ADVANCED STATISTICAL AND COMPUTATIONAL TECHNIQUES FOR GENOMIC DATA ANALYSIS

By

Sikta Das Adhikari

A DISSERTATION

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

Statistics—Doctor of Philosophy
Computational Mathematics, Science and Engineering—Dual Major

2024

ABSTRACT

The human body is an incredibly complex system that researchers have studied for decades to uncover its secrets. Genetic, transcriptomic, and epigenetic data each offer unique insights into its functioning. Recent technological advancements have enabled the generation of vast amounts of high-quality biological data, creating unprecedented opportunities to explore molecular mechanisms underlying health and disease. Analyzing these diverse datasets is crucial for developing targeted therapies, personalized medicine, and advancing our understanding of biology. Building advanced statistical and computational models to handle these complex datasets is now more important than ever for translating biological information into actionable insights and driving breakthroughs in medical research and treatment strategies.

In this dissertation, I first developed BayesKAT, a kernel-based testing methodology for assessing the association between user-defined groups of SNPs or genes and a phenotype of interest (Chapter 2). Unlike existing kernel-based tests that use predefined single or average kernels and often yield ambiguous results, this algorithm adaptively selects the optimal composite kernel using a Gaussian process model within a Bayesian framework, providing more interpretable outcomes.

Next, I explored the emerging field of spatial transcriptomics, where similar Gaussian process models and kernel-based testing have significant potential. To complement the recent surge in spatial transcriptomics research, Chapter 3 presents a comprehensive literature review of significant methodologies, particularly for spatial gene detection, which is a crucial step in spatial transcriptomics data analysis. This review provides an overview of the current state of research in the field.

In Chapter 4, I extended the kernel-based testing procedure to address challenges in spatial transcriptomic data. The newly developed algorithm, cSVG, not only detects spatially variable genes, but also improves spatial domain detection accuracy and addresses additional problems in this field.

Finally, to tackle the scarcity of crucial TF binding information for many transcription factors (TFs) across various cell types, I developed a computational model, 3D-TF-IMPUTE (Chapter

5). This model predicts TF binding sites by utilizing readily available epigenetic datasets and leveraging the three-dimensional structure of the genome in an unsupervised manner, efficiently predicting TF binding sites essential for understanding the functional genome.

By tackling key challenges in the analysis of genetic, transcriptomic, and epigenetic data, this dissertation makes significant contributions to the field. It provides powerful tools for researchers to better understand the molecular underpinnings of health and disease, paving the way for future breakthroughs in biomedical research.

ACKNOWLEDGEMENTS

First and foremost, I would like to express my deepest gratitude to my advisors, Dr. Jianrong Wang and Dr. Yuehua Cui, for their invaluable guidance and support throughout this journey. I am incredibly fortunate to have had such wonderful mentors during my PhD studies. Their doors were always open whenever I needed help, and their positive attitudes always helped me recover from setbacks and push through obstacles.

I am also profoundly grateful to my committee members Professor Lyudmila Sakhanenko, Professor Haolei Weng and Professor Rongrong Wang, for their insightful feedback on my research and professional guidance on my thesis. Their expertise and advice have been instrumental in shaping my work.

I would like to express my gratitude to the Department of Statistics and Probability, my primary department for the past five years. I am grateful to have been part of this department and to have had the opportunity to learn from exceptional professors. Special thanks to Dr. Haolei Weng and Dr. Yimin Xiao for teaching the core courses so beautifully. I feel fortunate to have had such outstanding teachers. I also want to thank the amazing STT staff, especially Tami and Andy, for solving all my problems so quickly.

My time in graduate school has been an incredible five-year journey, with the last 3.5 years spent as part of Dr. Jianrong Wang's lab. I extend my heartfelt thanks to my lab members: Dr. Jiaxin Yang, Dr. Hao Wang, Dr. Binbin Huang, Dr. Wenjie Qi, Dr. Jiwoong Kim, Dr. Pronoy Kanti Mondol, Tairan Song, Shuyang Zhang, and Xinrui Yu, for fostering a positive environment, always being helpful, and generously sharing their knowledge. Dr. Wang's passion for research and his positive mindset have been truly inspiring, often boosting my self-confidence and productivity. I have inherited a wealth of research techniques and biological concepts from him.

A special thank you goes to my husband, Arka, for his unwavering support and for being with me through all the highs and lows. This journey has been so fulfilling because we experienced it together. I am immensely grateful for the privilege of discussing new research papers, admiring each other's work, providing honest critiques, and cheering each other on after long, overwhelming

days.

The past five years have been the most enriching period of my life, and I have made incredible new friends along the way. Thanks to Sumegha, Hema, Tathagata, and Nisarg for introducing board games and many more wonderful things into my life. Anirban da and Sampriti, thank you for always being there, for our joint adventures, endless debates, and spontaneous plans. Your friendships have made this journey truly memorable, and I have found friends for life. Thanks to Nian, Haoxiang, and Sang Kyu for making our coursework time so enjoyable.

I would like to acknowledge my internship supervisor, Dr. Joyce Hsiao, and my mentor, Dr. Michael kleyman, who significantly enriched my learning journey and broadened my horizons. This internship provided me with the confidence to step out of my comfort zone and taught me how to plan and execute projects within limited time frames.

My heartfelt thanks go to my professors at the University of Calcutta—Prof. Manisha Pal, Prof. Asis Kumar Chattopadhyay, Prof. Uttam Bandyopadhyay, Prof. Sugata Sen Roy, Prof. Gaurangadeb Chattopadhyay, Prof. Nripes Kumar Mandal, Prof. Rahul Bhattacharya, and especially Prof. Bhaswati Ganguli—whose help, guidance and motivation inspired me to pursue a PhD.

Finally, I thank my parents, Alpana Das Adhikari and Samir Das Adhikari, for their emotional support and love, which have been the foundation of my achievements. I couldn't have come this far without their boundless generosity and wonderful parenting. I also thank my elder brother Saikat Das Adhikari and sister-in-law Arpita Dutta for their constant support. I thank them for always believing in me. I am grateful to my in-laws for always cheering me on and supporting me.

This journey has allowed me to gain expertise in my field of interest and acquire skills that have prepared me for my next career path. The personal growth I have experienced has been remarkable, and I am forever thankful to everyone who has helped me along the way.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

In contemporary biomedical research, the analysis of diverse biological datasets has emerged as a cornerstone in deciphering the intricate mechanisms of human diseases. Within the human body, comprised of trillions of cells, each cell type is specialized to execute distinct functions. At the core of cellular function lies the flow of genetic information, wherein DNA within the nucleus undergoes transcription into RNA, subsequently translated into proteins pivotal for various biological processes.

Throughout this transcription and translation cascade, distinct types of data are collected at each stage. Beginning with genome sequencing data extracted from DNA, researchers scrutinize variations such as single nucleotide polymorphisms (SNPs) to discern associations with diseases or phenotypic traits. Transitioning to transcriptomic data, derived from the second stage, unveils the remarkable diversity in gene expression among cells, despite harboring identical DNA sequences. Gene expression data analysis plays a crucial role in biomedical research by providing insights into the activity levels of genes within cells or tissues. Gene expression profiling can identify genes whose expression levels correlate with specific disease states or physiological conditions. These genes can serve as biomarkers for disease diagnosis, prognosis, and treatment response prediction.

In addition to genetic and transcriptomic datasets, epigenetic data play a pivotal role in understanding functional regulation. Epigenetic modifications, including DNA methylation, histone modifications, and transcription factor (TF) binding, orchestrate changes in gene expression crucial for the specialized functions of diverse cell types within various organs. These chemical modifications to DNA and associated proteins exert profound effects on gene expression, influencing cellular phenotypes and contributing to the complexity of human diseases.

Understanding the molecular underpinnings of human diseases requires comprehensive analysis of genetic, transcriptomic, and epigenetic datasets. This thesis addresses key challenges associated with these three types of datasets by developing advanced statistical and computational methodologies.

Genome-wide association studies (GWAS) have been instrumental over the past two decades in identifying millions of disease-associated single nucleotide polymorphisms (SNPs). However, many complex diseases and phenotypes are influenced by multiple genetic variants, where individual SNPs may only have a weak linear association with the phenotype. These SNPs can collectively contribute to the phenotype through their involvement in crucial biological processes or pathways. The problem of evaluating the strength of association between user-defined SNP groups and specific phenotypes or diseases is often addressed by semiparametric models and kernel-based tests. However, existing kernel-based testing methods rely on pre-specified single or average kernels, which can sometimes lead to inconsistent or ambiguous results. Chapter 2 introduces BayesKAT, a Bayesian optimal kernel-based test that adaptively selects the optimal composite kernel using a Gaussian process model within a Bayesian framework, providing more interpretable results. This algorithm is applicable not only to SNP-groups but also to sets of genes, whether user-specified or from genome-wide biologically important pathways. Based on a series of performance comparisons using both simulated and real large-scale genetics data, BayesKAT outperforms the available methods in detecting complex group-level associations.

