

METHODOLOGY

Open Access



# Identifying cell-type-specific spatially variable genes with ctSVG

Haotian Zhuang<sup>1</sup>, Xinyi Shang<sup>2</sup>, Wenpin Hou<sup>2\*</sup> and Zhicheng Ji<sup>1\*</sup>

\*Correspondence:

Wenpin Hou  
wh2526@cumc.columbia.edu  
Zhicheng Ji  
zhicheng.ji@duke.edu

Full list of author information is available at the end of the article

## Abstract

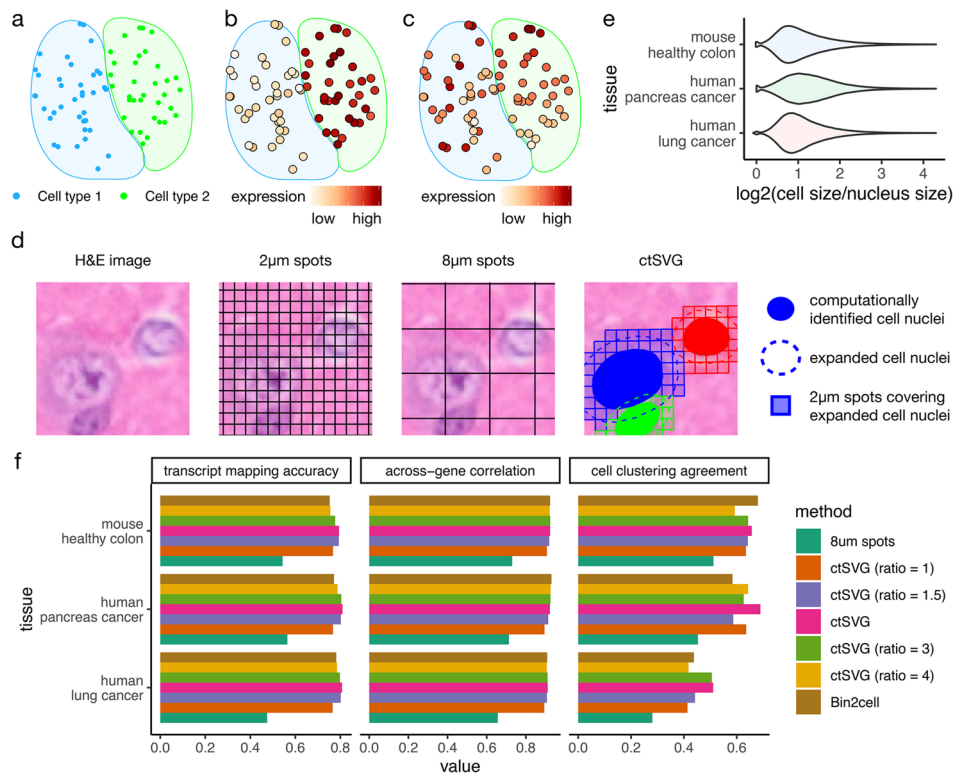
Spatially variable genes (SVGs) reveal the molecular and functional heterogeneity of cells across different spatial regions of a tissue. Sample-wide SVGs identified by existing methods largely overlap with cell-type marker genes derived from single-cell gene expression, leaving the spatial location information largely underutilized. We develop ctSVG, a computational method specifically tailored for Visium HD spatial transcriptomics at single-cell resolution. We show that cell-type-specific SVGs identified by ctSVG include many new genes that do not overlap with sample-wide SVGs or cell-type marker genes and that these genes reveal important biological functions in real spatial datasets.

## Background

Spatial transcriptomics (ST) technologies, which measure both gene expression and the spatial locations of cells, enable the study of how gene expression patterns change spatially across tissue regions. Identifying spatially variable genes (SVGs) is crucial for understanding the spatial heterogeneity of tissue structures and organization. Several computational methods have been developed to identify sample-wide SVGs across all cells in an ST sample, including SpatialDE [1], SPARK [2], Giotto [3], nnSVG [4], MERINGUE [5], and PreTSA [6]. These methods have been applied to study the spatial heterogeneity of gene expression in various tissues, such as breast cancer and the hippocampus [2].

However, sample-wide SVGs identified by these methods are confounded by the non-uniform spatial distribution of different cell types. In many tissues, such as the brain, cells belong to multiple cell types, and certain cell types are restricted to specific spatial regions. Thus, genes specifically expressed in a given cell type will also exhibit expression patterns specific to the spatial region occupied by that cell type (Fig. 1a-b). Since gene expression variation is largely driven by these cell type marker genes, existing methods tend to prioritize these genes as top SVGs. However, such genes can also be identified through differential analysis across cell types in conventional single-cell RNA-seq (scRNA-seq) data and do not necessarily depend on spatial information. Therefore,





**Fig. 1** **a–c** A schematic example showing the spatial distribution of two cell types (**a**), the expression of a sample-wide SVG that is also a marker gene for cell type 2 (**b**), and the expression of a cell-type-specific SVG in cell type 2, with a spatial expression pattern that decreases vertically (**c**). **d** An example spatial region showing the original H&E image, 2 $\mu$ m spots generated by Visium HD, the 8 $\mu$ m spots approach, and the expanded cell nuclei approach by ctSVG. **e** Log<sub>2</sub> ratios of cell sizes and nuclei sizes for different tissues. **f** Comparison of transcript mapping accuracy, across-gene correlation, and cell clustering agreement between ctSVG and competing methods

sample-wide SVGs introduce little new knowledge beyond what has already been learned from scRNA-seq data.

To address this issue, ideally, one needs to identify genes with spatially varying expression patterns within each cell type. These genes may not be cell type marker genes and may not appear at the top of the differential gene list in scRNA-seq analysis (Fig. 1c). Identifying such cell-type-specific SVGs will fully unlock the power of ST, leading to new insights into the spatially heterogeneous functions of each cell type. However, studying cell-type-specific SVGs has been a challenging task due to the low resolution of ST technologies, such as 10x Visium. In addition, a systematic comparison of sample-wide SVGs, cell-type-specific SVGs, and cell type marker genes is lacking.

The newly developed 10x Visium HD platform [7] represents a major breakthrough in ST technology, providing expression profiles for the whole transcriptome at single-cell resolution. Compared to other ST technologies that do not achieve single-cell resolution (e.g., 10x Visium) or can only reliably measure the expression profiles of a limited number of genes (e.g., 10x Xenium), Visium HD is ideal for identifying cell-type-specific SVGs. However, several obstacles remain in analyzing data from Visium HD. First, Visium HD measures gene expression profiles in 2-micron squares, and this information must be converted into single-cell gene expression before identifying individual cell types and cell-type-specific SVGs. Second, since cell types are inferred computationally from the data, they cannot be treated as fixed, unlike the spatial locations of

cells. It is necessary to account for the additional variation introduced by the uncertainty of inferred cell types when statistically modeling cell-type-specific SVGs in a rigorous manner, similar to previous work in pseudotime analysis [8].

In this study, we developed *ctSVG*, a computational method to extract single-cell gene expression profiles from Visium HD data and identify cell-type-specific SVGs. We systematically compared sample-wide SVGs identified by previous methods, cell-type-specific SVGs identified by *ctSVG*, and cell type marker genes across seven Visium HD datasets from different species and tissue types. We evaluated the accuracy of *ctSVG* and other SVG identification methods in detecting cell-type-specific SVGs. We also explored the spatial gene expression patterns of identified cell-type-specific SVGs in two tissues. Our results demonstrate the unique advantage of cell-type-specific SVGs in understanding the spatial heterogeneity of cells.

## Results

### *ctSVG* accurately assigns Visium HD squares to cells

The default analysis pipeline of Visium HD pools  $2\mu\text{m}$  squares into  $8\mu\text{m}$  squares and treats these  $8\mu\text{m}$  squares as the smallest units in downstream analysis. However, these  $8\mu\text{m}$  squares cannot accurately reflect the gene expression profiles of single cells. Figure 1d shows an example where a single cell is captured by multiple  $8\mu\text{m}$  squares, and one  $8\mu\text{m}$  square overlaps with multiple cells. In comparison, *ctSVG* first performs cell segmentation on the accompanying H&E images to identify the boundaries of cell nuclei, then expands the nuclei boundaries to approximate the whole cell boundaries, and finally pools  $2\mu\text{m}$  squares covering the expanded cell nuclei (Fig. 1d, [Methods](#) section). The computational expansion of cell nuclei is necessary because the whole cell boundary is difficult to directly obtain from H&E images.

