Categorization of 31 computational methods to detect spatially variable genes (SVGs) from spatial transcriptomics data

Jingyi Jessica Li

Department of Statistics and Data Science
University of California, Los Angeles

http://jsb.ucla.edu

joint work with Guan’ao Yan and Shuo Harper Hua
Spatially Transcriptomics Technologies

https://doi.org/10.1016/j.xinn.2022.100342
Highly Variable Genes (HVGs) vs. Spatially Variable Genes (SVGs)

**Informative features** to screen for before linear dimension reduction and Euclidean distance calculation

- **HVG detection**
  - Used in *single-cell* transcriptomics data analysis
  - Identifies genes with high expression variability across single cells
  - Helps in clustering cells and identifying subpopulations

- **SVG detection**
  - Used in *spatial* transcriptomics data analysis
  - Identifies genes with high expression variability across spatial locations
  - Helps in identifying spatial patterns and regions with distinct molecular signatures
Highly Variable Genes (HVGs) vs. Spatially Variable Genes (SVGs)

**Highly Variable genes (HVGs)**
- Single-cell transcriptomic data
- Cell clustering
- Cluster-marker gene

**Spatially variable genes (SVGs)**
- Spatial domain identification
- Domain-marker gene

**Gene expression**
- HVGs
- non-HVGs

**Spatially resolved transcriptomic data**
- spatial image spots

**Cell clustering**
- cell clusters

**Spatial domain identification**
- spatial domains
Highly Variable Genes (HVGs) vs. Spatially Variable Genes (SVGs)

Highly variable gene

Spatially variable gene

https://www.sc-best-practices.org/_images/svg.jpeg
There is no consensus in SVG definitions
Existing Review and Benchmark Studies

Review

• Adhikari et al., *Computational and Structural Biotechnology Journal*, 2024 (19 methods)

Benchmark studies

• Charitakis et al., *Genome Biology*, 2023 (6 methods)
• Chen et al., *Genome Biology*, 2024 (7 methods)
• Li et al., *bioRxiv*, 2023 (14 methods)

Categorization of SVG definitions is not the focus
Proposal: Three Categories of SVGs

1. **Overall SVGs:**
   - Informative genes for downstream analysis (e.g., spatial domain identification)

2. **Cell-type-specific SVGs:**
   - Revealing spatial variation within a cell type $\rightarrow$ cell subpopulations or states

3. **Spatial-domain-marker SVGs:**
   - Marker genes to annotate and interpret spatial domains already detected

Relationships among the three categories depends on

- Detection methods’ null and alternative hypotheses
SVG Categories: Overall, Cell-type-specific, and Spatial-domain-marker SVGs
Categorization of 31 SVG Detection Methods
Hierarchy of 31 SVG Detection Methods (Part 1: Three Categories)

- SVG detection methods
  - Overall SVGs
  - Spatial-domain-marker SVGs
  - Cell-type-specific SVGs

  - Statistical inference
    - Yes
      - Statistical inference type
        - Regression fixed-effect test
          - SpaGCN
          - DESpace
        - C-SIDE
        - CTSV
        - spVC
Hierarchy of 31 SVG Detection Methods (Part 2: Overall SVGs)

Overall SVGs

Graph conversion

No (Euclidean-space-based)

Yes (Graph-based)

Kernel-based patterns

No (Kernel-free)

Yes (Kernel-based)

Statistical inference

Statistical inference

Statistical inference

Statistical inference
Hierarchy of 31 SVG Detection Methods (Part 3: Kernel-free Methods)

- Statistical inference
  - No (Kernel-free)
    - MULTILAYER
      - sepal
      - BSP
      - PROST
  - Yes
    - Statistical inference type
      - Bayesian inference
        - BOOST-MI
        - BOOST-HMI
      - Regression fixed-effect test
        - SPADE
      - Dependence test
        - Trendsceek
Hierarchy of 31 SVG Detection Methods (Part 4: Kernel-based Methods)