Recent advancements in spatial transcriptomic technology have transformed the field of gene expression analysis, sparking a surge in research over the past few years. Given the extensive research in this field, Chapter 3 offers a detailed literature review, particularly on techniques for detecting spatially variable genes (SVGs), which is a crucial initial step in spatial transcriptomic data analysis. While various methods have been proposed for SVG detection, the significance of these genes lies in their contribution to downstream analyses, particularly in tasks like spatial domain detection. Conventional approaches typically rely on using all or a predetermined number of top-ranked SVGs for spatial domain detection. However, in datasets characterized by high diversity and a large number of SVGs, this strategy may not ensure accurate spatial domain detection or subsequent downstream analyses. Alternatively, grouping SVGs based on their expression patterns and leveraging all SVG groups in the downstream analysis can enhance accuracy, as seen in many examples. Furthermore, classifying SVGs in this manner is akin to identifying cell type marker genes, offering valuable biological insights. The challenge lies in accurately categorizing SVGs into relevant clusters, aggravated by the absence of ground truth and prior knowledge regarding the number and spectrum of spatial expression patterns exhibited by genes. Addressing this challenge, Chapter 4 introduces cSVG, a framework that begins with SVG detection and proceeds to precisely classify SVGs based on their spatial patterns by adjusting for confounding effects caused by shared cell types. Notably, this method eliminates the need for prior knowledge of gene cluster numbers, distinct spatial patterns, or cell type information. Through comprehensive simulation studies and real data analyses, this approach demonstrates considerable efficiency and holds promise as a potent tool in spatial transcriptomics analysis.

Transcription factors (TFs) are essential in the biological system as they are key regulators of gene expression. By binding to specific parts of DNA, TFs can activate or repress the transcription of target genes, thereby controlling the flow of genetic information from DNA to mRNA. TFs orchestrate complex gene networks, ensuring that genes are expressed at the right time, place, and levels, which is vital for maintaining cellular function. Understanding TF binding sites and their regulatory mechanisms is fundamental for insights into health, disease, and potential therapeutic interventions. Epigenetic data analysis often faces the challenge of unavailable or poor-quality TF binding data across various cell types and species. Accurately predicting TF binding sites remains a significant hurdle. Existing methods for predicting TF binding sites are typically based on supervised models, which involve computationally intensive preprocessing and training steps, and require various types of input data that may not be available for many cell types and species. Moreover, these methods overlook the three-dimensional structure of the genome, which holds valuable insights into underlying processes. To tackle these limitations, Chapter 5 presents a pioneering computational model, 3D-TF-IMPUTE. This method predicts TF binding sites for various TFs by leveraging the three-dimensional DNA structure, requiring only motif information and chromatin accessibility data for unsupervised prediction of TF binding sites.

This thesis addresses critical challenges in the analysis of genetic, transcriptomic, and epigenetic data, paving the way for a deeper understanding of disease biology and potential therapeutic interventions.

CHAPTER 2

BAYESKAT: BAYESIAN OPTIMAL KERNEL-BASED TEST FOR GENETIC ASSOCIATION STUDIES REVEAL JOINT GENETIC EFFECTS IN COMPLEX DISEASES

A modified version of part of this chapter was previously published in Briefings in Bioinformatics journal by Oxford University Press:

Sikta Das Adhikari, Yuehua Cui, Jianrong Wang, BayesKAT: bayesian optimal kernel-based test for genetic association studies reveals joint genetic effects in complex diseases, Briefings in Bioinformatics, Volume 25, Issue 3, May 2024, bbae182, https://doi.org/10.1093/bib/bbae182

2.1 Introduction

Deciphering the genetic basis of complex traits, such as the Alzheimer's disease, plays pivotal roles in functional genomics and precision medicine [1],[2]. Based on the advancement in high-throughput sequencing techniques, specific associated genetic variants, e.g., single-nucleotide polymorphisms (SNPs), have been identified for a large panel of phenotypes using Genome-wide Association Studies (GWAS) [3]. However, traditional GWAS approaches treat SNPs independently and can only discover individual SNPs that have strong marginal statistical associations with the phenotype of interest. It is well documented that many complex diseases and phenotypes are often associated with multiple genetic variants [4], [5], [6], [7] where an individual variant itself might be weakly associated with the phenotype. In contrast, groups of such SNPs may jointly contribute to the phenotype, potentially mediated via their cooperative participation in important biological processes or pathways [8], [9]. Therefore, the traditional GWAS framework of testing individual SNPs separately without considering the correlation structures and the potential interactions among SNPs may not capture the group-wise joint SNP effects. Separate testings of SNP associations by traditional GWAS approaches are also limited to reveal the underlying biological mechanisms of complex phenotypes. Alternative approaches based on multivariate regression significantly suffer from the large degrees of freedom in genome-wide association tests and can substantially lose the statistical power [10].

To overcome this critical challenge, kernel-based testing (KBT) framework has been introduced to test group-wise joint SNP effects [10], [11], [12], [13], [14], [15], [16]. By incorporating a kernel function to measure the similarity among genetic variants and compare with the phenotype similarities, the KBT framework simultaneously models the joint effects of multiple genetic variants. Wu et al. 2011 [12] first proposed the widely-used sequence kernel association test (SKAT) model to test rare-variant associations. As a supervised, versatile and computationally streamlined regression approach, SKAT accesses the associations between genetic variants within a specific region and the trait. As the outputs from the SKAT model, p-values of the statistical associations are generated, facilitating straightforward interpretations of the findings. An R package [17] has been developed for implementing different kinds of kernel-based testing models, including SKAT.

To enable novel discoveries of the genetic basis underlying complex diseases, maximizing statistical power in genome-wide association tests while effectively controlling type 1 errors is strongly desired. Under the KBT framework, statistical power heavily depends on the specific choice of kernel functions [16], [18], [19], [20]. However, the existing KBT models, including SKAT, require the kernel function to be specified a priori. Because the true functional relationship between the genetic variants and phenotypes is usually unknown in practice, selecting the ideal kernel function in advance for the KBT model, one that maximizes statistical power without increasing the type 1 error rate, poses statistical and computational challenges. One common approach that has been used is to repeat the KBT procedures based on different choices of kernels and then select the one resulting in the minimum p-value, which has been discussed by multiple studies [18], [21]. The major problem of this straightforward approach is the inflated type 1 error. Although data-dependent permutation or perturbation methods [18] can help ease the problem, they are not computationally scalable, especially when applied to high-dimensional datasets in large-scale genomic studies. An alternative approach is to use an equal-weighted average of multiple candidate kernels to form an averaged composite kernel [18], which performs better than the worst performing candidate kernel but does not usually achieve the performance of the optimal kernel function. Tests based on the average kernel approach may lead to inconsistent or incorrect conclusions in applications, as we will demonstrate below. He et al. 2018 [21] proposed a maximum kernel test model based on the U statistic, i.e. the mKU model, which claims to achieve the statistical power as close as to the best candidate kernel in high-dimensional settings under certain distributional assumptions. However, the specific distributional assumptions may not hold in practice and hence lead to inflated p-values, which will also be discussed in this study.

To further illustrate the significance and difficulty of choosing appropriate kernels in genetic association testings, Figure 2.1A demonstrates an example based on the genotype data for the trait of whole brain volume collected from the ADNI project for the Alzheimer's Disease Neuroimaging Initiative (https://adni.loni.usc.edu/)[22]. The group of genetic variants located in genes belonging to the caffeine metabolism pathway [23], [24], [25] are included into the kernel-based testing model to test the hypothesis that whether the caffeine metabolism pathway is associated with the whole brain volume phenotype. Using different kernel functions, the SKAT model leads to inconsistent conclusions. For instance, based on Quadratic kernel, the SKAT model rejects the null hypothesis (p-value<0.05), while the use of Gaussian kernel or IBS kernel does not lead to any rejection of the null hypothesis Figure 2.1A. On the other hand, using the equal-weighted average composite kernel, the SKAT model tends to reject the null hypothesis (p-value= 0.047). Since there is no clear mechanistic link between the caffeine metabolism pathway and whole brain volume, the rejection of the null hypothesis based on the Quadratic and the average composite kernels is likely a false discovery. To further quantify this issue, in Figure 2.1B, a cohort of total 500 replicate synthetic datasets based on the same covariates and genotype data is created, where the phenotype variables are generated by a Quadratic function $h(\cdot)$. Applying the SKAT test based on different kernel functions on the synthetic datasets, inconsistent testing results appear to be a persistent problem. Although the Quadratic kernel function leads to the correct hypothesis testing result as expected, the average composite kernel usually leads to incorrect conclusions (Figure 2.1B). As shown in the barplot of Figure 2.1B, using different kernels (Linear, Quadratic, Gaussian and average composite kernels) across the 500 synthetic datasets, inconsistent results are observed and the overall fractions of correct testing results are very low. Hence, the inconsistent conclusions based on different kernels suggest the fundamental need of developing a systematic data-adaptive approach of selecting appropriate kernel functions for KBT models in genome-wide association tests.