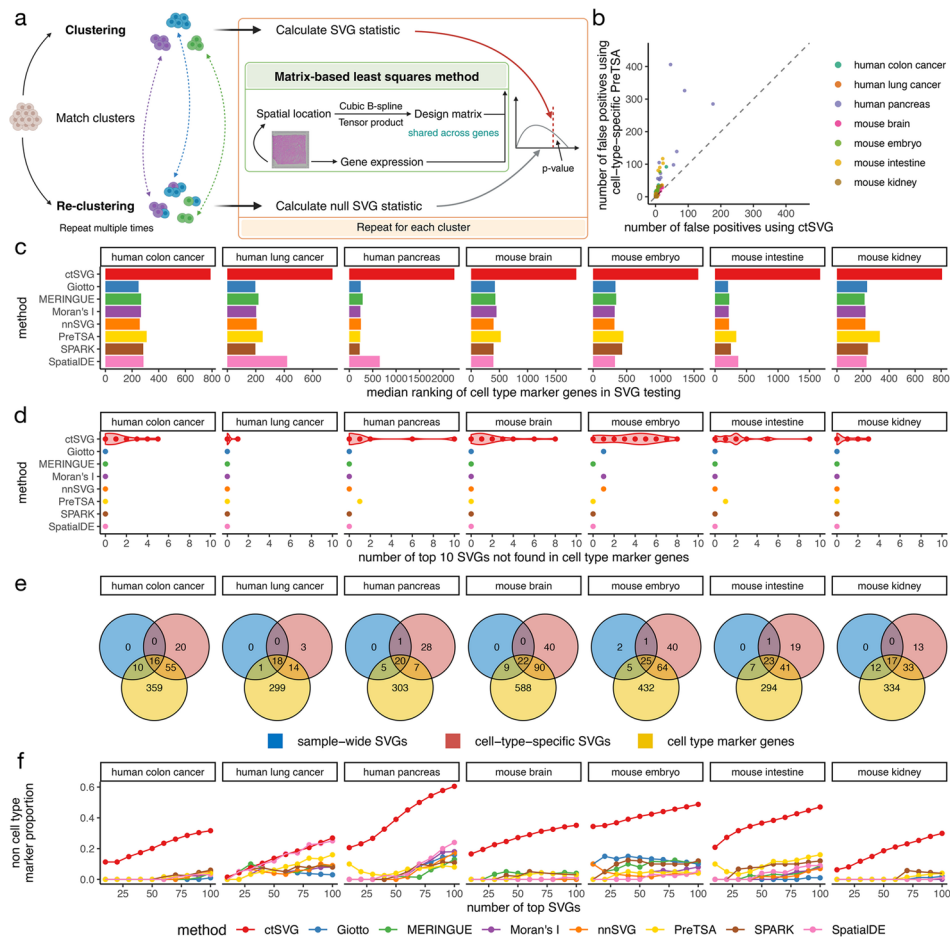
We tested the optimal strategy for cell nuclei expansion using 10x Xenium data with multimodal cell segmentation. This dataset contains both whole cell boundaries and cell nuclei boundaries. We found that the area occupied by a whole cell is typically twice the area occupied by its nucleus (Fig. 1e). Therefore, in *ctSVG*, the area of the expanded cell nuclei is set to be twice the area of the original cell nuclei.

We evaluated the performance of Visium HD's default strategy of  $8\mu\text{m}$  squares, a published method, *Bin2cell* [9], *ctSVG* with different levels of cell nuclei expansion, and default *ctSVG* using the 10x Xenium data (Fig. 1f, [Methods](#) section). Results obtained through 10x Xenium's whole cell segmentation are treated as the gold standard. We found that *ctSVG* is able to correctly assign most transcripts to their corresponding cells with an accuracy of around 80%. The gene expression profiles obtained by *ctSVG* are highly consistent with the gold standard, as demonstrated by high across-gene correlations. In terms of downstream analysis, cell clustering results obtained by *ctSVG* also show high agreement with the gold standard. *ctSVG* shows the best overall performance across the three metrics compared to other methods, demonstrating that *ctSVG* more accurately captures single-cell gene expression profiles. These single-cell gene expression profiles can then be processed using pipelines designed for single-cell analysis, such as Seurat [10] and GPTCelltype [11], for identifying cell clusters and cell types.

### Cell-type-specific SVGs identified by ctSVG provide new biological insights

Next, ctSVG uses a computationally efficient approach to identify cell-type-specific SVGs (Fig. 2a), based on our previous work with PreTSA [6]. We have demonstrated that PreTSA is the only method capable of fitting and testing SVGs for large ST datasets in a reasonable amount of time [6]. For each cell cluster, ctSVG fits a B-spline regression model to capture the spatial expression pattern of a gene. Since the spatial location of a cell is fixed, these regression models share the same design matrix, enabling ctSVG to perform all computations related to the design matrix once, greatly increasing computational efficiency. ctSVG does not provide global SVG results that reflect spatial gene expression patterns across the whole tissue. Users who wish to obtain global SVG results may refer to traditional SVG methods such as SPARK and PreTSA.

Identifying cell-type-specific SVGs differs from identifying global SVGs in that the cell clustering step introduces additional variation that must be accounted for in the statistical model. This variation arises from the inherent randomness of clustering algorithms



**Fig. 2** **a** Framework of ctSVG for identifying cell-type-specific SVGs. **b** Number of false positives identified by ctSVG, or by PreTSA in each cell type separately, in a null simulation study. **c** Median ranking of all cell type marker genes in SVG results obtained by each method. For ctSVG, the median of SVG results across all cell types is shown. **d** Number of top 10 SVGs identified by each method that are not cell type marker genes. For ctSVG, each dot represents the result in one cell type. **e** Venn diagrams showing the overlap among the union of top 10 sample-wide SVGs identified by all existing methods, union of top 10 cell-type-specific SVGs identified by ctSVG, and cell type marker genes. **f** The proportion of top SVGs identified by each method that are not cell type marker genes. For ctSVG, the average proportion across cell types is shown

used in contemporary scRNA-seq analysis pipelines, such as the Louvain clustering algorithm employed by Seurat [10]. Even when using the exact same clustering procedure, the cell clustering results may vary depending on the random seed. To account for this variation, ctSVG reassigns cells to clusters in the statistical testing step (Methods section). This non-parametric approach allows ctSVG to account for additional statistical variance induced by computationally inferred cell clusters, similar to pseudo-time analysis [8]. In a null simulation study, where the cell locations were randomly permuted, the non-parametric strategy used by ctSVG substantially reduces false positives compared to the parametric strategy used by PreTSA, which does not account for the additional variance (Fig. 2b). In additional sensitivity analyses, we showed that the differential results produced by ctSVG are robust to varying levels of cell nuclei expansion during the preprocessing of Visium HD data (Additional file 1: Fig. S1) and to small perturbations in cell clustering (Additional file 1: Fig. S2).

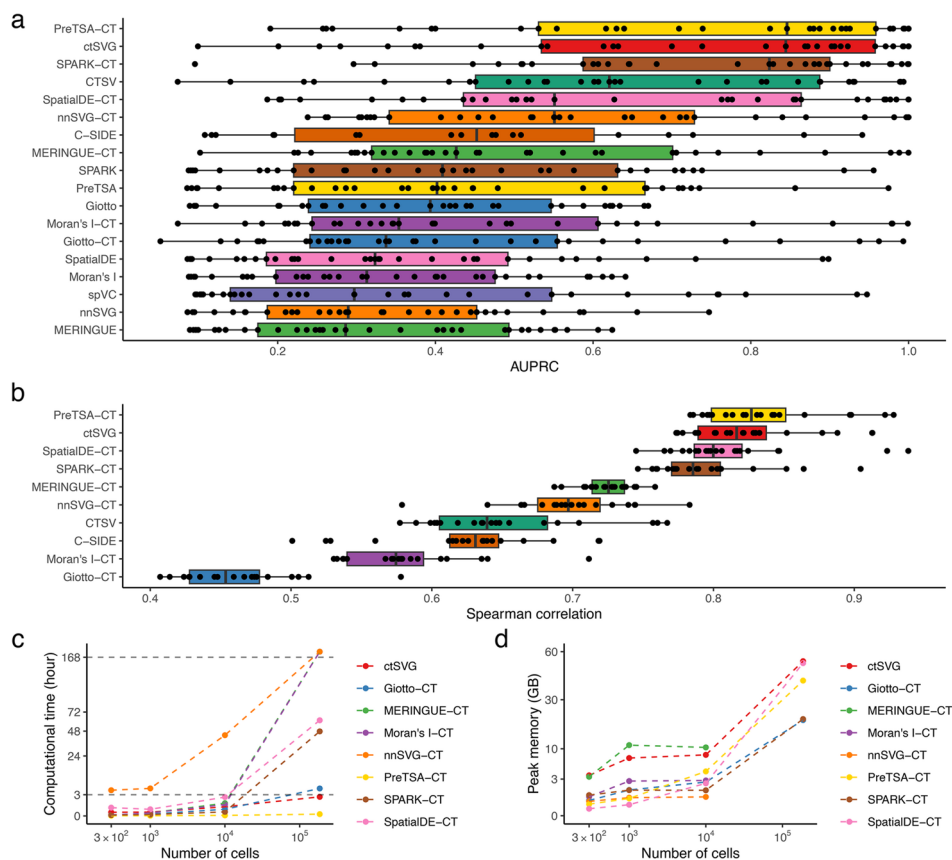
We then evaluated whether cell-type-specific SVGs identified by ctSVG can reveal new genes that are not cell type marker genes in three human tissues and four mouse tissues. Seven sample-wide SVG methods were also included for comparison. Cell type marker genes are defined as differential genes compared across cell clusters (Methods section). Compared to ctSVG, sample-wide SVG methods have a substantially higher tendency to rank cell type markers as top SVGs (Fig. 2c). In almost all cases, the top 10 sample-wide SVGs are all cell type marker genes, whereas the top 10 SVGs identified by ctSVG include many new genes that are not cell type marker genes (Fig. 2d-e). A similar trend holds for larger numbers of top SVGs (Fig. 2f). While around 40% of the top 100 SVGs identified by ctSVG are new genes in many tissues, only around 10% of the top 100 sample-wide SVGs are new genes in most tissues. These results suggest that sample-wide SVGs are highly consistent with cell type marker genes, whereas ctSVG can identify new genes that cannot be discovered by simply comparing gene expression across cell clusters.