- Yes (Kernel-based)
  - Statistical inference
    - Statistical inference type
      - Bayesian inference
      - Regression random-effect test
      - Dependence test
        - Gene expression distribution
        - BOOST-GP
        - singlecellHaystack
    - SPARK-X
Hierarchy of 31 SVG Detection Methods (Part 5: Kernel-based Methods)

- Regression random-effect test
- Gene expression distribution
- Gaussian
- SpatialDE
- nnSVG
- SOMDE
- SVCA
- Poisson
- SPARK
- Negative binomial
- GPCounts
Notations for SVG Detection (Per Gene)

For a given gene with expression levels measured at $n$ spatial spots

**Observed variables at spot $i = 1, \ldots, n$**

- Gene expression level
  - $y_i \in \mathbb{R}$
  - $Y_i \in \mathbb{R}$: random variable notation
- 2D spatial location
  - $s_i = (s_{i1}, s_{i2})^\top \in \mathbb{R}^2$
  - $s = [s_1, \ldots, s_n]^\top \in \mathbb{R}^{n \times 2}$: spatial location matrix

**Inferred variables at spot $i = 1, \ldots, n$**

- Spatial-domain indicator vector
  - $d_i = (d_{i1}, \ldots, d_{iL})^\top \in \{0, 1\}^L$, with $\sum_{l=1}^L d_{il} = 1$
- Cell-type proportion vector
  - $c_i = (c_{i1}, \ldots, c_{iK})^\top \in [0, 1]^K$, with $\sum_{k=1}^K c_{ik} = 1$
Among the 31 SVG detection methods, 21 use frequentist inference to detect SVGs:

- Define a test statistic
- Derive the test statistic’s null distribution
- Convert the test statistic value to a p-value

Types of null hypotheses:

- **Dependence tests**: a gene’s expression level is independent of spatial location
- **Regression-based tests**: spatial location has no “effect” on a gene’s expression level
  - Fixed-effect tests
  - Random-effect tests (variance component tests)
Dependence Tests

Null hypothesis:

\[ H_0 : Y \perp S \]

Assume that \((y_1, s_1), \ldots, (y_n, s_n)\) are independently sampled from the distribution of \((Y, S)\).

If \(H_0\) is rejected, the gene is detected as an overall SVG.

Nine methods adopt the dependence test formulation:

- **Conventional test statistics** (with theoretical null distribution):
  - SPARK-X, Hotspot, MERINGUE, BinSpect, scGCO

- **Unconventional test statistics** (with permutation-based null distribution):
  - Trendsceek, singlecellHaystack, RayleighSelection, SpaGene
SPARK-X (Zhu et al., Genome Biology, 2021)

**SPARK-X**: a non-parametric test that compares two $n \times n$ spot similarity matrices:

- Matrix 1 based on the gene’s expression levels at the $n$ spots
- Matrix 2 based on the kernel-transformed spatial locations of the $n$ spots
**SPARK-X** (Zhu et al., Genome Biology, 2021)

**SPARK-X**: a non-parametric test that compares two $n \times n$ spot similarity matrices:

- Matrix 1 based on the gene’s expression levels at the $n$ spots
- Matrix 2 based on the kernel-transformed spatial locations of the $n$ spots

To detect diverse spatial patterns, SPARK-X transforms the spatial locations $s_i = (s_{i1}, s_{i2})$, $i = 1, \ldots, n$, using two kernel-based functions:

- Gaussian transformation $s'_{il} = \exp\left(\frac{-s_{il}^2}{2\sigma_l^2}\right)$, $l = 1, 2$, to detect clustered or focal patterns
- Cosine transformation $s'_{il} = \cos\left(\frac{2\pi s_{il}}{\phi_l}\right)$, $l = 1, 2$, to detect periodic patterns

where $\sigma_1$, $\sigma_2$, $\phi_1$, and $\phi_2$ are tuning parameters.