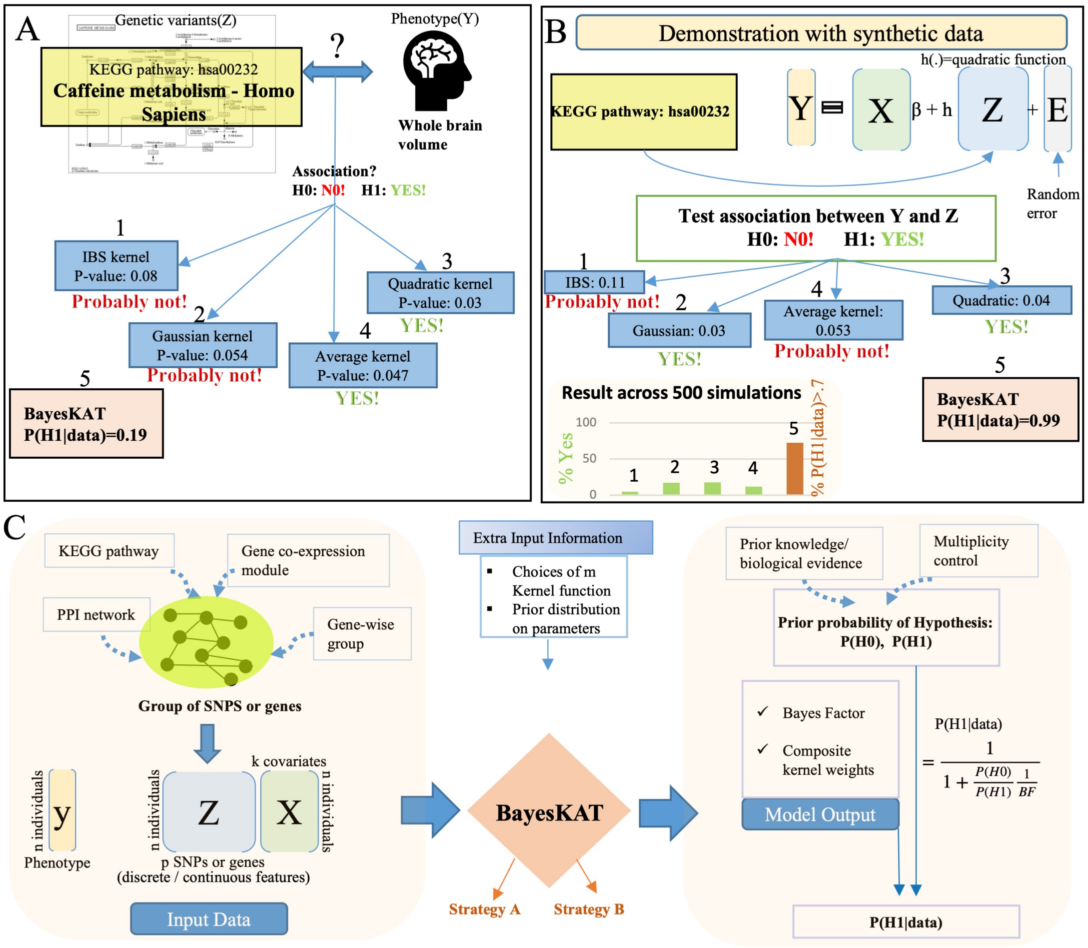
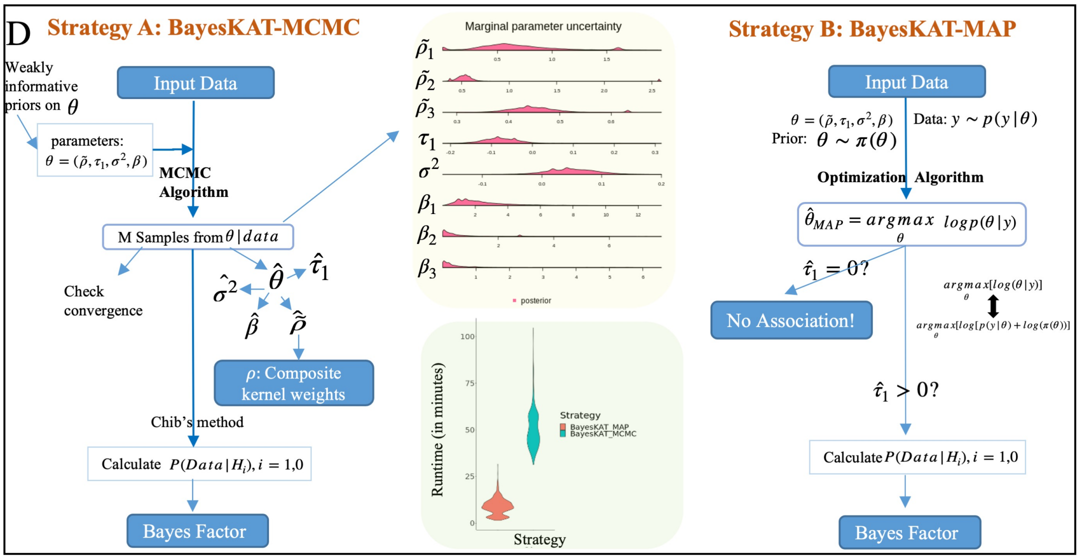


Figure 2.1 Overview of the significance and model design for BayesKAT. (A) Real-world example of association tests with inconsistent results depending on specific kernels. where BayesKAT offers a more interpretable metric. (B) Based on a synthetic data cohort simulated assuming a true quadratic function, different kernel functions lead to inconsistent results. Across 500 simulation replicates, each individual kernel yields inconsistent and ambiguous conclusions (barplot). In comparison, BayesKAT generates both interpretable and highly consistent results, with substantially boosted power. (C) Workflow of BayesKAT implementation for diverse types of genetic association tests to derive biological meaningful interpretations. (D) Model structures and the inference algorithms for the two BayesKAT strategies: BayesKAT-MCMC (left) and BayesKAT-MAP (right). BayesKAT-MCMC samples from posterior parameter distributions, providing a comprehensive view of the posterior parameter distributions. On the other hand, BayesKAT-MAP provides a more scalable solution, particularly well-suited for high-dimensional data.

Figure 2.1 (cont'd)


D
Strategy A: BayesKAT-MCMC
Weakly informative priors on θ
parameters: $\theta = (\tilde{\rho}, \tau_1, \sigma^2, \beta)$
Input Data
MCMC Algorithm
M Samples from $\theta | data$
Check convergence
$\hat{\theta}$
$\hat{\tau}_1$
$\hat{\sigma^2}$
$\hat{\beta}$
$\hat{\tilde{\rho}}$
ρ: Composite kernel weights
Chib's method
Calculate $P(Data | H_i), i = 1,0$
Bayes Factor
Marginal parameter uncertainty
$\tilde{\rho}_1$
$\tilde{\rho}_2$
$\tilde{\rho}_3$
τ_1
σ^2
β_1
β_2
β_3
posterior
Runtime (in minutes)
Strategy
BayesKAT_MAP
BayesKAT_MCMC
Strategy B: BayesKAT-MAP
Input Data
$\theta = (\tilde{\rho}, \tau_1, \sigma^2, \beta)$
Prior: $\theta \sim \pi(\theta)$
Data: $y \sim p(y | \theta)$
Optimization Algorithm
$\hat{\theta}_{MAP} = \underset{\theta}{argmax} \ log p(\theta | y)$
$\hat{\tau}_1 = 0?$
No Association!
$\underset{\theta}{argmax}[log(\theta | y)]$
$\underset{\theta}{argmax}[log[p(y | \theta) + log(\pi(\theta))]$
$\hat{\tau}_1 > 0?$
Calculate $P(Data | H_i), i = 1,0$
Bayes Factor

In this study, we developed a novel **Bayes**ian **K**ernel-Based **A**ssociation **T**esting algorithm, BayesKAT (https://github.com/wangjr03/BayesKAT). This algorithm effectively tackles the kernel selection challenge by choosing the optimal kernel in a data-adaptive way and calculating the posterior probability of association by evaluating the joint statistical associations of specific SNP groups with a complex phenotype. Moreover, compared to existing KBT-based methods, BayesKAT simultaneously achieves four goals in genome-wide association tests: (i) superior statistical power, by selecting the optimal kernel function based on the dataset under study; (ii) consistent results, by avoiding repeated tests based on a variety of different kernels; (iii) controlled type-1 error, without relying on unverified distributional assumptions or minimum p-value kernels; and (iv) strong computational scalability for high-dimensional and large-sample genome-wide data. Two alternative computational strategies, i.e., MCMC and MAP, are incorporated in BayesKAT, leading to additional implementation flexibilities for users. Extensively tested on a series of simulated datasets under different parameter settings, BayesKAT consistently demonstrates superior performance against existing methods. Furthermore, applied on the ADNI genotype datasets of the complex trait of whole brain volume (https://adni.loni.usc.edu), BayesKAT successfully discovered mechanistically related genes and biological pathways with higher accuracy. Specific genes and pathways related with neurodegenerative diseases, as reported by previous studies, are consistently prioritized by BayesKAT while not prioritized by other methods. Strikingly, BayesKAT is able to identify group-level SNP effects of novel co-expressed gene modules and protein complexes that potentially participate in the molecular processes modulating the whole brain volume phenotype. These algorithmic advantages and new biological discoveries robustly support the statistical innovation of BayesKAT and strongly highlight its effectiveness in decoding the genetic basis and associated molecular mechanisms underlying complex diseases.

2.2 Material and methods

2.2.1 Overview of kernel based testing models for genetic data

Under the kernel machine regression framework, continuous quantitative traits can be associated to genetic variants or molecular features, along with additional covariates, through a semiparametric model:

$$Y_i = X_i\beta + h(Z_i) + \epsilon_i, \qquad i = 1, 2, \cdots, n \tag{2.1}$$

where Y_i denotes the continuous value of the trait for the ith person in a sample of size n; $X_i = [X_{i1}, X_{i2}, \cdots, X_{ik}]$ is a set of k covariates for the ith individual that need to be controlled; and $\beta = [\beta_1, \beta_2, \cdots, \beta_k]$ are the corresponding effects of covariates. $Z_i = [Z_{i1}, Z_{i2}, \cdots, Z_{ip}]$ is the vector for the p genetic variants or molecular features, where Z_{ij} denotes the jth genetic variant or molecular level feature for the ith individual . The unknown errors ϵ_i are assumed to be independent and follow $N(0, \sigma^2)$, where the value σ^2 is also unknown. The most common genetic features are SNPs and the widely used molecular-level features include gene expressions. The features, i.e. $Z_{.j}, j = 1, 2, \cdots, p$ are associated with the trait, i.e. y, through an arbitrary function $h(\cdot)$ which is assumed to lie in a function space H_K generated by a kernel function $K(\cdot, \cdot)$. The supplementary files provide More discussion about different kernel function types.