### ctSVG outperforms existing methods in identifying cell-type-specific SVGs

We then evaluated the performance of ctSVG and other competing methods in identifying cell-type-specific SVGs through a simulation study. The simulation datasets were generated from each of the seven real Visium HD datasets using scDesign3 [12], following the procedures of a recent benchmark study [13] (Methods section). Each simulation dataset consists of 10,000 cells. Within each cell type, a subset of genes was constructed as SVGs, while the remaining genes were designated as non-SVGs. Note that a gene constructed as an SVG in one cell type may not be an SVG in another cell type. This information was treated as ground truth. We compared three types of competing methods. The first type includes existing SVG identification methods, such as SPARK and SpatialDE, applied to the entire tissue slide without considering cell types. The second type, denoted as SPARK-CT, SpatialDE-CT, etc., includes the same methods as in the first group, but applied separately to each cell type, similar to ctSVG. The third type includes CTSV [14], C-SIDE [15], and spVC [16], which are cell-type-specific methods designed mainly for low-resolution ST platforms such as 10x Visium. Each SVG identification method was applied to each simulated ST dataset, and the area under the precision-recall curve (AUPRC) was calculated to quantitatively assess the accuracy of the identified SVGs.

PreTSA-CT and ctSVG are the two best-performing methods and produce nearly identical results (Fig. 3a). Existing SVG methods applied in a cell-type-specific manner, such as SPARK-CT, also demonstrate strong performance and outperform their sample-wide counterparts, such as SPARK. Among the three existing cell-type-specific methods, CTSV shows the strongest performance and ranks among the top-performing approaches, though it remains inferior to PreTSA-CT, ctSVG, and SPARK-CT. The performance of C-SIDE is comparable to that of other cell-type-specific methods, whereas spVC performs poorly. As noted in the original spVC study [16], a likely reason is that when a covariate is binary and the spots with value one are spatially clustered rather than dispersed, the spatially varying effect of the covariate cannot be separated from the residual spatial effect. This limitation suggests that spVC may be less suitable for Visium HD data, where cell-type covariates are typically binary, than for Visium data, where cell-type covariates are often continuous and derived from deconvolution methods. These results suggest that cell-type-specific SVG methods are more effective at identifying cell-type-specific SVGs and that ctSVG is among the top-performing approaches.

To overcome the limitations of simulation studies, which may not fully capture the complexity of biological signals in real tissues, we further assessed method performance on real Visium HD datasets. Because the original datasets were too large for some



**Fig. 3** **a** Area under the precision-recall curve (AUPRC) for the simulation analysis. Methods are ranked in decreasing order based on their median AUPRC values. Each dot represents a cell type in a simulated dataset. **b** Spearman correlation coefficients from the reproducibility analysis. Methods are ranked in decreasing order of their median Spearman correlation coefficient values. **c** Computational time for each method with varying numbers of cells in the dataset. **d** Peak memory usage for each method with varying numbers of cells in the dataset

methods to run within a reasonable time, we analyzed rectangular subsets of each dataset. The number of cells in each dataset after subsetting is shown in Additional file 1: Fig. S3. In the absence of ground-truth labels for cell-type-specific SVGs, we adopted a reproducibility analysis as a proxy for accuracy, following the procedures of a recent benchmark study [13] (Methods section). Specifically, we randomly subsampled 80% of the cells within each cell cluster, applied each SVG identification method to both the original and subsampled datasets, and computed the Spearman correlation coefficient between the overall gene rankings. Methods that reliably detect true SVGs are expected to yield high consistency, as genuine spatial patterns should be preserved, whereas genes lacking robust spatial structure are more prone to random variation and thus less likely to be reproducibly identified.

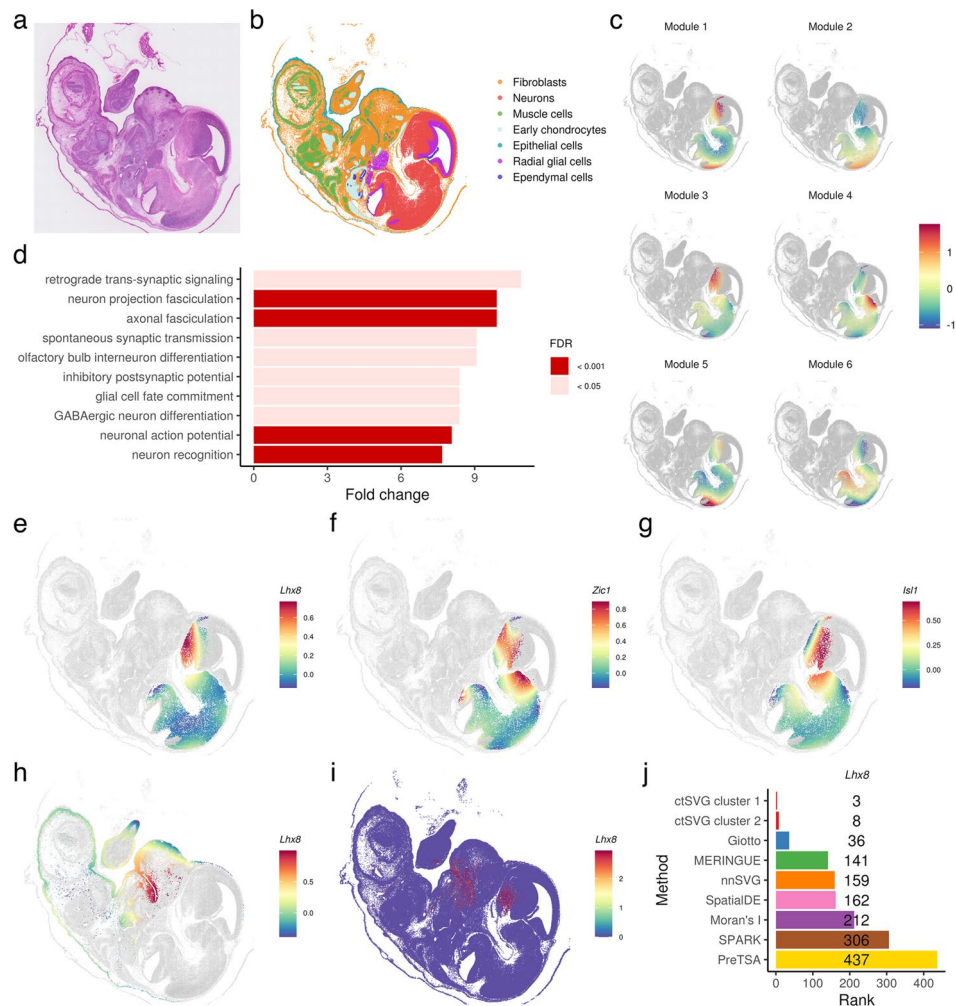
We compared ten methods that performed well in the simulation study. As shown in Fig. 3b, ctSVG and PreTSA-CT consistently achieve the highest reproducibility scores across all datasets, indicating that they outperform other existing methods in identifying stable, spatially variable genes. The similar performance of ctSVG and PreTSA-CT can be attributed to their shared approach in modeling the spatial patterns of gene expression. However, ctSVG additionally accounts for variation in cell clustering, which further reduces false positives.