Test statistic: Pearson correlation of the two matrices

Theoretical null: mixture chi-square distribution
**SPARK-X (Zhu et al., Genome Biology, 2021)**

**SPARK-X**: a non-parametric test that compares two $n \times n$ spot similarity matrices:

- Matrix 1 based on the gene’s expression levels at the $n$ spots
- Matrix 2 based on the kernel-transformed spatial locations of the $n$ spots

To detect diverse spatial patterns, SPARK-X transforms the spatial locations $s_i = (s_{i1}, s_{i2})$, $i = 1, \ldots, n$, using two kernel-based functions:

- Gaussian transformation $s_{il}' = \exp\left(\frac{-s_{il}^2}{2\sigma_l^2}\right)$, $l = 1, 2$, to detect clustered or focal patterns
- Cosine transformation $s_{il}' = \cos\left(\frac{2\pi s_{il}}{\phi_l}\right)$, $l = 1, 2$, to detect periodic patterns

where $\sigma_1$, $\sigma_2$, $\phi_1$, and $\phi_2$ are tuning parameters

**Test statistic**: Pearson correlation of the two matrices

**Theoretical null**: mixture chi-square distribution
singlecellHaystack: a unconventional test involves two pre-processing steps:

- Binarize the gene’s expression levels at spots into two states: detected and undetected
- Divide the 2D Euclidean space into grid points as coarse spatial coordinates
**singlecellHaystack** (Vandenbon and Diez, Nature Communications, 2020)

**singlecellHaystack**: a unconventional test involves two pre-processing steps:

- Binarize the gene’s expression levels at spots into **two states**: detected and undetected
- Divide the 2D Euclidean space into **grid points** as coarse spatial coordinates

**singlecellHaystack** uses a 2D independent Gaussian kernel, assuming independence of the two dimensions of the Euclidean space, to define **three distributions** of grid points:

- A **reference distribution** based on all grid points
- A **conditional distribution** based on grid points in the detected state
- Another **conditional distribution** based on grid points in the undetected state

**Test statistic**: sum of Kullback-Leibler divergences of the two conditional distributions from the reference distribution

**Permutation null**
**singlecellHaystack** (Vandenbon and Diez, Nature Communications, 2020)

**singlecellHaystack**: a unconventional test involves two pre-processing steps:

- Binarize the gene’s expression levels at spots into **two states**: detected and undetected
- Divide the 2D Euclidean space into **grid points** as coarse spatial coordinates

**singlecellHaystack** uses a 2D independent Gaussian kernel, assuming independence of the two dimensions of the Euclidean space, to define **three distributions** of grid points:

- A **reference distribution** based on all grid points
- A **conditional distribution** based on grid points in the detected state
- Another **conditional distribution** based on grid points in the undetected state

**Test statistic**: sum of Kullback-Leibler divergences of the two conditional distributions from the reference distribution

**Permutation null**
Regression-based Tests

Two types: **fixed-effect tests** and **random-effect tests**

**Linear mixed-effect model (LMM)** for a given gene:

\[ Y_i = \beta_0 + \mathbf{x}_i^\top \beta + \mathbf{z}_i^\top \gamma + \epsilon_i \]

- \( Y_i \): a gene’s expression level at spot \( i \) (response variable)
- \( \beta_0 \): (fixed) intercept
- \( \mathbf{x}_i \in \mathbb{R}^p \): fixed-effect covariates of spot \( i \)
- \( \beta \in \mathbb{R}^p \): fixed effects
Regression-based Tests

Two types: fixed-effect tests and random-effect tests

Linear mixed-effect model (LMM) for a given gene:

\[ Y_i = \beta_0 + x_i^\top \beta + z_i^\top \gamma + \epsilon_i \]