It has been shown (see Appendix A) [11] that the kernel machine regression model in equation (2.1) is equivalent to the following linear mixed model:

$$Y = X\beta + h + \epsilon \tag{2.2}$$

where $\beta \in \mathbb{R}^k$ is a vector of effect sizes for covariates $X \in \mathbb{R}^{n\times k}$, h is an $n \times 1$ vector of random effects which is distributed as $h \sim N(0, \tau K)$, where K is the $n \times n$ kernel matrix and the error is distributed as $\epsilon \sim N(0, \sigma^2 I)$, where σ^2 is the error variance and τ is a variance component for the genetic effect.

The main goal is to test if the genetic variants have any combined effect on the outcome variable Y. Testing for the presence of group effect of Z is equivalent to testing the hypothesis $H_0 : \tau = 0$ vs. $H_1 : \tau > 0$. Choosing an appropriate kernel is crucial and that topic has been discussed further in the supplementary files. Most KBT methods, including SKAT, choose the kernel function first and then perform the testing based on the specified kernel function.

2.2.2 Importance of choosing appropriate Kernel

Although a variety of different kernels are available, for a given dataset, it is practically impossible to know a priori which kernel will fit the dataset best and maximize the testing power. Genetic data related to complex phenotypes pose particular challenges, primarily stemming from our limited understanding of how the interplay among genetic or molecular features influences their collective association with a phenotype. Therefore, choosing a kernel randomly can lead to a less powerful testing procedure for genome-wide applications. For example, if the outcome variable Y is related to the features through a Quadratic function, using a Linear kernel in the model will lead to weak tests that are not able to reject the null hypothesis even when the association is strong. On the other hand, by repeatedly applying KBT models based on different candidate kernels and choosing the one resulting in the minimum p-value, there is a high chance of making a false discovery, i.e., rejecting the null hypothesis when there is no association.

Combining a panel of candidate kernels together to create a composite kernel is thus a natural and effective strategy to overcome this issue. While a straightforward strategy of averaging kernels to form a composite kernel, i.e., a linear combination of candidate kernels with equal weights, can perform better than the worst-performing kernel function, it usually cannot perform as efficiently as the best kernel to accurately represent the association between the trait and features for a given dataset, thus, is not guaranteed to increase the statistical power. As shown in 2.1A and 2.1B, evaluated on both real and synthetic datasets, the average kernel strategy can often lead to incorrect and inconsistent results in practice. Thus, a systematic data-adaptive approach of optimal kernel selection is highly desirable for high-dimensional genome-wide association tests, especially for complex human disease phenotypes that are genetically modulated by multiple inter-dependent genetic variants.

Although a variety of different kernels are available, for a given dataset, it is practically impossible to know a priori which kernel will fit the data best and maximize the testing power. Genetic datasets of complex phenotypes are particularly challenging due to the limited prior knowledge about the structure of interdependence among genetic or molecular features and how the features are quantitatively associated with the phenotypes. Therefore, choosing a kernel randomly can lead to a less powerful testing procedure for genome-wide applications. For example, if the outcome variable Y is related to the features through a Quadratic function, using a Linear kernel in the model will lead to weak tests that are not able to reject the null hypothesis even when the association is strong. On the other hand, by repeatedly applying KBT models based on different candidate kernels and choosing the one resulting in the minimum p-value, there is a high chance of making a false discovery, i.e., rejecting the null hypothesis when there is no association.

2.2.3 BayesKAT

Our new algorithm, BayesKAT (https://github.com/wangjr03/BayesKAT) employs a novel Bayesian modeling strategy to automatically select the optimal composite kernel based on the data and does not require the composite kernel function to be set a priori by the user. Based on the inferred optimal composite kernel function, BayesKAT can efficiently test the joint effects induced by a group of genetic or molecular features associated with a phenotype. The optimal composite kernel is a linear combination of candidate kernels where the weight of each candidate kernel reflects the degree of usefulness of the kernel explaining the complex relationship between a group of features and the phenotype of interest. As an illustration, suppose there are three potential kernels: Quadratic, Gaussian, and IBS. If the IBS kernel effectively captures the underlying relationship, it will carry greater significance within the composite kernel, hence have a larger weight, while the impact of other kernels may be relatively weak, as indicated by their lower weights. Figure 2.1C provides an overview of the workflow of BayesKAT and its two computational strategies: 1) the Markov Chain Monte Carlo (MCMC) strategy; and 2) the Maximum a Posteriori (MAP) strategy. Additionally, it is noteworthy that while BayesKAT is primarily developed to test genetic associations, it can also be employed for a wide range applications, including testing the association between continuous gene expression features and complex traits.

Consider a set of m candidate kernels $K_1, K_2, \cdots, K_m$, the composite kernel is in the form of $\sum_{i=1}^{m} \rho_i K_i$, where $0 \leq \rho_i \leq 1$ and $\sum_{i=1}^{m} \rho_i$=1 and

$$h \sim N(0, \tau \textstyle\sum_{i=1}^{m} \rho_i K_i),$$

Therefore, selecting the optimal composite kernel is equivalent to selecting the optimal value for the weight ρ_i $(i = 1, \cdots, m)$ so that it can capture the underlying relationship between the genetic or molecular features and the trait, when testing the group-level effect of a set of multiple features.

As a kernel-based testing model, BayesKAT relies on a set of candidate kernel functions, which are incorporated to infer the optimal composite kernel for the association tests. For the convenience of practical implementations, BayesKAT infers the optimal composite kernel consisting of three candidate kernels as the default setting. And the default candidate kernels include Quadratic, Gaussian and IBS kernel. To construct a composite kernel, the candidate kernels are normalized in BayesKAT based on the previously proposed technique [21] so that they are in the same scale and comparable.

To gain robust performance, weakly informative prior distributions are used for model parameters by default, although the users can incorporate more informative priors based on specific knowledge about the data. The important model parameters are $\theta = [\tilde{\rho}, \tau_1, \sigma^2, \beta]$, where $\tilde{\rho} = [\tilde{\rho_1}, \tilde{\rho_2}, ..., \tilde{\rho_m}]$ are the unscaled weights of the candidate kernels ($\sum_{i=1}^{m} \tilde{\rho_i} \neq 1$), $\beta = (\beta_1, \beta_2, \cdots, \beta_k)^T$ and $\tau_1 = \frac{\tau}{\sigma^2}$ after reparameterization. And the weakly informative prior distributions are:

$$\sigma^2 \sim InverseGamma(2, 2),$$

$$\tau_1 \sim Uniform(0,2),$$

$$\beta \sim MultivariateNormal(0, 10I),$$

$$\tilde{\rho_1}, \tilde{\rho_2}, \tilde{\rho_3} \sim Gamma(1,1).$$

The actual weights for candidate kernels $\rho = (\rho_1, \rho_2, \rho_3) = (\tilde{\rho_1}, \tilde{\rho_2}, \tilde{\rho_3})/\sum_{i=1}^{3} \tilde{\rho_i}$. Clearly $\sum_{i=1}^{3} \rho_i = 1$ and $\rho \sim dirichlet(1,1,1)$. The data distribution is defined as:

$$y|\theta \sim N(X\beta, \sigma^2(\tau_1 K_c + I)), \tag{2.3}$$

where the composite kernel $K_c = \sum_{i=1}^{3} \rho_i K_i$.

2.2.3.1 BayesKAT strategy:

Here the main hypothesis to test is: $H_0 : \tau = 0$ vs. $H_1 : \tau > 0$. It is equivalent to test $H_0 : \tau_1 = 0$ vs. $H_1 : \tau_1 > 0$. Bayes factor(BF_{10}) is calculated to test the hypothesis, which evaluates the evidence in favor of the alternative hypothesis. Bayes factor is defined as the ratio of marginal likelihoods under two hypotheses [26]:

$$BF_{10} = \frac{P(Data|H_1)}{P(Data|H_0)} \tag{2.4}$$

where $P(Data|H_0)$ and $P(Data|H_1)$ are the marginal likelihoods under H_0 and H_1, respectively. Given the input data, BayesKAT mainly uses two efficient and easy-to-implement strategies to select the composite kernel and calculate the Bayes Factor BF_{10} by estimating $P(Data|H_0)$ and $P(Data|H_1)$. The two computational strategies are explained in subsequent sections.

2.2.3.2 Interpreting BayesKAT output:

The preliminary output from BayesKAT, BF_{10}, is a summary of evidence provided by the data in favor of H_1 as opposed to H_0. In addition, the posterior probability of the association ($P(H_1|Data)$) is calculated as the final output, which has a one-one relation with BF_{10}, i.e.,

$$\frac{P(H_1|Data)}{P(H_0|Data)} = \frac{P(H_1|Data)P(H_0)}{P(H_0|Data)P(H_1)} \times \frac{P(H_1)}{P(H_0)}$$

$$\Rightarrow P(H_1|Data) = \frac{1}{1+\frac{P(H_0)}{P(H_1)}\frac{1}{BF_{10}}}$$

Here $P(H_1)$ and $P(H_0)$ are the prior probabilities under H_1 and H_0, respectively, i.e. the probabilities of existence of association and no association, respectively. Depending on the specific biological problem and dataset, the values of $P(H_0)$ and $P(H_1)$ can be set given biological evidence or prior knowledge, and $P(H_1|Data)$ can be calculated. $P(H_1|Data)$ is the posterior probability of the model under H_1 given the data. $P(H_1|Data)$ gives a quantitative evaluation of how probable there exists an association between the genetic features and the phenotype of interest or how strong the evidence is against the null hypothesis.

