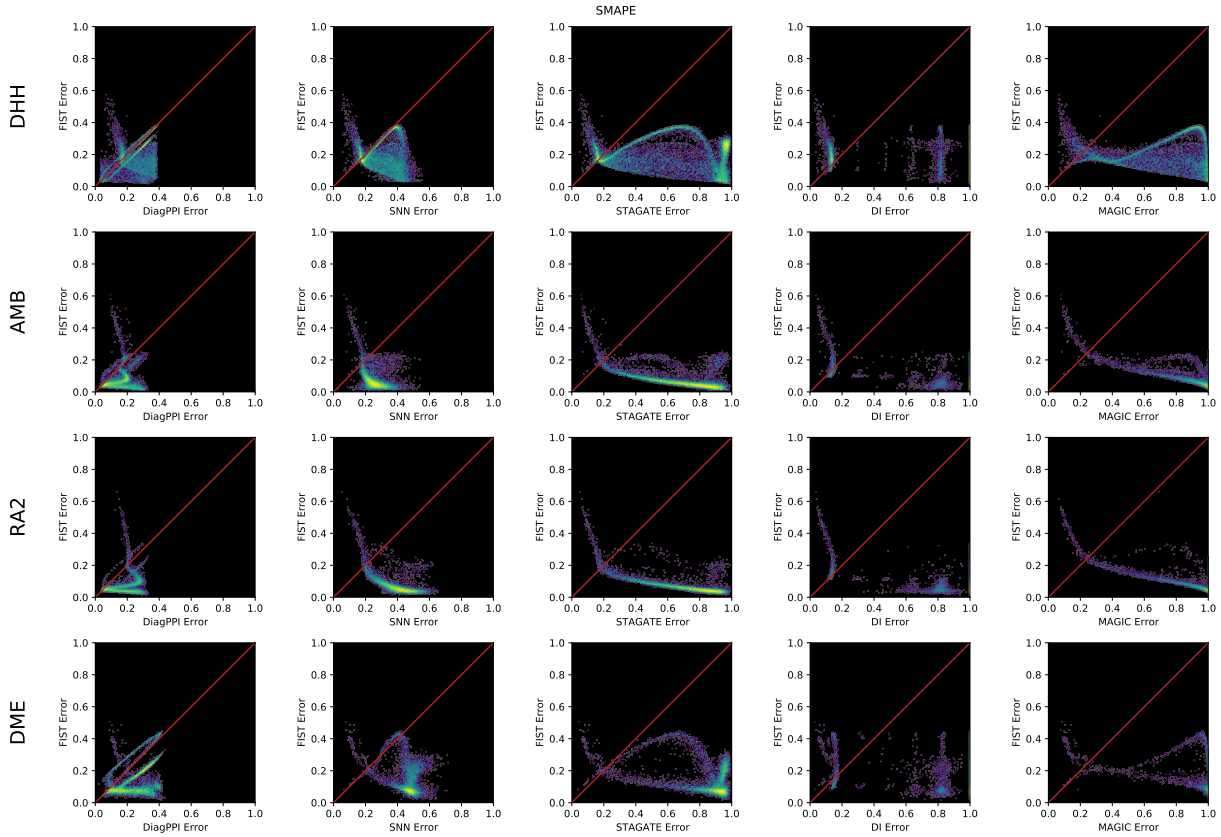

Figure S2: 2D histogram of SMAPEs for each gene for FIST against each method.