- \( Y_i \): a gene’s expression level at spot \( i \) (response variable)
- \( \beta_0 \): (fixed) intercept
- \( x_i \in \mathbb{R}^p \): fixed-effect covariates of spot \( i \)
- \( \beta \in \mathbb{R}^p \): fixed effects
- \( z_i \in \mathbb{R}^q \): random-effect covariates of spot \( i \)
- \( \gamma \in \mathbb{R}^q \): random effects with zero means \( \mathbb{E}[\gamma] = 0 \) and covariance matrix
  \[ \text{Cov}(\gamma) \in \mathbb{R}^{q \times q} \]
- \( \epsilon_i \): independent random error at spot \( i \) with \( \mathbb{E}[\epsilon_i] = 0 \)
- \( \gamma \perp \epsilon = (\epsilon_1, \ldots, \epsilon_n)^\top \)
Fixed-effect Tests

\[ Y_i = \beta_0 + x_i^\top \beta + z_i^\top \gamma + \epsilon_i \]

**Fixed-effect tests** examine whether \( x_i \) contribute to \( \mathbb{E}[Y_i] \)

If \( x_i \) makes no contribution, then \( \mathbb{E}[Y_i|x_i] = \mathbb{E}[Y_i] \)
Fixed-effect Tests

\[ Y_i = \beta_0 + x_i^\top \beta + z_i^\top \gamma + \epsilon_i \]

**Fixed-effect tests** examine whether \( x_i \) contribute to \( \mathbb{E}[Y_i] \)

If \( x_i \) makes no contribution, then \( \mathbb{E}[Y_i|x_i] = \mathbb{E}[Y_i] \)

**Null hypothesis**

\[ H_0 : \beta = 0 \]

implies \( \mathbb{E}[Y_i|x_i] = \mathbb{E}[Y_i], \ i = 1, \ldots, n \)
Random-effect Tests

\[ Y_i = \beta_0 + x_i^\top \beta + z_i^\top \gamma + \epsilon_i \]

**Random-effect tests** examine whether \( z_i \) contribute to \( \text{Var}(Y_i) \):

\[
\text{Var}(Y_i) = \text{Var}(\mathbb{E}[Y_i|z_i]) + \mathbb{E}[\text{Var}(Y_i|z_i)] = z_i^\top \text{Cov}(\gamma)z_i + \text{Var}(\epsilon_i)
\]

If \( z_i \) makes no contribution, then \( \text{Var}(\mathbb{E}[Y_i|z_i]) = 0 \)
Random-effect Tests

\[ Y_i = \beta_0 + \mathbf{x}_i^\top \beta + \mathbf{z}_i^\top \gamma + \epsilon_i \]

Random-effect tests examine whether \( \mathbf{z}_i \) contribute to \( \text{Var}(Y_i) \):

\[ \text{Var}(Y_i) = \text{Var}(\mathbb{E}[Y_i|\mathbf{z}_i]) + \mathbb{E}[\text{Var}(Y_i|\mathbf{z}_i)] = \mathbf{z}_i^\top \text{Cov}(\gamma) \mathbf{z}_i + \text{Var}(\epsilon_i) \]

If \( \mathbf{z}_i \) makes no contribution, then \( \text{Var}(\mathbb{E}[Y_i|\mathbf{z}_i]) = 0 \)

Null hypothesis

\[ H_0 : \text{Cov}(\gamma) = 0 \]

implies \( \text{Var}(\mathbb{E}[Y_i|\mathbf{z}_i]) = 0, \ i = 1, \ldots, n \)
Generalization of LMM

Assume $\epsilon_i \overset{iid}{\sim} N(0, \sigma^2)$ and $\gamma \perp \epsilon = (\epsilon_1, \ldots, \epsilon_n)^T$

$$Y_i = \beta_0 + x_i^T \beta + z_i^T \gamma + \epsilon_i \iff \begin{cases} Y_i | \mu_i \overset{\text{ind}}{\sim} N(\mu_i, \sigma^2) \\ \mu_i = \beta_0 + x_i^T \beta + z_i^T \gamma \end{cases}$$
Generalization of LMM