As already demonstrated by the example in Figure 2.1A and 2.1B in the introduction section, existing methods based on pre-selected kernels or average composite kernels suffer from inconsistent hypothesis testing results and may lead to false discoveries. In contrast, for the Figure 2.1A scenario, BayesKAT calculated the posterior probability of a true association given the data P($H_1|data$)=0.19, i.e. the evidence of association is very low and the existence of true association is not very likely. This is consistent with the fact that there is a lack of documented evidence of the association between the caffeine metabolism pathway and the whole brain volume phenotype. Strikingly, in the scenario of the synthetic data cohort simulated with known association based on the Quadratic kernel (see Figure 2.1B), BayesKAT successfully calculated the posterior probability $P(H_1|Data) = 0.99$ to suggest the existence of association, without incorporating any prior information. Moreover, tested on 500 repeated simulation cohorts, BayesKAT achieves much higher statistical power than other methods and also demonstrates more consistent testing results (see Figure 2.1B). Investigating the inferred weights for each candidate kernel functions in the final composite kernel selected by BayesKAT further shows that the Quadratic kernel is correctly assigned with the largest weights (Supplementary Figure E.1), suggesting that BayesKAT can efficiently capture the functional form of the underlying statistical associations in a data adaptive way.

2.2.4 BayesKAT-MCMC

As a Bayesian model, BayesKAT-MCMC employs the Markov Chain Monte Carlo (MCMC) sampling-based strategy to infer the optimal composite kernel function. Leveraging MCMC for efficient and traceable samplings from complex target distributions, BayesKAT-MCMC avoids direct sampling from the posterior distribution $P(\theta|Data)$ (see section 2.2.6), which does not have a closed mathematical form and is computationally intractable. Instead, Metropolis-Hastings method [27, 28] is used to iteratively draw samples based on the generated Markov chain, which are able to approximate the target probability distribution $P(\theta|Data)$.

Let θ_0 denote the initial value for θ. The tth iteration of the Metropolis-Hastings algorithm consists of the following steps [29], [30]:

1. Sample a candidate point θ_t from a proposal distribution $J_t(\theta^*|\theta_{t-1})$.
2. Calculate the acceptance ratio for jumping to the new point $r_1 = \frac{p(\theta^*|Data)/J_t(\theta^*|\theta_{t-1})}{p(\theta_{t-1})/J_t(\theta_{t-1}|\theta^*)}$
3. Set $\theta_t = \theta^*$ with probability r_1 and $\theta_t = \theta_{t-1}$ with probability $1 - r_1$. That is, it jumps to the new proposed value with probability r_1 and stays at the same value with probability $1 - r_1$.

Here is the main workflow for BayesKAT-MCMC:

Input: Genotype matrix Z, covariate matrix X, response y.

Parameters under $H_1 : \tau_1 > 0$: $\theta_{H_1} = [\tilde{\rho}, \tau_1, \sigma^2, \beta]$

Parameters under $H_0 : \tau_1 = 0$: $\theta_{H_0} = [\sigma^2, \beta]$

Prior distribution of $\theta_{H_1}, \theta_{H_0}$ as mentioned in "BayesKAT" section.

Define the data distribution based on θ_{H_i}, X,Z and y, as defined in (2.3) where i =0 or 1.

Step 1: Using the Metropolis-Hastings MCMC method mentioned above, draw samples from the posterior distribution of θ_{H_1} using three separate MCMC chains, each of which ran 50,000 iterations.

Step 2: check if the algorithm has converged, otherwise run more iterations.

Step 3: Draw samples from the posterior distribution of θ_{H_1} by similarly Repeating step 1 and 2.

Step 4: Using the posterior samples of θ_{H_1}, the unknown parameters are estimated: $\hat{\theta_{H_1}} = [\hat{\tilde{\rho}}, \hat{\tau_1}, \hat{\sigma^2}, \hat{\beta}]$. From $\hat{\tilde{\rho}}$, the optimal kernel weights ρ can be estimated.

Step 5: the posterior samples of $\theta_{H_1}, \theta_{H_0}$ are used to calculate $P(data|H_1), P(data|H_0)$ respectively using Chib's method[31].

Step 6: Bayes Factor calculated from $P(data|H_1), P(data|H_0)$ using (2.4).

BayesKAT-MCMC uses the R package BayesianTools [32] to generate two sets of samples from the posterior distribution of θ under the hypotheses H_1 and H_0, using the Metropolis-Hastings algorithm in an adaptive way [33] to leverage the history of the stochastic process and appropriately fine-tune the proposal distributions. Three separate MCMC chains initiated from different random start points are generated for 50,000 iterations. To ensure that the MCMC chains are converged, trace plot is used to visualize the moves of the Markov chains in the state space [34] . In addition, based on the Gelman-Rubin diagnostic method [35], the potential scale reduction factor, i.e. PSRF score, is also calculated and presented at the end of MCMC sampling to inspect whether the chain is converged in which the PSRF score is close to one.

Based on the generated posterior samples, the marginal distributions of the parameters are further visualized as shown in Figure 2.1D. The posterior samples are used to estimate the composite kernel weights and also the marginal likelihoods $P(Data|H_i), i = 1, 0$ using the Chib's method [31]. Bayes Factor is subsequently calculated using the formula presented in equation (2.4).

2.2.5 BayesKAT-MAP

Although BayesKAT-MCMC yields comprehensive information of the marginal distributions of model parameters, drawing large sets of samples from the posterior distributions in the MCMC strategy is computationally expensive. Here, we provide an alternative strategy, termed BayesKAT-MAP, which is easy to implement, allows parallel calculations, and has higher computational scalability. Instead of drawing numerous MCMC samples from the posterior distributions and then estimating the parameter values, BayesKAT-MAP employs the quick optimization technique to estimate the parameters of interest directly, based on the Maximum A Posteriori (MAP) strategy such that,

$$\begin{aligned} \hat{\theta}_{MAP} &= \underset{\theta}{argmax}[\log p(\theta|y)] \\ &= \underset{\theta}{argmax}[\log p(y|\theta) + \log \pi(\theta)] \end{aligned} \tag{2.5}$$

where $\pi(\theta)$ denotes the prior distribution θ. Because the objective function is nondifferentiable at some points, a derivative-free optimization algorithm by Quadratic approximation using the

R package Minqa [36] is implemented. The most important model parameter τ_1 indicates the existence of association between the feature set and the trait variable, with $\tau_1 = 0$ suggesting that there is no association. In BayesKAT-MAP, if the calculated MAP estimator of τ_1, i.e., $\hat{\tau_1}_{MAP} = 0$, it follows that the Bayes Factor =0 (i.e., no evidence of association is found), and the computational process terminates. On the other hand, if $\hat{\tau_1}_{MAP} > 0$, it implies that there might be some evidence of association and BayesKAT-MAP proceeds to calculate the MAP estimator again under H_0 and then computes the marginal likelihoods $P(Data|H_i)$, $(i = 1, 0)$, along with the Bayes Factor.

Due to the practical limitations of exact analytical methods, such as relying on specific distributional assumptions, efficient numerical integration approaches [37] are needed to calculate the marginal likelihoods under hypotheses H_i, $P(Data|H_i) = \int Pr(Data|\theta, H_i)\ \pi(\theta|H_i)d\theta$, so that the model can be applied on diverse panels of data. BayesKAT-MAP employs the Laplace's method [38], [39], [26] for approximating the integral $T = \int Pr(Data|\theta, H_i)\pi(\theta|H_i)d\theta$ by $\hat{T}$, where $\hat{T} = (2\pi)^{d/2}|\tilde{\Sigma}|^{1/2}Pr(Data|\tilde{\theta}, H_i)\pi(\tilde{\theta}|H_i)$ and d is the dimension of θ, $\tilde{\theta}$ is the mode of the log-likelihood function $l(\theta|Data)$, $\tilde{\Sigma}$ is the inverse of the negative Hessian matrix of the second derivative of $l(\theta|Data)$ computed at $\tilde{\theta}$. For boundary regions of the parameter space, the Laplace approximation is modified according to the previously developed protocol [40] to accommodate the boundary cases. Based on the estimated marginal likelihood densities, the Bayes Factor is then computed and the posterior probabilities are finally inferred, given user-defined priors $p(H_0)$ and $p(H_1)$ for which equal values are used as the default setting in BayesKAT.

Here is the summary of the workflow for BayesKAT-MAP:

Input: Genotype matrix Z, covariate matrix X, response y.

Parameters under $H_1 : \tau_1 > 0$: $\theta_{H_1} = [\tilde{\rho}, \tau_1, \sigma^2, \beta]$

Parameters under $H_0 : \tau_1 = 0$: $\theta_{H_0} = [\sigma^2, \beta]$

Prior distribution of $\theta_{H_1}, \theta_{H_0}$ as mentioned in "BayesKAT" section.

Define the data distribution based on θ_{H_i}, X,Z and y as mentioned in (2.3), where i =0 or 1

Step 1: $\hat{\theta}_{H_1} = \hat{\theta}_{MAP}$ is calculated using this formulation 2.5 using optimization technique.