To assess the scalability of each method, we measured and compared the runtime and memory consumption of ctSVG and other existing approaches using the human colon cancer dataset, along with its down-sampled variants (Methods section). As shown in Fig. 3c, ctSVG demonstrates strong computational efficiency, processing large datasets within a reasonable timeframe. It ranks as the second fastest method, closely following PreTSA-CT. The additional runtime required by ctSVG is due to its incorporation of an extra step to account for variation introduced by cell clustering, which is essential for controlling false positives. In contrast, many existing methods fail to complete the analysis of a single dataset within a week. Moreover, ctSVG's memory usage remains within a practical range for most users with access to moderate computational resources (Fig. 3d). Additional file 1: Fig. S4 provides additional results on the runtime and memory usage of ctSVG across seven real Visium HD datasets.

### Cell-type-specific SVGs in mouse embryo

After identifying cell-type-specific SVGs, ctSVG provides comprehensive functions for visualizing and analyzing these genes as part of downstream analysis. As an example, we applied ctSVG to a Visium HD dataset from an E15.5 mouse embryo (Fig. 4a). Using ctSVG, we extracted single-cell gene expression profiles, performed cell clustering, and identified seven cell types based on marker genes (Fig. 4b, Additional file 1: Fig. S5-6). For each cell cluster, ctSVG organizes cell-type-specific SVGs into distinct gene modules and identifies enriched gene ontology (GO) terms for each module. Additionally, ctSVG visualizes the spatial gene expression patterns of each module through a metagene, constructed by averaging the fitted gene expression values within each module (Fig. 4c).

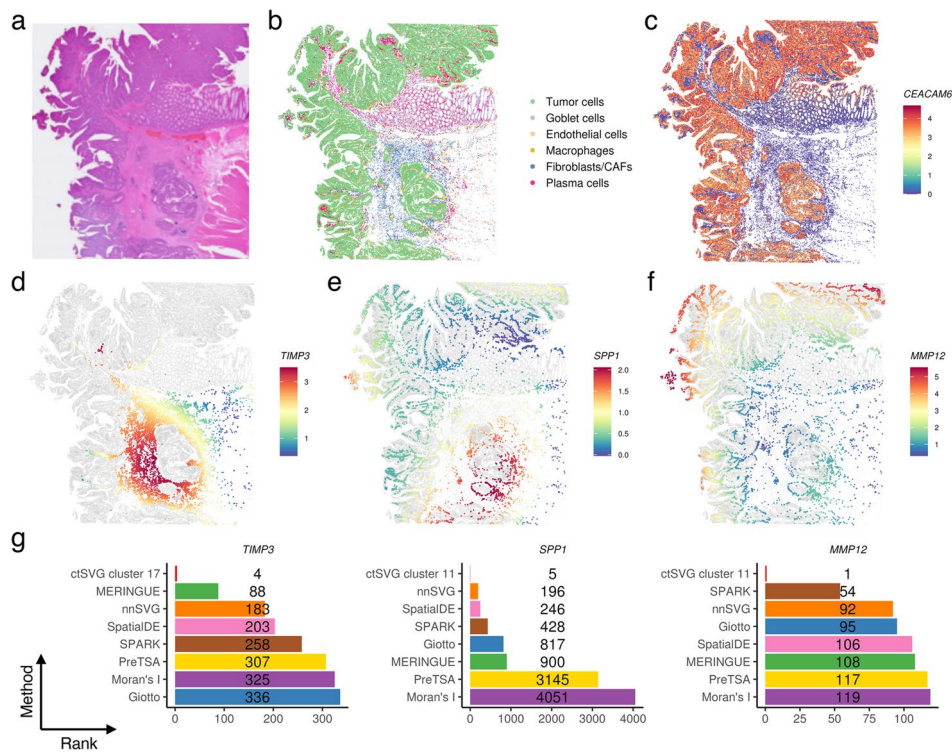
The gene modules identified by ctSVG align well with known biology in the mouse embryo. For example, consider cell cluster 1, which represents neurons (Fig. 4c-g). The metagene of gene module 3 shows substantially higher expression levels in spatial regions corresponding to the head (Fig. 4c). Consistent with neuronal development, GO



**Fig. 4** **a** H&E image of the mouse embryo tissue. **b** Cell type annotations based on unsupervised clustering and marker genes. **c** Spatial gene expression pattern of metagenes for each gene module in cell cluster 1 (neurons). **d** Top GO terms enriched in gene module 3 of cell cluster 1. **e–g** Fitted spatial gene expression patterns of *Lhx8* (**e**), *Zic1* (**f**), and *Isl1* (**g**) in cell cluster 1. **h** Fitted spatial gene expression pattern of *Lhx8* in cell cluster 2 (fibroblasts). **i** Spatial gene expression pattern of *Lhx8* across all cells. **j** Ranking of *Lhx8* across different SVG methods

terms enriched in gene module 3 include neuron projection fasciculation, axonal fasciculation, and olfactory bulb interneuron differentiation (Fig. 4d). Many of the top-ranked cell-type-specific SVGs in gene module 3 are also reported to be associated with neuronal development, including *Lhx8* [17, 18], *Zic1* [19–21], and *Isl1* [22–24] (Fig. 4e–g).

We further investigated the function of *Lhx8*, a gene critical for the formation of fore-brain cholinergic neurons [17] and associated with tooth development [25]. ctSVG ranked *Lhx8* as the third and eighth most differential gene in cell cluster 1 (neurons) and cell cluster 2 (fibroblasts), respectively. In neurons, *Lhx8* shows high expression in the head region (Fig. 4e), while in fibroblasts, it exhibits high expression in the mouth region (Fig. 4h). These expression patterns are consistent with the known biological functions of *Lhx8*. In comparison, these cell-type-specific patterns are masked when examining spatial gene expression across the entire sample (Fig. 4i), and *Lhx8* does not appear among the top 100 SVGs identified by most sample-wide SVG methods (Fig. 4j). These results further demonstrate that cell-type-specific SVGs identified by ctSVG can reveal new biological insights that sample-wide SVG methods may overlook.



**Fig. 5** **a** H&E image of the human colon cancer tissue. **b** Cell type annotations based on unsupervised clustering and marker genes. **c** Spatial gene expression pattern of the tumor marker *CEACAM6*. **d** Fitted spatial gene expression pattern of *TIMP3* in cell cluster 17 (fibroblasts/CAFs). **e-f** Fitted spatial gene expression patterns of *SPP1* (**e**) and *MMP12* (**f**) in cell cluster 11 (macrophages). **g** Rankings of *TIMP3*, *SPP1*, and *MMP12* across different SVG methods

### Cell-type-specific SVGs in human colon cancer

As another example, we used *ctSVG* to identify cell-type-specific SVGs related to tumor progression in a human colorectal cancer (CRC) sample (Fig. 5a). After processing the data with *ctSVG*, we identified six cell types, including tumor cells, based on marker genes (Fig. 5b, Additional file 1: Fig. S7-8). The spatial locations of tumor cells align with the H&E image (Fig. 5a) and the expression of *CEACAM6*, a tumor marker gene (Fig. 5c). The tumor cells were spatially concentrated in two main areas, referred to as the central and peripheral tumor regions.

Among the top-ranked cell-type-specific SVGs identified by *ctSVG*, many have been reported to play roles in tumor. For instance, *ctSVG* ranked *TIMP3* as the fourth most significant gene in cell cluster 17, a fibroblast/CAF cell cluster (Additional file 1: Fig. S7). *TIMP3* shows higher expression in cells situated between the central and peripheral tumor regions, suggesting its potential role in tumor suppression (Fig. 5d). This finding aligns with previous research showing that *TIMP3* inhibits matrix metalloproteinases, preventing extracellular matrix degradation and subsequent tumor invasion. This activity is crucial for maintaining structural integrity between tumor clusters to avoid their spread and merging [26, 27].

We further examined the spatial heterogeneity of immune responses in the central and peripheral tumor regions, focusing on macrophages, the most abundant immune cell type in this dataset (Fig. 5b). *ctSVG* identified *MMP12* and *SPP1* as two top-ranked cell-type-specific SVGs in a macrophage cluster (cell cluster 11, Additional file 1: Fig. S7). When expressed in macrophages, *MMP12* may inhibit intestinal tumor growth by

influencing macrophage polarization [28]. Conversely, *SPP1*+ macrophages in CRC can promote immune evasion and tumor progression by supporting a desmoplastic tumor structure through interactions with *FAP*+ fibroblasts [29]. We observed that macrophages nearer the central tumor region expressed higher levels of *SPP1* and lower levels of *MMP12*, while macrophages closer to the peripheral tumor region exhibited the opposite expression pattern of *SPP1* and *MMP12* (Fig. 5e-f). These findings suggest that the central tumor region is more likely to be an immunosuppressive environment promoting tumor growth, whereas tumor growth may be inhibited in the peripheral tumor region.

Similar to the previous example, *TIMP3*, *SPP1*, and *MMP12* do not appear among the top 100 SVGs in most sample-wide SVG methods (Fig. 5g).