Assume $\epsilon_i \overset{iid}{\sim} N(0, \sigma^2)$ and $\gamma \perp \epsilon = (\epsilon_1, \ldots, \epsilon_n)^T$

$$Y_i = \beta_0 + x_i^T \beta + z_i^T \gamma + \epsilon_i \iff \begin{cases} Y_i | \mu_i \overset{\text{ind}}{\sim} N(\mu_i, \sigma^2) \\ \mu_i = \beta_0 + x_i^T \beta + z_i^T \gamma \end{cases}$$

Generalized LMM (GLMM): The distribution of $Y_i$ can be non-Gaussian

e.g., \[
\begin{cases} Y_i | \mu_i \overset{\text{ind}}{\sim} \text{Poisson}(\mu_i) \\ \log(\mu_i) = \beta_0 + x_i^T \beta + z_i^T \gamma \end{cases}
\]
Generalization of LMM

Assume $\epsilon_i \sim \text{iid } N(0, \sigma^2)$ and $\gamma \perp \epsilon = (\epsilon_1, \ldots, \epsilon_n)^T$

$$Y_i = \beta_0 + x_i^T \beta + z_i^T \gamma + \epsilon_i \iff \begin{cases} Y_i | \mu_i \sim \text{N}(\mu_i, \sigma^2) \\
\mu_i = \beta_0 + x_i^T \beta + z_i^T \gamma \end{cases}$$

**Generalized LMM (GLMM):** The distribution of $Y_i$ can be non-Gaussian

e.g., \begin{align*}
Y_i | \mu_i \sim \text{Poisson}(\mu_i) \\
\log(\mu_i) = \beta_0 + x_i^T \beta + z_i^T \gamma
\end{align*}

**Generalized non-parametric mixed-effect model:**

The effects of $x_i$ is modeled as non-parametric:

e.g., \begin{align*}
\log(\mu_i) = \beta_0 + f(x_i) + z_i^T \gamma
\end{align*}
Generalization of LMM

Assume $\epsilon_i \sim iid N(0, \sigma^2)$ and $\gamma \perp \epsilon = (\epsilon_1, \ldots, \epsilon_n)^T$

$$Y_i = \beta_0 + x_i^T \beta + z_i^T \gamma + \epsilon_i \iff \begin{cases} Y_i | \mu_i \sim ind N(\mu_i, \sigma^2) \\ \mu_i = \beta_0 + x_i^T \beta + z_i^T \gamma \end{cases}$$

Generalized LMM (GLMM): The distribution of $Y_i$ can be non-Gaussian

e.g., $\begin{cases} Y_i | \mu_i \sim \text{Poisson}(\mu_i) \\ \log(\mu_i) = \beta_0 + x_i^T \beta + z_i^T \gamma \end{cases}$

Generalized non-parametric mixed-effect model:

The effects of $x_i$ is modeled as non-parametric:

e.g., $\log(\mu_i) = \beta_0 + f(x_i) + z_i^T \gamma$

Q: Is spatial location $s_i$ modeled as $x_i$ or $z_i$?
Fixed-effect Tests for SVG Detection

Six methods use regression fixed-effect tests, covering all three SVG categories:

- **Overall SVGs**: SPADE
  - $x_i$ includes $s_i$

- **Cell-type-specific SVGs**: C-SIDE, CTSV, and spCV
  - $x_i$ includes $s_i$ and $c_i$ (cell-type proportion vector)

- **Spatial-domain-marker SVGs**: SpaGCN and DESpace
  - $x_i$ includes $s_i$ and $d_i$ (spatial-domain indicator vector)
SPADE (Bae et al., Nucleic Acids Research, 2021)

SPADE: linear-model fixed-effect test that detects overall SVGs:

\[ \mu_i = \beta_0 + x_i(s)^\top \beta \]

- \( x_i(s) \): principal components of 512 features from a pre-trained convolutional neural network applied to the \( n \) spots’ spatial locations \( s \) in an H&E image