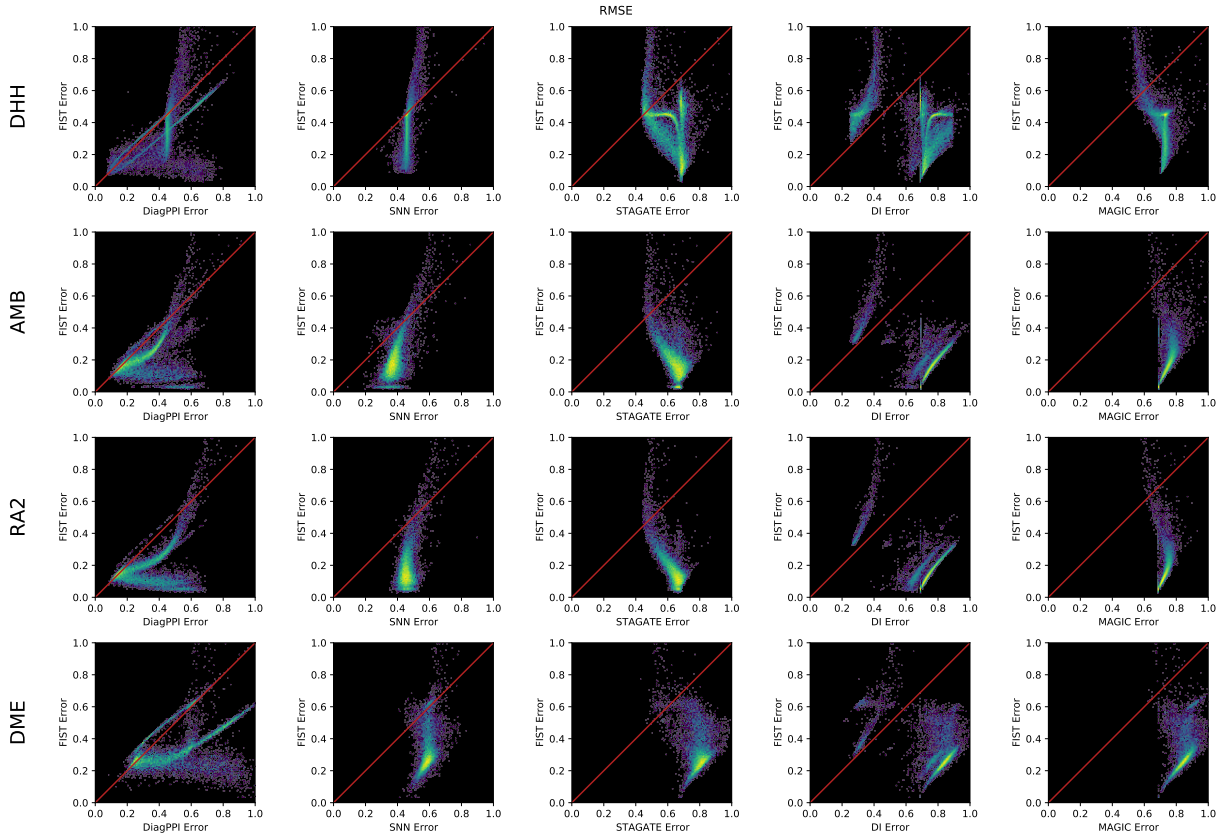

Figure S3: 2D histogram of RMSEs for each gene for FIST against each method.