## Discussion

Similar to a recent study [13], our work relies on simulation datasets and reproducibility analyses in real datasets to benchmark method performance. The absence of ground truth for cell-type-specific SVGs makes direct evaluation in real data infeasible. Reproducibility therefore serves as a practical proxy, while simulations generated by *scDesign3* provide realistic controlled assessments. Notably, the consistent top performance of *ctSVG* across both evaluations strongly supports its reliability for identifying cell-type-specific SVGs.

## Conclusions

In this study, we developed *ctSVG*, a computational tool for processing Visium HD data and identifying cell-type-specific SVGs. We demonstrated that while sample-wide SVGs largely overlap with cell type marker genes, cell-type-specific SVGs introduce many new genes that are not found among cell type marker genes. We further showed that these cell-type-specific SVGs are important for understanding the molecular and functional heterogeneity of cell types across spatial regions. Beyond Visium HD data, *ctSVG* can also be applied broadly to other types of ST data with single-cell resolution, such as 10x Xenium.

## Methods

### *ctSVG*

#### *Input data*

For analyzing 10x Visium HD data, *ctSVG* requires two inputs. The first input is the output from the standard 10x Space Ranger pipeline. The second input is the nuclei segmentation results obtained by running segmentation methods, such as *StarDist* [30], on the H&E images accompanying the Visium HD data.

For platforms other than 10x Visium HD, *ctSVG* can identify cell-type-specific SVGs without performing the data processing steps specific to Visium HD. In this case, *ctSVG* requires only the gene expression count matrix and a matrix of cell spatial coordinates as inputs.

#### *Obtaining aggregated single-cell gene expression profiles*

*ctSVG* first filters out cell nuclei with abnormally large sizes. The area of each segmented cell nucleus is calculated using the *sf* package (version 1.0–16.0) in R, and then

log-transformed. A cutoff is determined as the mean of the log-transformed areas across all nuclei, plus two times the standard deviation of the log-transformed areas. Nuclei with log-transformed areas larger than this cutoff are filtered out.

`ctSVG` then assigns each Visium  $2\mu\text{m}$  square to the cell nucleus it overlaps with. For a  $2\mu\text{m}$  square that overlaps with multiple nuclei, the square is uniquely assigned to the nucleus with the largest area of overlap. Note that, for computational efficiency, the area of overlap is approximated. Specifically, each square is divided into 100 subsquares, and the area of overlap is estimated by the number of subsquares that the nucleus overlaps with.

Next, `ctSVG` expands each cell nucleus to approximate its whole cell boundary, so that the area of the expanded nucleus is twice that of the original nucleus. The ratio of areas between the expanded and original nucleus can be optionally specified by the user. To perform the expansion, the centroid of each nucleus is first calculated using the `st_centroid` function from the `sf` package. Denote the coordinates of the centroid as  $(x_0, y_0)$ , and the coordinates of the  $N$  contour points that define the segmentation as  $(x_i, y_i)$ , where  $i = 1, \dots, N$ . The coordinates of the expanded contour points are calculated as  $\sqrt{r}(x_i - x_0) + x_0$  and  $\sqrt{r}(y_i - y_0) + y_0$ , where  $r$  is the ratio of areas between the expanded and original nucleus.

For any  $2\mu\text{m}$  square that has not yet been assigned to a cell, `ctSVG` repeats the square assignment procedure described above, this time using the expanded cell nuclei.

After the squares have been assigned to cells, `ctSVG` identifies and removes three types of abnormal cells. A cell is considered abnormal if no  $2\mu\text{m}$  square is assigned to its nucleus, if the total area covered by all assigned  $2\mu\text{m}$  squares is less than half of the cell's area, or if the assigned  $2\mu\text{m}$  squares are disconnected. Note that these abnormal cells are rare in real datasets.

Finally, `ctSVG` generates the single-cell gene expression count matrix. For each gene, `ctSVG` aggregates the gene expression counts across all  $2\mu\text{m}$  squares assigned to a cell to obtain that cell's gene expression profile.

### ***Processing gene expression data***

Seurat (version 4.4.0) was used to process the single-cell gene expression count matrix. Specifically, cells with positive expression in at least 300 genes were retained, and genes with positive expression in at least 1% of all retained cells were kept. The log-normalized gene expression matrix was obtained using the `NormalizeData` function with default settings. Highly variable genes were identified using the `FindVariableFeatures` function with default parameters. The matrix was scaled using the `ScaleData` function. PCA was performed using the `RunPCA` function. Cell clustering was conducted using the `FindNeighbors` function on the top 10 PCs, followed by `FindClusters` with the resolution set to 1.2.

### ***Removing spatially isolated cells***

Before fitting and testing cell-type-specific SVGs, `ctSVG` filters out cells that are spatially distant from all other cells within each cell cluster. This filtering is done separately for each cell cluster. First, the Euclidean distance from each cell to all other cells within the same cell cluster is calculated. The isolation score of a cell is then defined as the average Euclidean distance to the 1% nearest cells (with a maximum of 50 cells and a

minimum of 10 cells). Cells with an isolation score greater than the mean isolation score across all cells, plus six times the standard deviation, are removed.

Finally, genes with positive expression in at least 1% of all retained cells within each cell cluster are kept. Note that this gene filtering was not applied in the analyses shown in Fig. 2 to ensure the results of ctSVG are consistent with those of the sample-wide SVGs.

#### **Fitting cell-type-specific spatial gene expression patterns**

ctSVG uses the same approach as PreTSA to fit the spatial expression pattern of a gene. The details of estimating the fitted gene expression values and associated test statistics were described in our previous work, PreTSA [6]. For the readers' convenience, we briefly introduce the fitting procedure here.

ctSVG sequentially fits the following regression model for each cell cluster. For a given cell cluster  $c$ , let  $Y$  be its  $m \times n$  gene expression matrix, where  $m$  is the number of genes and  $n$  is the number of cells in cluster  $c$ . The entry  $y_{ij}$  denotes the expression level of gene  $i$  in cell  $j$ , with cell  $j$  belonging to cluster  $c$ . Let  $S_j = (s_{j1}, s_{j2})$  represent the 2-dimensional spatial coordinates of cell  $j$ .

For each gene  $i$ , ctSVG models its expression values across spatial locations as a functional surface:

$$y_{ij} = \alpha_{i,c} + \sum_{k_1=1}^{K+3} b_{1,k_1,c}(s_{j1})\beta_{i,k_1,0,c} + \sum_{k_2=1}^{K+3} b_{2,k_2,c}(s_{j2})\beta_{i,0,k_2,c} \\ + \sum_{k_1=1}^{K+3} \sum_{k_2=1}^{K+3} b_{1,k_1,c}(s_{j1})b_{2,k_2,c}(s_{j2})\beta_{i,k_1,k_2,c} + \epsilon_{i,j,c} \\ \epsilon_{i,j,c} \stackrel{\text{iid}}{\sim} N(0, \sigma_{i,c}^2).$$

Here,  $b_{d,1,c}(s), \dots, b_{d,K+3,c}(s)$  represent the  $K+3$  cubic B-spline basis functions for each dimension ( $d=1,2$ ), where  $K$  is the number of equidistant internal knots used to define the cubic B-spline bases. The parameters  $\alpha_{i,c}, \beta_{i,1,0,c}, \dots, \beta_{i,K+3,0,c}, \beta_{i,0,1,c}, \dots, \beta_{i,0,K+3,c}, \beta_{i,1,1,c}, \dots, \beta_{i,K+3,K+3,c}$  and  $\sigma_{i,c}^2$  are all unknown and will be estimated using the least squares method. An F statistic is subsequently calculated for the fitted model as the test statistic.

#### **Testing cell-type-specific SVGs by accounting for cell clustering variations**

ctSVG applies a permutation procedure to test the statistical significance of cell-type-specific SVGs. Specifically, cell clustering is redone by changing the `random.seed` parameter to  $1, 2, \dots, 1,000$  in the `FindClusters` function. A Jaccard index is calculated between each original cluster and each reassigned cluster. For each original cluster, the reassigned cluster with the highest Jaccard index is retained. This process is repeated 1,000 times, and for each original cluster, the 100 reassigned clusters with the highest Jaccard indices are selected. This reclustering step accounts for additional variation introduced by computationally inferred cell clusters.

For each original cluster, the spatial locations of cells in its corresponding reassigned cluster are randomly permuted. The same fitting approach is then applied to the reassigned and permuted cluster to obtain the null test statistic for each gene. This step is repeated for each of the 100 reassigned clusters to generate 100 null test statistics. To enhance numerical accuracy, a Gamma distribution is fitted to these 100 null test

statistics using the R package `fitdistrplus` (version 1.1–11.1). The  $p$ -value is calculated as the tail probability of the fitted Gamma distribution exceeding the test statistics calculated from the original data. All  $p$ -values are then adjusted for multiple testing using the Benjamini-Hochberg (BH) procedure to obtain false discovery rates (FDRs) [31]. By default, an FDR of  $\leq 0.05$  is used as the significance cutoff.