Null hypothesis:

\( H_0: \beta = 0 \)

If \( H_0 \) is rejected, the gene is detected as an overall SVG.
**SPADE (Bae et al., Nucleic Acids Research, 2021)**

**SPADE**: linear-model fixed-effect test that detects **overall SVGs**:

\[ \mu_i = \beta_0 + x_i(s)\top \beta \]

- \( x_i(s) \): principal components of 512 features from a pre-trained convolutional neural network applied to the \( n \) spots’ spatial locations \( s \) in an H&E image

**Null hypothesis**:

\[ H_0 : \beta = 0 \]

If \( H_0 \) is rejected, the gene is detected as an **overall SVG**
SPADE (Bae et al., Nucleic Acids Research, 2021)

**SPADE**: linear-model fixed-effect test that detects **overall SVGs**:

\[ \mu_i = \beta_0 + x_i(s)^\top \beta \]

- \( x_i(s) \): principal components of 512 features from a pre-trained convolutional neural network applied to the \( n \) spots’ spatial locations \( s \) in an H&E image

**Null hypothesis**:

\[ H_0 : \beta = 0 \]

If \( H_0 \) is rejected, the gene is detected as an **overall SVG**

**Test**: R package limma

spVC: fixed-effect test that detects **cell-type-specific SVGs**

Assume

\[ Y_i \mid \mu_i \overset{\text{ind}}{\sim} \text{Poisson}(\mu_i) \]

**Two-step procedure:**

1. A **reduced model** without interactive effects between \( c_i \) and \( s_i \):

\[
\log(\mu_i) = \beta_0 + \sum_{k=1}^{K} c_{ik} \beta_k + f_0(s_i)
\]

It tests two null hypotheses:

- \( H_0 : \beta = (\beta_1, \ldots, \beta_K)^T = 0 \) using the likelihood ratio test
- \( H_0 : f_0(\cdot) = 0 \) using the Wald test

If both null hypotheses are rejected, it proceeds to the second step
2. A full model with interactive effects between $c_i$ and $s_i$:

$$\log(\mu_i) = \beta_0 + \sum_{k=1}^{K} c_{ik} \beta_k + f_0(s_i) + \sum_{k=1}^{K} c_{ik} f_k(s_i)$$

It tests if any of the interactive effects $f_1(\cdot), \ldots, f_K(\cdot)$ are zero using the likelihood ratio test.
2. A **full model** with interactive effects between \( c_i \) and \( s_i \):

\[
\log(\mu_i) = \beta_0 + \sum_{k=1}^{K} c_{ik} \beta_k + f_0(s_i) + \sum_{k=1}^{K} c_{ik} f_k(s_i)
\]

It tests if any of the interactive effects \( f_1(\cdot), \ldots, f_K(\cdot) \) are zero using the likelihood ratio test.

If

\[
H_0 : f_k(\cdot) = 0
\]

is rejected, the gene is detected as a **SVG specific to cell type** \( k \)
DESpace (Cai et al., Bioinformatics, 2024)

**DESpace**: fixed-effect test that detects **spatial-domain-marker SVGs**

Assume

\[ Y_i \mid \mu_i \overset{\text{ind}}{\sim} \text{NegativeBinomial}(\mu_i, \phi) \]

\[ \log(\mu_i) = \beta_0 + \sum_{l=1}^{L} d_{il} \beta_l \]

where \( \beta_l \) indicates the effect of spatial domain \( l \)
**DESpace (Cai et al., Bioinformatics, 2024)**

**DESpace**: fixed-effect test that detects **spatial-domain-marker SVGs**

Assume

\[ Y_i | \mu_i^{\text{ind}} \sim \text{NegativeBinomial}(\mu_i, \phi) \]

\[ \log(\mu_i) = \beta_0 + \sum_{l=1}^{L} d_{il} \beta_l \]

where \( \beta_l \) indicates the effect of spatial domain \( l \)