Step 2: check if $\hat{\tau}_1 = 0$.

If 0, stop. conclusion: no association!

If >0, go to next step.

Step 3: Calculate $\hat{\theta}_{H_0}$ using same technique in step 1 under H_0.

Step 4: Calculate $P(data|H_1), P(data|H_0)$ using Laplace approximation(look at previous paragraph) based on $\hat{\theta}_{H_1}$ and $\hat{\theta}_{H_0}$.

Step 5: Bayes Factor calculated from $P(data|H_1), P(data|H_0)$ using (2.4).

The performance and runtime of BayesKAT-MCMC and BayesKAT-MAP under different settings are systematically compared and further discussed in supplementary files. Because of the superior computational scalability of BayesKAT-MAP, as shown in 2.1D, the results in the paper are generated using BayesKAT-MAP.

2.2.6 Input data organization and model set up for BayesKAT

Data containing the information of genotypes or molecular features for complex phenotypes can be collected from large public-accessible or user-generated cohorts (e.g., ADNI, UK biobank, All of Us (https://allofus.nih.gov/), GTEx, PsychENCODE (https:// psychencode.synapse.org/), etc.). Individual-level data can be pre-processed and efficiently undergo steps of quality controls using software such as Plink [41]. Additionally, biological meaningful feature groups need to be defined and created depending on the goals of genome-wide association tests. In this study, we have explored four different biology inspired ways of grouping functionally related genetic variants, including 1) gene-wise groups: aggregating SNPs located within genomic regions of genes; 2) pathway-level groups: aggregating SNPs situated within genes that belong to a specific molecular pathway; 3) co-expression gene modules: aggregating SNPs located within genes that belong to a specific co-expression module; and 4) Protein-protein interaction(PPI) modules: aggregating SNPs located within genes that belong to a specific PPI module, which may represent a protein complex.

2.2.7 Comparison with other methods and performance evaluation

The performance of BayesKAT is compared to two state-of-the-art algorithms: 1) SKAT using the average composite kernel, denoted as SKAT(Avg) [18]; and 2) the U statistic-based method, denoted as mKU [21]. Both of these two methods are frequentist approaches that maximize the power after restricting the type 1 error to a fixed level of α, such as setting α to 0.05, and have been shown to outperform other existing methods. In contrast, as the first Bayesian model for this problem, BayesKAT uses a fixed threshold on the posterior probability or the Bayes factor, based on previously suggested guidelines [26], to reject the null hypothesis. To make fair comparisons, the performance of each method (i.e., the empirical statistical power), is evaluated at a fixed and equal empirical type 1 error across all three algorithms. A systematic comparison based on rigorous simulations are presented in the Results section. As multiple groups are simultaneously tested, a multiplicity correction technique is implemented, which is discussed in detail in the supplementary files.

2.3 Enhanced efficacy of BayesKAT benchmarked on simulation studies

As defined in the section of Materials and Methods, the rows of the feature matrix $Z \in \mathbb{R}^{n\times p}$ correspond to n individuals and the columns correspond to the p features. Depending on the particular genetic association studies, the features may encompass discrete genetic characteristics, like alleles with values of 0, 1, or 2, or continuous molecular features, such as gene expressions. We have conducted simulations for both cases, with Z corresponding to discrete or continuous features, under both low and high dimensional settings.

2.3.1 Simulation with continuous features:

As shown in Figure 2.2A, the simulations based on continuous features are first conducted to evaluate the performance of BayesKAT using similar scenarios presented in [21]. With the specified parameters ($n = 500$, $p = 100$, $k = 2$, $m = 3$), the feature matrix Z is simulated from a multivariate normal distribution with mean 0_p and an AR(1) correlation matrix R where $(R(j, j') = r^{|j-j'|})$. In the simulation, the correlation r is set to be 0.6. The covariate matrix $X \in \mathbb{R}^{n\times 2}$ has one binary covariate generated from a Bernoulli (0.6) distribution and one continuous covariate generated

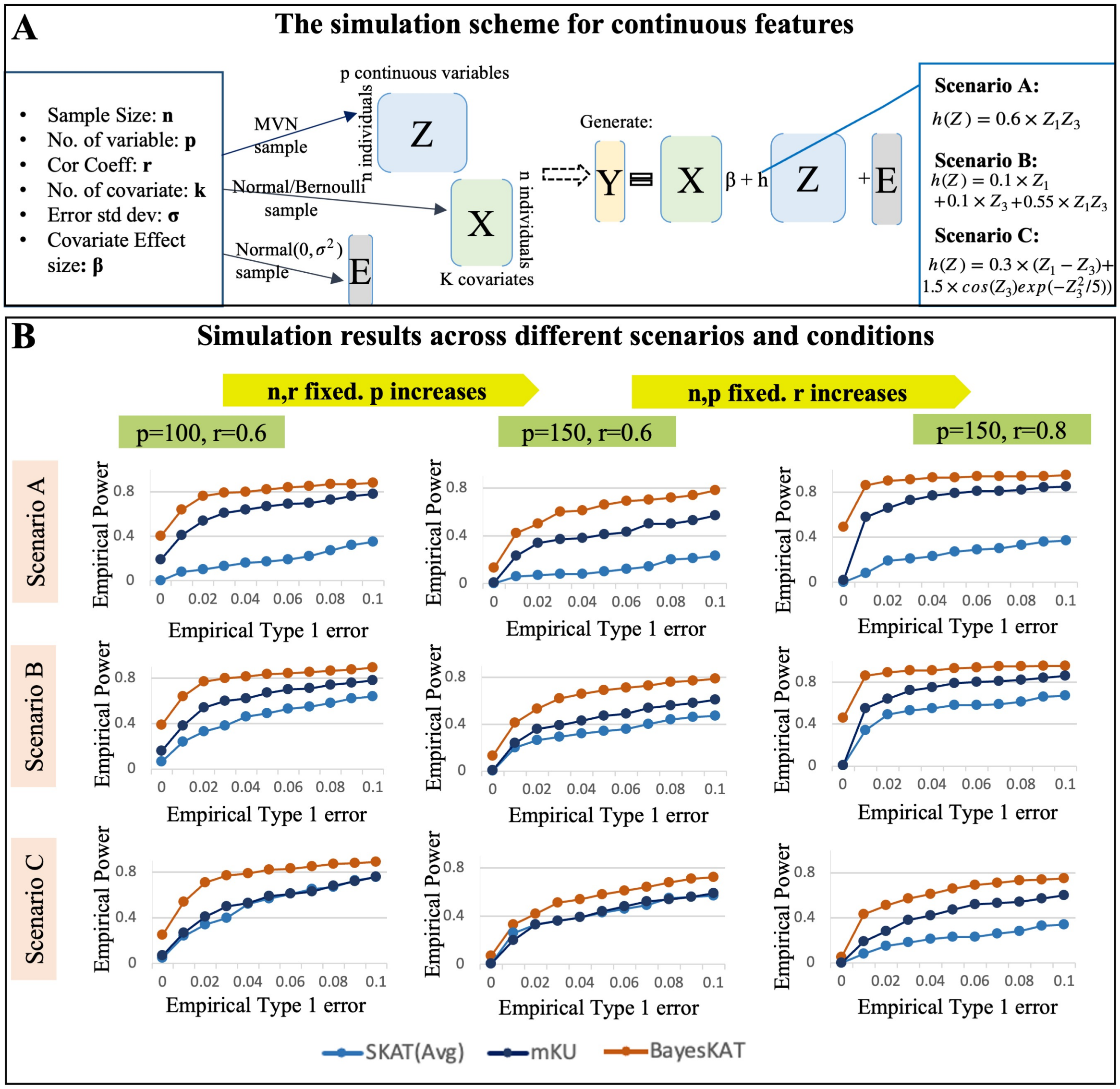


Figure 2.2 Performance comparison based on simulations using continuous features. (A) Schematic summary of the data generation process for continuous molecular features (e.g. gene expression features) and the demonstration of the implementation under various scenarios. (B) Performance comparison across different simulation settings with systematic performance evaluations, i.e. the empirical power versus empirical type 1 error, for SKAT(Avg), mKU, and BayesKAT across different scenarios and parameter settings. When increasing *p* while keeping other factors constant, all methods exhibit a slight decline in power, but BayesKAT consistently outperforms SKAT(Avg) and mKU. Additionally, as *r* increases under fixed parameters, BayesKAT also consistently surpasses SKAT(Avg) and mKU.

from N(2,1). The covariate coefficients are set as $\beta = (0.03, 0.5)$ and the outcomes Y are simulated based on model (2.1). The error term with variance $\sigma^2 = 1$ is added as the random noise. Three commonly used candidate kernels are incorporated in BayesKAT: Linear, Quadratic, and Gaussian. Different functional forms of $h(\cdot)$ are used to create different scenarios to test the performance. The empirical type 1 errors of each method are calculated based on the specific scenario where $h(Z) = 0$, i.e. there is no association between Z and Y in the simulated data. The different scenarios are given below:

- Scenario A: $h(Z) = 0.6 \times Z_1 Z_3$
- Scenario B: $h(Z) = 0.55 \times Z_1 Z_3 + 0.1 \times Z_1 + 0.1 \times Z_3$
- Scenario C: $h(Z) = 0.3 \times (Z_1 - Z_3) + 1.5 \times cos(Z_3) exp(-Z_3^2/5))$

In all the simulation scenarios, as shown in Figure 2.2B, the empirical power versus empirical type 1 error for BayesKAT is consistently better than that of SKAT(Avg) and mkU. Moreover, sensitivity analyses are conducted to evaluate the effects with different simulation parameters on the final performance of different algorithms. Notably, when the number of features p increases from 100 to 150, while keeping all other parameters fixed, BayesKAT consistently achieves superior empirical power compared to other methods (see Figure 2.2B). Similarly, when the correlation r increases from 0.6 to 0.8 with p fixed as 150, BayesKAT consistently demonstrates higher empirical power and outperforms other methods. Taken together, these simulation results demonstrate the robust superior performance of BayesKAT compared to SKAT(Avg) and mKU in various settings with continuous features.