#### **Gene clustering and functional analysis**

In each cell cluster, the fitted values of each cell-type-specific SVG across all cells are standardized to have a mean of zero and a standard deviation of one. These cell-type-specific SVGs are then grouped into different gene modules based on their spatial patterns using  $k$ -means clustering. The number of modules is automatically determined based on the proportion of the within-cluster sum of squares to the total sum of squares, using `findPC` [32] with default settings.

In each gene module, GO enrichment is performed using the R package `topGO` (version 2.56.0). All  $p$ -values are then adjusted for multiple testing using the BH procedure to obtain FDRs [31]. GO terms with an FDR of  $\leq 0.05$  are retained and then ordered in decreasing order by fold change.

#### **Evaluation of aggregated single-cell gene expression**

##### ***Xenium datasets***

Three Xenium datasets were downloaded directly from the 10x Genomics website (<https://www.10xgenomics.com/datasets>): human lung cancer, human pancreas cancer, and mouse colon. These Xenium datasets include both whole cell segmentation and cell nuclei segmentation results.

Since the original Xenium datasets are quite large, we selected a rectangular region in the middle of each image for computational efficiency. The center of the rectangle corresponds to the center of the original image, and the width and height of the rectangle are 20% of the original image's width and height, respectively. Only cells within this rectangle are considered in the subsequent analysis.

Transcripts labeled as “NegControlProbe”, “NegControlCodeword”, and “Unassigned-Codeword” were all removed.

##### ***ctSVG and competing method***

Instead of performing nuclei segmentation on H&E images, `ctSVG` directly uses the nuclei segmentation results provided by the original Xenium datasets. To mimic the dataset generated by Visium HD, the entire image was split into consecutive, non-overlapping  $2\mu\text{m}$  squares. The remaining steps of the `ctSVG` pipeline were then performed to assign the  $2\mu\text{m}$  squares to cells. As competing methods, nuclei boundaries were expanded such that the area of the expanded nuclei was set to be equal to the original area (no expansion), 1.5 times, 2 times (default in `ctSVG`), 3 times, and 4 times the area of the original nuclei.

For Visium HD's default strategy of using  $8\mu\text{m}$  squares, the entire image was split into consecutive, non-overlapping  $8\mu\text{m}$  squares. Each cell was assigned to the  $8\mu\text{m}$  square with the largest area of overlap with the cell. An approximation method similar to that used in `ctSVG` was applied to calculate the area of overlap.

Since `Bin2cell` was designed for Visium HD data and cannot be directly applied to Xenium datasets, we implemented an in-house version of `Bin2cell`. Specifically, the  $2\mu\text{m}$  squares were first assigned to cell nuclei using the same steps as `ctSVG`. Following the approach of `Bin2cell`, the assigned  $2\mu\text{m}$  squares were then expanded to include their two neighboring  $2\mu\text{m}$  squares to approximate whole-cell boundaries.

#### ***Evaluation of transcript mapping accuracy***

In the original Xenium datasets, each transcript is already assigned to a cell if it falls within the cell's whole cell boundary. A transcript is unassigned if it is not within any cell boundary. This information is treated as the gold standard.

Transcript mapping accuracy was calculated as the proportion of transcripts with matching assignments (either both assigned to the same cell or both unassigned) between the gold standard and the method being evaluated.

Note that with the  $8\mu\text{m}$  square method, a square can be assigned to multiple cells. Consequently, a transcript within such a square may also be assigned to multiple cells, complicating the evaluation of assignment agreement. To address this, we arbitrarily assigned the square to a randomly selected cell from the original list of cells it was assigned to, ensuring a one-to-one mapping between squares and cells. Transcript mapping accuracy was calculated afterward. This operation was not performed in other evaluations discussed below.

#### ***Evaluation of across-gene correlation and cell clustering agreement***

For both `ctSVG` and the  $8\mu\text{m}$  square method, single-cell gene expression matrices were obtained by counting the number of RNA transcripts falling within the squares assigned to each cell. The gene expression count matrices from the original Xenium data, `ctSVG`, and the  $8\mu\text{m}$  square method were then processed using `Seurat`, following the same procedure as in the standard `ctSVG` pipeline described above. The only differences were that cells with at least 10 total reads were retained, and the `scale.factor` parameter in the `NormalizeData` function was set to 100.

For gene expression count matrices obtained from different methods, we first retained cells with at least 10 total counts and genes with positive expression in at least 1% of all retained cells. Only genes and cells shared across all count matrices were kept. For each dataset, the smallest total count among all count matrices was used as the down-sampling target. Within each matrix, we preserved at least one count for every gene–cell pair that originally had positive expression. The remaining count budget was allocated by randomly sampling from the remaining transcript-level counts. The final down-sampled count matrix had an identical total count for comparison and preserved the sparsity structure of the original matrix.

The across-gene correlation was calculated using the scaled and log-normalized gene expression values. For each cell, a Pearson correlation coefficient was computed across genes between the original Xenium gene expression and the gene expression from either `ctSVG` or other competing method. The median correlation coefficient across all cells was then taken.

Cell clustering agreement was calculated using the adjusted Rand index (via the `adjustedRandIndex` function in the `mclust` R package) between the cell clustering

obtained from the original Xenium gene expression and the cell clustering obtained from either ctSVG or other competing method.

### **Evaluation of cell-type-specific SVGs**

#### ***Visium HD datasets***

Seven Visium HD datasets were downloaded directly from the 10x Genomics website (<https://www.10xgenomics.com/datasets>): human colorectal cancer, mouse small intestine, human lung cancer, mouse brain, human pancreas, mouse embryo, and mouse kidney.

All Visium HD datasets were processed using the standard ctSVG pipeline described above.

#### ***Null simulation***

For each Visium HD dataset, the spatial locations of cells were randomly permuted. Cell-type-specific PreTSA or ctSVG were applied to identify cell-type-specific SVGs. Genes with an FDR  $\leq 0.05$  were considered false positives.

#### ***Sensitivity analysis for cell nuclei expansion***

For each Visium HD dataset, the same procedure used to obtain aggregated single-cell gene expression profiles, as described above, was performed, except that nuclei boundaries were expanded such that the area of the expanded nuclei was set to be equal to (no expansion), 1.5 times, 2 times (default in ctSVG), 3 times, and 4 times the area of the original nuclei. The Spearman correlation coefficient was calculated for the overall gene rankings produced by ctSVG when applied to the original dataset and to datasets with varying levels of nuclei expansion, within each cell cluster.

#### ***Sensitivity analysis for perturbed cell clustering***

For each Visium HD dataset, we randomly selected 5% of cells and permuted the cluster labels of the selected cells. The Spearman correlation coefficient for the overall gene rankings by ctSVG on the original and perturbed datasets was calculated within each cell cluster.

#### ***Cell type marker gene identification***

Genes with differential expression between cells from one cluster and all other cells were identified using Seurat's FindAllMarkers function, with the `max.cells.per.ident` parameter set to 5000. In each cluster, cell type marker genes were identified as the 50 genes with the smallest  $p$ -values. A union set of cell type marker genes was taken across all clusters.

#### ***Competing methods***

Giotto (version 1.0.4), MERINGUE (version 1.0), the RunMoransI function in Seurat (version 4.4.0), nnSVG (version 1.7.4), PreTSA (version 1.1), the Gaussian version of SPARK (version 1.1.1), and SpatialDE (version 1.1.3) were used to identify sample-wide SVGs with default settings. The same methods were used to identify cell-type-specific SVGs by applying them to each cell type separately.

In the analyses shown in Fig. 2, `PreTSA` and `ctSVG` were applied directly to the original datasets. All other sample-wide SVG methods were performed on subsets of the original datasets, where 10,000 cells were randomly selected for each dataset. This was necessary because the original datasets were too large for sample-wide SVG methods other than `PreTSA` to complete in a reasonable time. In the analyses shown in Fig. 3, all methods were applied to the same datasets without additional subsetting.

`CTSV` (version 0.99.9), the `run.CSIDE.nonparam` function in `spacexr` (version 2.2.1), and `spVC` (version 0.1.0) were adapted to identify cell-type-specific SVGs. These three methods were designed mainly for low-resolution ST platforms and required a spot-level cell-type proportion matrix as the input. To adapt these methods to Visium HD data, a cell was treated as a spot that consists of only one cell, and a binary matrix indicating which cell type each cell belongs to was used as the surrogate of the cell-type proportion matrix. `CTSV` was originally designed to test the statistical significance of cell-type-specific SVGs along the horizontal and vertical directions separately, returning two corresponding  $p$ -values. The smaller of the two was then used as the final output  $p$ -value. `C-SIDE` inevitably filtered out a substantial proportion of genes for certain cell types, even after adjusting all filtering parameters. To ensure comparability, we retained results only for cell types with at least 1,000 genes remaining in both the simulation studies and the reproducibility analysis. `spVC` was originally developed as a two-step procedure, testing the statistical significance of cell-type-specific SVGs only for genes passing the first step. In our analysis, we applied the two-step procedure to all genes to ensure fair comparison.