If

\[ H_0 : \beta_l = 0 \]

is rejected, the gene is detected as a **marker SVG of spatial domain** \( l \)
Six methods use regression random-effect tests to detect overall SVGs:

\[ Y_i = \beta_0 + x_i^T \beta + z_i^T \gamma(s) + \epsilon_i \]
**Six methods** use regression random-effect tests to detect **overall SVGs**:

SpatialDE, nnSVG, SOMDE, SVCA, SPARK, and GPcounts

\[ Y_i = \beta_0 + x_i^\top \beta + z_i^\top \gamma(s) + \epsilon_i \]

With \( n \) spots, \( z_i = (z_{i1}, \ldots, z_{in})^\top \in \{0, 1\}^n \) is a binary indicator vector for spot \( i \) s.t.

\[ z_{ii} = 1; \quad z_{ij} = 0 \text{ if } j \neq i \]
Random-effect Tests for SVG Detection

**Six methods** use regression random-effect tests to detect overall SVGs:

\[
Y_i = \beta_0 + x_i^\top \beta + z_i^\top \gamma(s) + \epsilon_i
\]

With \( n \) spots, \( z_i = (z_{i1}, \ldots, z_{in})^\top \in \{0,1\}^n \) is a binary indicator vector for spot \( i \) s.t.

\[
z_{ii} = 1; \quad z_{ij} = 0 \text{ if } j \neq i
\]

Random-effect vector \( \gamma(s) = (\gamma_1(s_1), \ldots, \gamma_n(s_n))^\top \in \mathbb{R}^n \) has

\[
\gamma_i(s_i) \text{ indicating the random effect of } s_i
\]

\( \text{Cov}(\gamma(s)) \) is assumed to depend on the spatial proximity of \( s_1, \ldots, s_n \) via a kernel.
**Random-effect Tests for SVG Detection**

**Six methods** use regression random-effect tests to detect overall SVGs:

SpatialDE, nnSVG, SOMDE, SVCA, SPARK, and GPcounts

\[
Y_i = \beta_0 + x_i^T \beta + z_i^T \gamma(s) + \epsilon_i
\]

With \( n \) spots, \( z_i = (z_{i1}, \ldots, z_{in})^T \in \{0, 1\}^n \) is a binary indicator vector for spot \( i \) s.t.

\[
z_{ii} = 1; \quad z_{ij} = 0 \text{ if } j \neq i
\]

Random-effect vector \( \gamma(s) = (\gamma_1(s_1), \ldots, \gamma_n(s_n))^T \in \mathbb{R}^n \) has

\[
\gamma_i(s_i) \text{ indicating the random effect of } s_i
\]

\( \text{Cov}(\gamma(s)) \) is assumed to depend on the spatial proximity of \( s_1, \ldots, s_n \) via a kernel

If

\[
H_0 : \text{Cov}(\gamma(s)) = 0
\]

is rejected, the gene is detected as an overall SVG
SpatialDE (Svensson et al., Nature Methods, 2018)

**SpatialDE**: a linear random-effect model:

\[ Y_i = \beta_0 + z_i^T \gamma(s) + \epsilon_i \]

- The random errors \( \epsilon_1, \ldots, \epsilon_n \) \( \text{iid} \sim \mathcal{N}(0, \delta) \)
- The random effects \( \gamma(s) \sim \text{MVN}(0, \sigma_s^2 \cdot K(s)) \)
  The kernel matrix \( K(s) = [K(s_i, s_j)]_{n \times n} \) is specified by a kernel function \( K(\cdot, \cdot) \)

This model is essentially a **Gaussian process**

If \( H_0 : \sigma_s^2 = 0 \) is rejected, the gene is detected as an **overall SVG**
26 methods for detecting overall SVGs:

9 kernel-based methods vs. 17 other methods (kernel-free or graph-based)

Kernel-based methods have

- Higher specificity for targeted patterns
- Lower overall power for other patterns
Discussion: Challenges in Detecting Non-Global Expression Patterns