2.3.2 Simulation with discrete features:

The performance of different models on discrete SNP features is first evaluated based on simulations where randomly selected SNP groups are used as features. Because randomly selected SNPs are generally not functionally related, the overall effectiveness of all KBT models decreases as expected, with BayesKAT still showing improved empirical power compared to other methods (Supplementary Figure E.2). Because the real-world implementations of KBT models for genetics

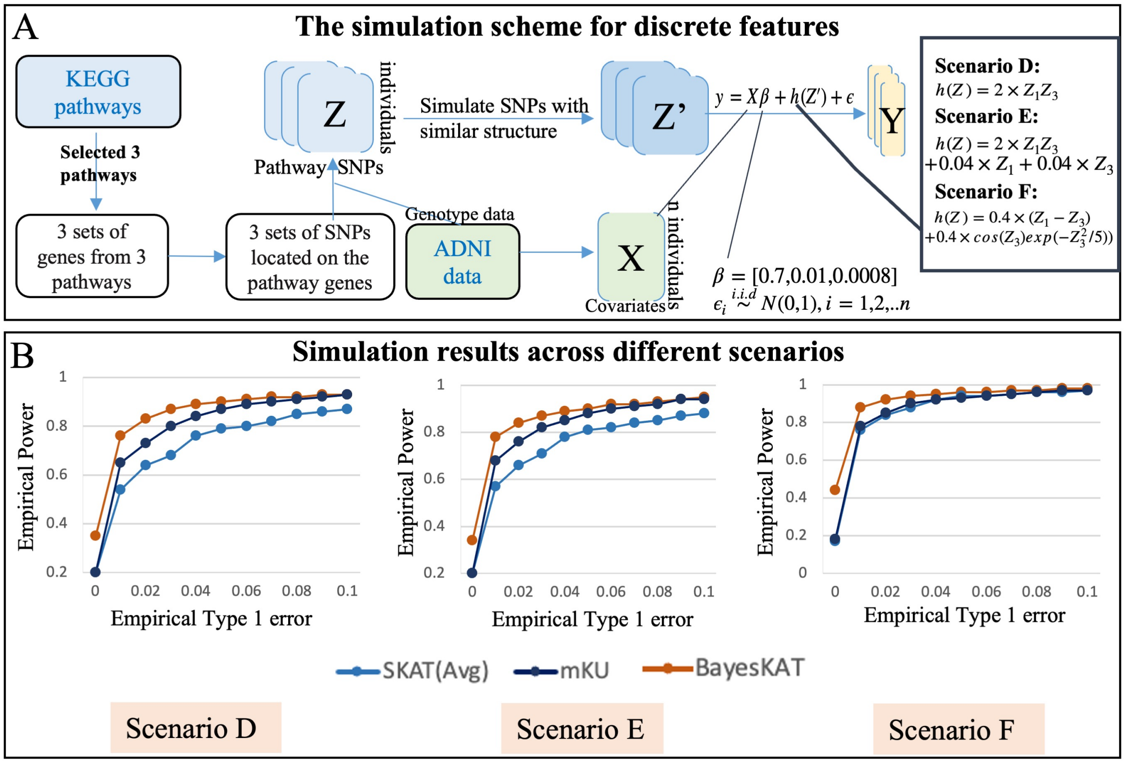


Figure 2.3 Performance comparison based on simulations with discrete features. (A) Schematic summary of the data generation process for discrete genetic features (e.g., SNP variants) and the demonstration of the implementation under various scenarios. Starting with the original Z matrix containing groups of SNPs from each pathway, a simulated SNP matrix Z' is generated, preserving the underlying interrelationships among the SNPs. Covariate variables (i.e., age, gender and occupation) are incorporated based on the data from ADNI. (B) Performance comparison across different simulation settings. Systematic performance evaluations, i.e. the curves of empirical power versus empirical type 1 error, for SKAT(Avg), mKU, and BayesKAT across different scenarios are plotted. The performance across three sets of simulated datasets is averaged. The performance curves based on each individual simulated dataset can be found in Supplementary Figure E.3.

studies usually focus on functionally related groups of SNPs, a more realistic strategy of simulating groups of discrete SNP features, instead of randomly selected unrelated SNPs, is employed to further benchmark the performance of BayesKAT (Figure 2.3A).

To rigorously capture the underlying linkage disequilibrium structures of discrete features in real-world SNP data, the ADNI dataset is used as the basis for the simulations, where the covariate matrix X is created from the real covariates (e.g. age, gender and education) of the corresponding individuals in the dataset, and SNPs located in specific genes belonging to the selected KEGG pathways are included into the testing (see Materials and Methods). Three different KEGG pathways are randomly selected for performance evaluations. For each pathway, the groups of SNPs located in the corresponding gene members are identified. The "SNPknock" package [42] is then used to simulate the knockoff SNP data, which maintains the structural dependency among SNPs in each group intact in the simulated knockoff SNPs (Figure 2.3A). The feature matrix Z is constructed based on the simulated SNP data, with $n = 755$ and p ranging between 4000 and 5000. The outcome variable Y is simulated based on three different scenarios. Each scenario corresponds to a different functional form $h(Z)$, that is:

- Scenario D: $h(Z) = 2 \times Z_1 Z_3$
- Scenario E: $h(Z) = 2 \times Z_1 Z_3 + 0.04 \times Z_i + 0.04 \times Z_3$
- Scenario F: $h(Z) = 0.4 \times (Z_1 - Z_3) + 0.4 \times cos(Z_3) exp(-Z_3^2/5))$

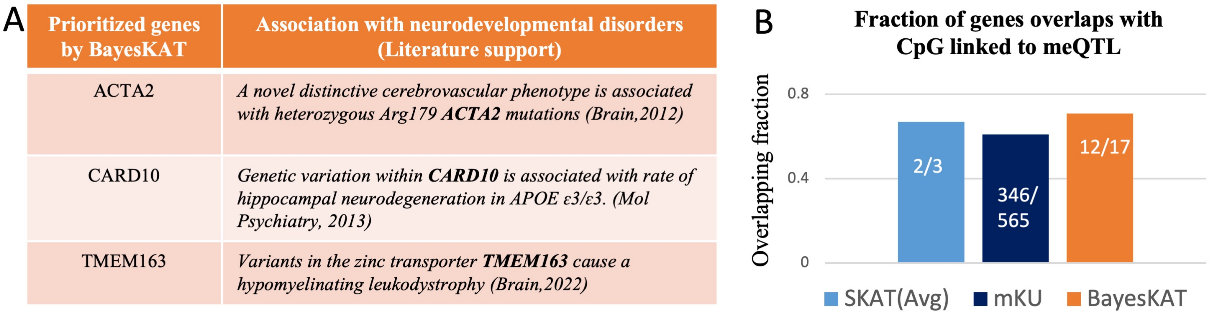
Where Z_i is the ith column of Z corresponding to the ith SNP. For each group of pathway-level knockoff SNPs, 500 simulations are generated. By applying BayesKAT and the other methods on the set of simulations, the corresponding empirical type 1 error and empirical power are calculated accordingly. The summary of performance comparisons based on this extensive set of simulations is shown in Supplementary Figure E.3. The empirical power and empirical type 1 error for each SNP group clearly demonstrates that BayesKAT robustly outperforms SKAT(Avg) and mKU, across different simulation scenarios and settings. The averaged performance over 3 pathway-level SNP groups can be found in Figure 2.3B. To demonstrate the robust superior performance of BayesKAT, the simulation study is repeated using different sample sizes, n=1000,1500. The comparison outcomes are illustrated in supplementary Figures E.4 and E.5. Remarkably, in addition to these consistent advantages, BayesKAT also achieves much lower empirical type 1 error across all simulation settings, when the suggested Bayes Factor threshold [26] is employed. It suggests that the associations between the SNP groups and the phenotype detected by BayesKAT exhibit a significantly higher level of reliability compared to other methods, a crucial attribute for genetic applications in complex diseases.

2.4 Application to ADNI datsets: BayesKAT reveals novel associated genetic basis of complex traits

To illustrate the novel biological insights generated by BayesKAT, the individual-level data, including the genotype, phenotype and demographic covariates, from the ADNI project (https://adni.loni.usc.edu) [43], [22] are used to conduct a series of group-level genetic association testings. Specifically, BayesKAT is used to test the group-wise associations between SNP sets and the complex phenotype of whole brain volume, based on the available information across 755 individuals in the ADNI cohort.