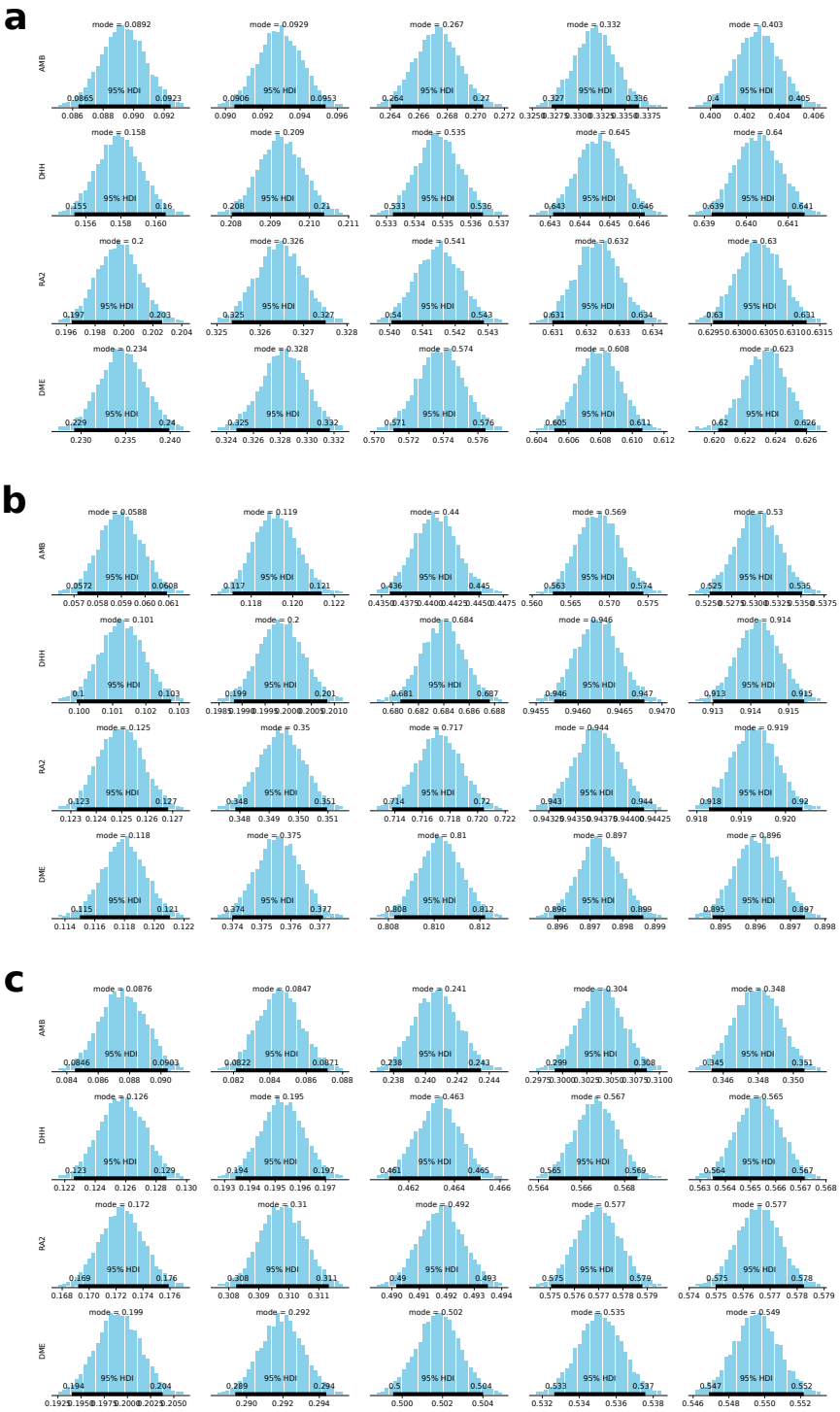

Figure S4: Posterior distributions of differences between mean (a) MAE, (b) SMAPE, and (c) RMSE between FIST and other methods. From left to right, FIST (diagonal PPI), SNN, STAGATE, DeepImpute, and MAGIC. The 95% HDI is highlighted. Computed using BEST (Bayesian Estimation Supersedes the t-Test).

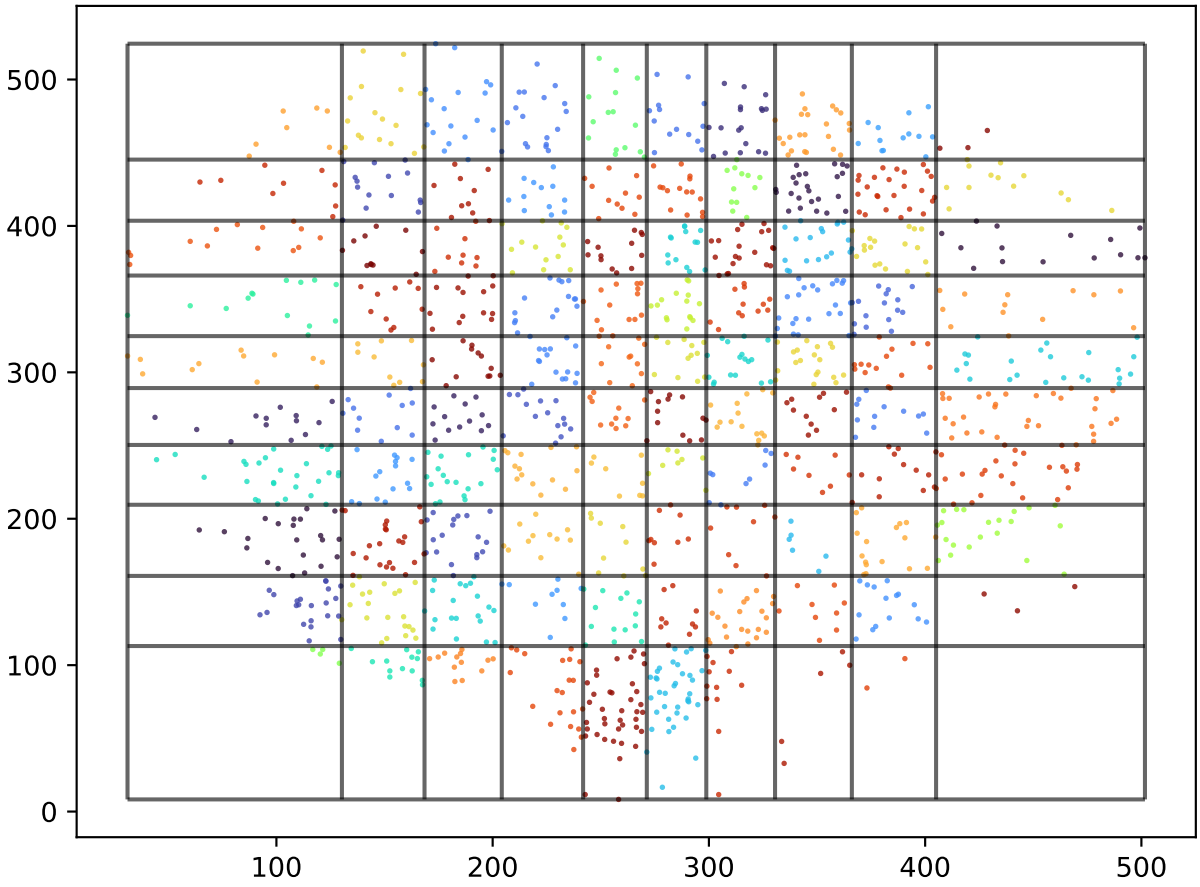

Figure S5: Visual representation of the binning process in two dimensions. Voxel boundaries are drawn on each axis corresponding to equally spaced quantiles of the data. The tensor entries for each gene are determined by the mean of all gene counts of spots that fall in the bin (here represented by color).