1. **Small regions of interests (ROIs)**
   - Spatial-domain-marker SVGs by first identifying ROIs as spatial domains (e.g., SpaGCN)

2. **Spatial-Domain-Specific SVGs**
   - Genes with spatial patterns in small ROIs but not marker genes
   - No existing methods

3. **Cell-Type-Specific SVGs**
   - Easily missed if cell types have small proportions
   - Existing methods’ model goodness-of-fit

4. **Sharp Expression Changes**
   - Genes with sharp changes at tissue layer boundaries (e.g., Belayer)
   - Adding H&E image can help refine tissue boundaries
1. Small regions of interests (ROIs)
   • Spatial-domain-marker SVGs by first identifying ROIs as spatial domains (e.g., SpaGCN)

2. Spatial-Domain-Specific SVGs
   • Genes with spatial patterns in small ROIs but not marker genes
   • No existing methods

3. Cell-Type-Specific SVGs
   • Easily missed if cell types have small proportions
   • Existing methods’ model goodness-of-fit

4. Sharp Expression Changes
   • Genes with sharp changes at tissue layer boundaries (e.g., Belayer)
   • Adding H&E image can help refine tissue boundaries

Future direction: Incorporate knowledge on “interesting genes” to improve specificity
Discussion: Scalability

1. Calculate a **summary statistic** for each gene.
2. Convert the summary statistic to a **p-value** (frequentist methods only)

**Summary Statistic Calculation** \((n: \text{number of spatial spots})\)

- Gaussian process: \(O(n^3)\) in SpatialDE and SPARK
- Nearest-neighbor Gaussian process approximation: \(O(n)\) in nnSVG

**p-value Conversion**

- Fast if closed-form null distribution is available (conventional statistics)
- Computationally intensive if by permutation (unconventional statistics)

**Improving Scalability**

- Use approximation algorithms to speed up summary statistic calculation
- Reduce number of permutations in the p-value conversion step
Future Direction 1: Accommodating Technological Differences

Two Key Differences:

- **Spatial Resolution**
  - Imaging-based Technologies: Single-cell or subcellular resolution
  - Sequencing-based Technologies: Multicellular level, coarser resolution

- **Positional Randomness**
  - Structured grids (e.g., Spatial Transcriptomics, 10x Visium)
  - Unstructured spots (e.g., Slide-seq, MERFISH, SeqFISH)
Future Direction 1: Accommodating Technological Differences

Two Key Differences:

- **Spatial Resolution**
  - Imaging-based Technologies: Single-cell or subcellular resolution
  - Sequencing-based Technologies: Multicellular level, coarser resolution

- **Positional Randomness**
  - Structured grids (e.g., Spatial Transcriptomics, 10x Visium)
  - Unstructured spots (e.g., Slide-seq, MERFISH, SeqFISH)

Current Limitations:

- Most SVG detection methods **lack consideration** of these technological differences
- **Lack of consensus** on pre-processing and modeling SRT data
Future Direction 2: Enhancing Statistical Rigor and Method Benchmarking

Challenges:

- **Double-dipping**: Same data analyzed more than once, leading to confirmation bias
- **Example**: Spatial-domain-marker SVG detection

Strategies:

- Use *in silico* negative control data to remove spurious discoveries (e.g., ClusterDE)
- Develop fast **visualization** tools for interpreting top-detected SVGs

Method Benchmarking:

- Benchmarking requires well-annotated datasets with SVG **ground truths**
- Synthetic datasets and realistic **simulators** (e.g., SRTsim, scDesign3)
- No method is optimal in every aspect; benchmarking should be specific to data characteristics and align with biological questions
Yan, G., Hua, S. H., & Li, J. J. (2024). Categorization of 31 computational methods to detect spatially variable genes from spatially resolved transcriptomics data. *arXiv.*
https://arxiv.org/abs/2405.18779