### **Simulation studies**

`scDesign3` (version 1.5.0) was used to generate synthetic data from seven Visium HD datasets. Each original dataset was down-sampled to 10,000 cells and then processed using `Seurat`, following the same procedure described above, except that the `resolution` parameter in the `FindClusters` function was set to 0.1. For each gene, the proportion of cells with positive expression was calculated within each cell cluster. The top 2,000 genes with the highest minimum proportion of positive expression across all clusters were selected for simulation. Within each cell cluster, `scDesign3` was used to generate a simulated gene expression matrix, where 10% of the genes were designated as ground truth SVGs and the remaining genes as non-SVGs. For sample-wide SVG methods, the SVG results obtained from the entire dataset were repeatedly applied to each cell cluster for evaluation. The area under the precision-recall curve (AUPRC) was computed using the `pr.curve` function from the `PRROC` package.

### **Reproducibility analysis**

For each Visium HD dataset, we manually selected a rectangular region that did not have large empty areas, preserved a clear tissue structure, and contained a sufficient number of cells. Only cells within these regions were retained, and only cell clusters with at least 1,000 cells were kept. Within each cluster, 80% of the cells were randomly subsampled, and each SVG method was applied to both the original and subsampled datasets. The Spearman correlation coefficient was then computed to compare the overall gene rankings obtained from the original and subsampled datasets.

### **Evaluation of computational time and peak memory**

We randomly sampled 300, 1,000, and 10,000 cells without replacement from the Visium HD dataset of human colon cancer. Each SVG identification method was applied to either the full human colon cancer dataset or the subsampled datasets. All computations were performed using a single CPU thread, with a maximum runtime of 168 hours and a memory limit of 208 GB. For `SpatialDE` and `SpatialDE-CT`, the `perf_counter` function from the `time` package was used to record computational time, and the `get_traced_memory` function from the `tracemalloc` package was used to record peak memory usage. For all other methods, the `system.time` and `gc` functions were used to measure computational time and peak memory, respectively.

### **Supplementary Information**

The online version contains supplementary material available at <https://doi.org/10.1186/s13059-025-03870-6>.

Additional file 1: Fig. S1. Spearman correlation coefficient between the overall gene rankings obtained by the default `ctSVG` and those obtained by `ctSVG` with different levels of cell nuclei expansion. Fig. S2. Spearman correlation coefficient between the overall gene rankings obtained by `ctSVG` from the original data and those from the perturbed data. Fig. S3. Number of cells in each Visium HD dataset after rectangular subsetting. Fig. S4. Computational time and peak memory usage of `ctSVG` across seven real Visium HD datasets. Fig. S5. Cell clustering and cell type annotation in the mouse embryo tissue. Fig. S6. Expression level of marker genes in the mouse embryo tissue. Fig. S7. Cell clustering and cell type annotation in the human colon cancer tissue. Fig. S8. Expression level of marker genes in the human colon cancer tissue.

### **Peer review information**

Wenjing She and Claudia Feng were the primary editors of this article and managed its editorial process and peer review in collaboration with the rest of the editorial team. The peer-review history is available in the online version of this article.

### **Authors' contributions**

W.H. and Z.J. conceived the study. H.Z. and Z.J. developed the `ctSVG` software. H.Z. and X.S. conducted the analysis. H.Z., X.S., W.H., and Z.J. wrote the manuscript. All authors read and approved the final manuscript.

### **Funding**

H.Z. is supported by the National Institutes of Health under Award Number R35GM154865. Z.J. is supported by the National Institutes of Health under Award Number U54AG075936 and R35GM154865. W.H. is supported by the National Institutes of Health under Award Number R00HG011468 and R35GM150887.

### **Data availability**

All datasets used in this study were downloaded from the 10x website (<https://www.10xgenomics.com/datasets>). Xenium human lung cancer dataset was downloaded from: <https://www.10xgenomics.com/datasets/preview-data-ff-pe-human-lung-cancer-with-xenium-multimodal-cell-segmentation-1-standard> [33]. Xenium human pancreas cancer dataset was downloaded from: <https://www.10xgenomics.com/datasets/ffpe-human-pancreas-with-xenium-multimodal-cell-segmentation-1-standard> [34]. Xenium mouse colon dataset was downloaded from: <https://www.10xgenomics.com/datasets/fresh-frozen-mouse-colon-with-xenium-multimodal-cell-segmentation-1-standard> [35]. Visium HD human colorectal cancer dataset was downloaded from: <https://www.10xgenomics.com/datasets/visium-hd-cytassist-gene-expression-libraries-of-human-crc> [36]. Visium HD human lung cancer dataset was downloaded from: <https://www.10xgenomics.com/datasets/visium-hd-cytassist-gene-expression-libraries-of-human-lung-cancer-if> [37]. Visium HD human pancreas dataset was downloaded from: <https://www.10xgenomics.com/datasets/visium-hd-cytassist-gene-expression-libraries-human-pancreas> [38]. Visium HD mouse brain dataset was downloaded from: <https://www.10xgenomics.com/datasets/visium-hd-cytassist-gene-expression-libraries-of-mouse-brain-he> [39]. Visium HD mouse embryo dataset was downloaded from: <https://www.10xgenomics.com/datasets/visium-hd-cytassist-gene-expression-libraries-of-mouse-embryo> [40]. Visium HD mouse small intestine dataset was downloaded from: <https://www.10xgenomics.com/datasets/visium-hd-cytassist-gene-expression-libraries-of-mouse-intestine> [41]. Visium HD mouse kidney dataset was downloaded from: <https://www.10xgenomics.com/datasets/visium-hd-cytassist-gene-expression-libraries-of-mouse-kidney> [42]. The R package `ctSVG`, along with a detailed user manual, is publicly available at <https://github.com/haotian-zhuang/ctSVG> [43] and Zenodo [44] under the MIT license. The source code to reproduce the results in this paper is available at [https://github.com/haotian-zhuang/ctSVG\\_Paper](https://github.com/haotian-zhuang/ctSVG_Paper) and Zenodo [44] under the MIT license. Figure 2a was created using BioRender (BioRender.com).

### **Declarations**

#### **Ethics approval and consent to participate**

Not applicable.

#### **Consent for publication**

Not applicable.

### Competing interests

The authors declare no competing interests.

### Author details

<sup>1</sup>Department of Biostatistics and Bioinformatics, Duke University School of Medicine, Durham, NC, USA

<sup>2</sup>Department of Biostatistics, Columbia University Mailman School of Public Health, New York City, NY, USA

Received: 16 December 2024 / Accepted: 12 November 2025

Published online: 08 December 2025

### References

1. Svensson V, Teichmann SA, Stegle O. Spatialde: identification of spatially variable genes. *Nat Methods*. 2018;15(5):343–6.
2. Sun S, Zhu J, Zhou X. Statistical analysis of spatial expression patterns for spatially resolved transcriptomic studies. *Nat Methods*. 2020;17(2):193–200.
3. Dries R, Zhu Q, Dong R, Eng CHL, Li H, Liu K, et al. Giotto: a toolbox for integrative analysis and visualization of spatial expression data. *Genome Biol*. 2021;22:1–13.
4. Weber LM, Saha A, Datta A, Hansen KD, Hicks SC. Nnsvgl for the scalable identification of spatially variable genes using nearest-neighbor Gaussian processes. *Nat Commun*. 2023;14(1):4059.
5. Miller BF, Bambah-Mukku D, Dulac C, Zhuang X, Fan J. Characterizing spatial gene expression heterogeneity in spatially resolved single-cell transcriptomic data with nonuniform cellular densities. *Genome Res*. 2021;31(10):1843–55.
6. Zhuang H, Ji Z. PreTSA: computationally efficient modeling of temporal and spatial gene expression patterns. *bioRxiv*. 2024. <https://doi.org/10.1101/2024.03.20.585926>.
7. Oliveira MFD, Romero JP, Chung M, Williams SR, Gottscho AD, Gupta A, et al. High-definition spatial transcriptomic profiling of immune cell populations in colorectal cancer. *Nat Genet*. 2025;57(6):1512–23.
8. Hou W, Ji Z, Chen Z, Wherry EJ, Hicks SC, Ji H. A statistical framework for differential pseudotime analysis with multiple single-cell RNA-seq samples. *Nat Commun*. 2023;14(1):7286.
9. Polański K, Bartolomé-Casado R, Sarropoulos I, Xu C, England N, Jahnsen FL, et al. Bin2cell reconstructs cells from high resolution Visium HD data. *Bioinformatics*. 2024;40(9):btae546.
10. Hao Y, Hao S, Andersen-Nissen E, Mauck WM III, Zheng S, Butler A, et al. Integrated analysis of multimodal single-cell data. *Cell*. 2021;184(13):3573–87.
11. Hou W, Ji Z. Assessing GPT-4 for cell type annotation in single-cell RNA-seq analysis. *Nat Methods*. 2024;21(8):1462–5.
12. Song D, Wang Q, Yan G, Liu T, Sun T, Li JJ. scDesign3 generates realistic in silico data for multimodal single-cell and spatial omics. *Nat Biotechnol*. 2024;42(2):247–52.
13. Chen C, Kim HJ, Yang P. Evaluating spatially variable gene detection methods for spatial transcriptomics data. *Genome Biol*. 2024;25(1):18.
14. Yu J, Luo X. Identification of cell-type-specific spatially variable genes accounting for excess zeros. *Bioinformatics*. 2022;38(17):4135–44.
15. Cable DM, Murray E, Shanmugam V, Zhang S, Zou LS, Diao M, et al. Cell type-specific inference of differential expression in spatial transcriptomics. *Nat Methods*. 2022;19(9):1076–87.
16. Yu S, Li WV. spVC for the detection and interpretation of spatial gene expression variation. *Genome Biol*. 2024;25(1):103.
17. Manabe T, Tatsumi K, Inoue M, Makinodan M, Yamauchi T, Makinodan E, et al. L3/Lhx8 is a pivotal factor for cholinergic differentiation of murine embryonic stem cells. *Cell Death Differ*. 2007;14(6):1080–5.
18. Zhao Y, Marin O, Hermes E, Powell A, Flames N, Palkovits M, et al. The LIM-homeobox gene Lhx8 is required for the development of many cholinergic neurons in the mouse forebrain. *Proc Natl Acad Sci USA*. 2003;100(15):9005–10.
19. Aruga J. The role of Zic genes in neural development. *Mol Cell Neurosci*. 2004;26(2):205–21.
20. Inoue T, Ota M, Ogawa M, Mikoshiba K, Aruga J. Zic1 and Zic3 regulate medial forebrain development through expansion of neuronal progenitors. *J Neurosci*. 2007;27(20):5461–73.
21. Sankar S, Yellajoshyula D, Zhang B, Teets B, Rockweiler N, Kroll KL. Gene regulatory networks in neural cell fate acquisition from genome-wide chromatin association of Geminin and Zic1. *Sci Rep*. 2016;6(1):37412.
22. Liang X, Song MR, Xu Z, Lanuza GM, Liu Y, Zhuang T, et al. Isl1 is required for multiple aspects of motor neuron development. *Mol Cell Neurosci*. 2011;47(3):215–22.
23. Filova I, Pysanen K, Tavakoli M, Vochyanova S, Dvorakova M, Bohuslavova R, et al. ISL1 is necessary for auditory neuron development and contributes toward tonotopic organization. *Proc Natl Acad Sci USA*. 2022;119(37):e2207433119.
24. Zhang Q, Huang R, Ye Y, Guo X, Lu J, Zhu F, et al. Temporal requirements for ISL1 in sympathetic neuron proliferation, differentiation, and diversification. *Cell Death Dis*. 2018;9(2):247.
25. Zhou C, Yang G, Chen M, Wang C, He L, Xiang L, et al. Lhx8 mediated Wnt and TGF $\beta$  pathways in tooth development and regeneration. *Biomaterials*. 2015;63:35–46.
26. Huang HL, Liu YM, Sung TY, Huang TC, Cheng YW, Liou JP, et al. TIMP3 expression associates with prognosis in colorectal cancer and its novel arylsulfonamide inducer, MPT0B390, inhibits tumor growth, metastasis and angiogenesis. *Theranostics*. 2019;9(22):6676.
27. Su CW, Lin CW, Yang WE, Yang SF. TIMP-3 as a therapeutic target for cancer. *Ther Adv Med Oncol*. 2019;11:1758835919864247.
28. Yang M, Zhang X, Liu Q, Niu T, Jiang L, Li H, et al. Knocking out matrix metalloproteinase 12 causes the accumulation of M2 macrophages in intestinal tumor microenvironment of mice. *Cancer Immunol Immunother*. 2020;69:1409–21.
29. Qi J, Sun H, Zhang Y, Wang Z, Xun Z, Li Z, et al. Single-cell and spatial analysis reveal interaction of FAP+ fibroblasts and SPP1+ macrophages in colorectal cancer. *Nat Commun*. 2022;13(1):1742.
30. Schmidt U, Weigert M, Broaddus C, Myers G. Cell detection with star-convex polygons. In: *Medical Image Computing and Computer Assisted Intervention—MICCAI 2018: 21st International Conference, Granada, Spain, September 16–20, 2018, Proceedings, Part II* 11. Springer; 2018. pp. 265–73.

31. Benjamini Y, Hochberg Y. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *J Roy Stat Soc Ser B (Methodol)*. 1995;57(1):289–300.
32. Zhuang H, Wang H, Ji Z. FindPC: an R package to automatically select the number of principal components in single-cell analysis. *Bioinformatics*. 2022;38(10):2949–51.
33. 10x Genomics. FFPE Human Lung Cancer with Xenium Multimodal Cell Segmentation. 2024. Datasets. 10x Genomics. <https://www.10xgenomics.com/datasets/preview-data-ffpe-human-lung-cancer-with-xenium-multimodal-cell-segmentation-1-standard>. Accessed 3 Dec 2025.
34. 10x Genomics. FFPE Human Pancreas with Xenium Multimodal Cell Segmentation. 2024. Datasets. 10x Genomics. <https://www.10xgenomics.com/datasets/ffpe-human-pancreas-with-xenium-multimodal-cell-segmentation-1-standard>. Accessed 3 Dec 2025.
35. 10x Genomics. Fresh Frozen Mouse Colon with Xenium Multimodal Cell Segmentation. 2024. Datasets. 10x Genomics. <https://www.10xgenomics.com/datasets/fresh-frozen-mouse-colon-with-xenium-multimodal-cell-segmentation-1-standard>. Accessed 3 Dec 2025.
36. 10x Genomics. Visium HD Spatial Gene Expression Library, Human Colorectal Cancer (FFPE). 2024. Datasets. 10x Genomics. <https://www.10xgenomics.com/datasets/visium-hd-cytassist-gene-expression-libraries-of-human-crc>. Accessed 3 Dec 2025.
37. 10x Genomics. Visium HD Spatial Gene Expression Library, Human Lung Cancer, IF Stained (FFPE). 2024. Datasets. 10x Genomics. Accessed 3 Dec 2025. <https://www.10xgenomics.com/datasets/visium-hd-cytassist-gene-expression-libraries-of-human-lung-cancer-if>.
38. 10x Genomics. Visium HD Spatial Gene Expression Library, Human Pancreas (FFPE). 2024. Datasets. 10x Genomics. <https://www.10xgenomics.com/datasets/visium-hd-cytassist-gene-expression-libraries-human-pancreas>. Accessed 3 Dec 2025.
39. 10x Genomics. Visium HD Spatial Gene Expression Library, Mouse Brain (FFPE). 2024. Datasets. 10x Genomics. <https://www.10xgenomics.com/datasets/visium-hd-cytassist-gene-expression-libraries-of-mouse-brain-he>. Accessed 3 Dec 2025.
40. 10x Genomics. Visium HD Spatial Gene Expression Library, Mouse Embryo (FFPE). 2024. Datasets. 10x Genomics. <https://www.10xgenomics.com/datasets/visium-hd-cytassist-gene-expression-libraries-of-mouse-embryo>. Accessed 3 Dec 2025.
41. 10x Genomics. Visium HD Spatial Gene Expression Library, Mouse Small Intestine (FFPE). 2024. Datasets. 10x Genomics. <https://www.10xgenomics.com/datasets/visium-hd-cytassist-gene-expression-libraries-of-mouse-intestine>. Accessed 3 Dec 2025.
42. 10x Genomics. Visium HD Spatial Gene Expression Library, Mouse Kidney (FFPE). 2024. Datasets. 10x Genomics. <https://www.10xgenomics.com/datasets/visium-hd-cytassist-gene-expression-libraries-of-mouse-kidney>. Accessed 3 Dec 2025.
43. Zhuang H, Shang X, Hou W, Ji Z. Identifying cell-type-specific spatially variable genes with ctSVG. 2025. GitHub. <https://github.com/haotian-zhuang/ctSVG>. Accessed 3 Dec 2025.
44. Zhuang H, Shang X, Hou W, Ji Z. Identifying cell-type-specific spatially variable genes with ctSVG. 2025. Zenodo. <https://doi.org/10.5281/zenodo.17574358>.

## Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.