2.4.1 Real data preprocessing

In addition to a series of simulated datasets, real genetic datasets are used to evaluate the performance of BayesKAT and its derived biological discoveries. The data used in the preparation of this article were obtained from the Alzheimer's Disease Neuroimaging Initiative (ADNI) database (adni.loni.usc.edu). The ADNI was launched in 2003 as a public-private partnership led by Principal Investigator Michael W. Weiner, MD. The primary goal of ADNI has been to test whether serial magnetic resonance imaging (MRI), positron emission tomography (PET), other biological markers and clinical and neuropsychological assessments can be combined to measure the progression of mild cognitive impairment (MCI) and early Alzheimer's disease (AD). Plink software [41] (http://pngu.mgh.harvard.edu/purcell/plink/) is used to pre-process the individual-level genotype data. Detailed information on data access, download and pre-processing steps can be found in the supplementary material. Four biology-based strategies are used to create the feature groups of functionally related SNPs, leading to complementary biological insights into the genetic basis underlying the specific phenotype. The four SNP feature grouping strategies are: (1) Gene-wise SNP groups for 18,999 protein-coding genes in the human genome[44]. For each protein-coding gene, all SNPs located within +/-5KB of the gene body are collected as the gene-wise SNP feature set; (2) Pathway-wise SNP groups for 352 KEGG pathways [23], [24], [25]. For each pathway, gene-wise SNP groups for all genes belonging to the specific pathway are collected as the pathway-wise SNP feature set. The number of SNPs per pathway varies from 35 to 22,555, with a mean of 1,721 SNPs. Figure 2.5A provides a schematic figure demonstrating the pathway-wise joint SNP testing procedure; (3) Co-expression gene module based SNP groups. Forty-one co-expression gene modules are identified using the R package "WGCNA" [45], [46] to find correlated gene co-expression clusters from expression data (available on the ADNI website). The co-expression gene modules have a different number of genes in them, which varies from 6 to 3361. SNPs within each module are extracted for further group-wise testing; and (4) 401 protein-protein interaction (PPI) gene module based SNP groups. The PPI gene modules are previously created in [47] based on the topology of the PPI network. The number of genes in PPI modules ranges from 2 to 497.



Prioritized genes by BayesKAT	Association with neurodevelopmental disorders (Literature support)
ACTA2	*A novel distinctive cerebrovascular phenotype is associated with heterozygous Arg179* ***ACTA2*** *mutations (Brain,2012)*
CARD10	*Genetic variation within* ***CARD10*** *is associated with rate of hippocampal neurodegeneration in APOE ε3/ε3. (Mol Psychiatry, 2013)*
TMEM163	*Variants in the zinc transporter* ***TMEM163*** *cause a hypomyelinating leukodystrophy (Brain,2022)*

Figure 2.4 Functional validation of BayesKAT's prioritized genes using orthogonal information. (A) Top-ranking genes prioritized by BayesKAT are strongly supported by previous literature of functional studies of brain-related diseases. (B) The selected genes by BayesKAT demonstrate higher fractions of overlapping meQTL's CpG sites than the genes selected by SKAT(Avg) and mKU. The CpG sites of significant meQTLs from the brain tissues represent orthogonal molecular-level evidence in support of the gene's functional involvement with whole brain volume.

2.4.2 BayesKAT prioritizes functionally related genes:

To identify genes associated with the trait of whole brain volume, we conduct a gene-wise association test with gene-level SNPs (see Materials and Methods). BayesKAT prioritized 17 genes, whose posterior probability of association ($P(H_1|Data)$) is greater than 0.7. Figure 2.4A shows some examples of the prioritized genes, which have been suggested to be associated with brain or neurodegenerative disorders by previous seminal studies [48], [49], [50]. The whole list of the 17 prioritized genes and their corresponding posterior probability of associations are provided in Supplementary Table E.1. Strikingly, as external evidence in support of BayesKAT's prioritized genes, 12 out of the 17 genes (71%) contain CpGs that have been found to be involved with significant meQTLs [51] in the human brain cortex. In contrast, the genes prioritized by SKAT(Avg) and mKU demonstrate much lower fractions of overlapping with CpGs linked to meQTLs (66% and 61% respectively, Figure 2.4B). The higher proportion of prioritized genes containing CpGs offers molecular-level support for BayesKAT's ability to uncover genes mechanistically linked to complex traits.

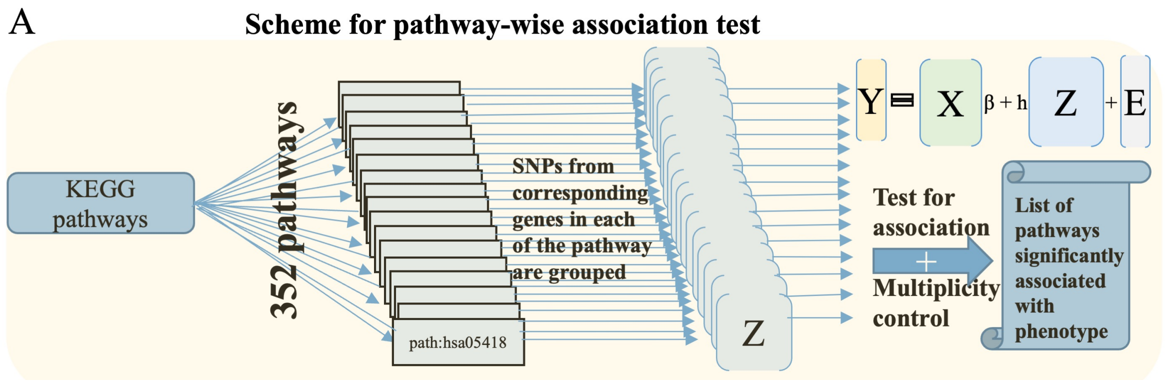
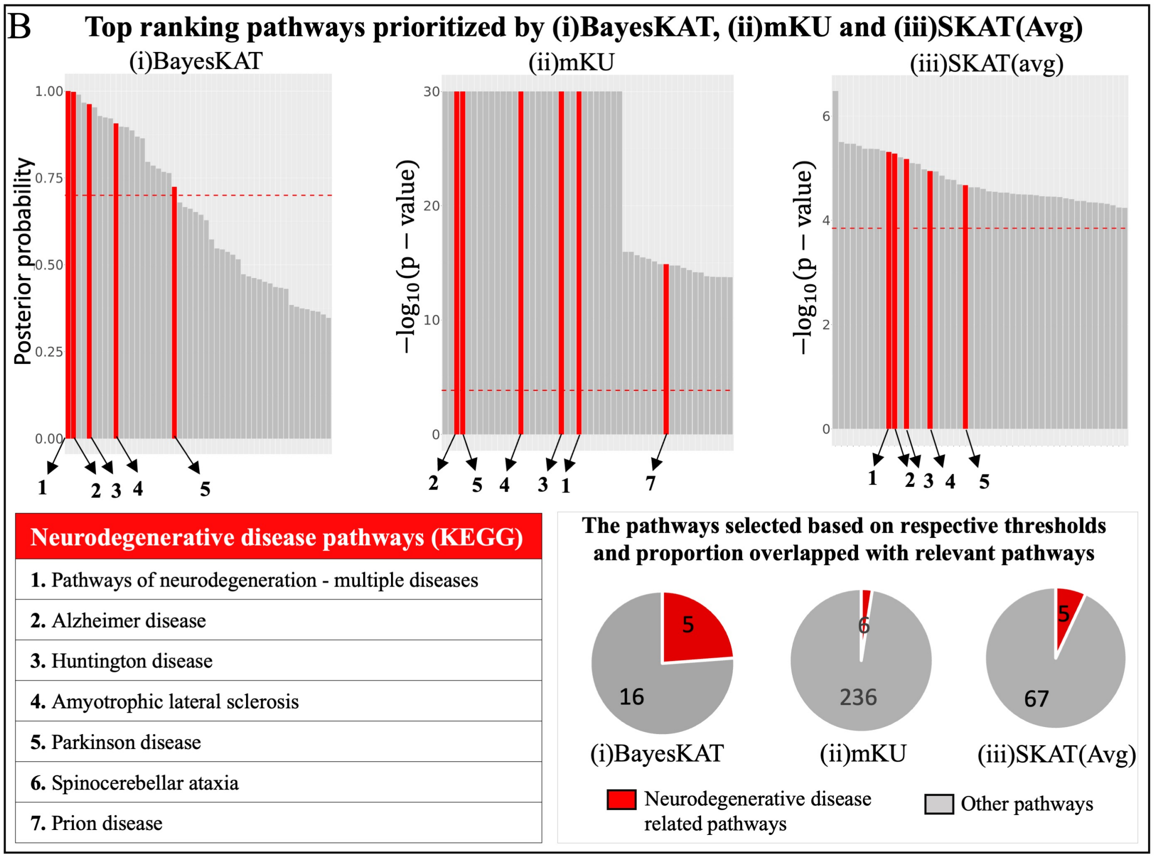

A
Scheme for pathway-wise association test
KEGG pathways
352 pathways
SNPs from corresponding genes in each of the pathway are grouped
path:hsa05418
Z
Y = X β + h Z + E
Test for association
+
Multiplicity control
List of pathways significantly associated with phenotype

Figure 2.5 Pathway-level association tests by BayesKAT prioritizes neurodegenerative disease related pathways. (A) Schematic representation illustrating the steps of pathway-level association tests. Sets of SNPs located within genes belonging to each of the 352 pathways are tested simultaneously for pathway-level associations with the phenotype of interest (e.g., the whole brain volume). Multiplicity control is implemented to identify the specific list of significant pathways linked to the phenotype.(B) The top-ranking pathways prioritized by (i) BayesKAT, (ii) mKU, and (iii) SKAT(Avg) demonstrate distinct enrichment with neurodegenerative disease associated pathways. Top 50 pathways are shown for fair comparison. The top-ranking pathways by BayesKAT are ranked by the estimated posterior probabilities of the pathway-level associations. The top-ranking pathways by the frequentist methods, mKU and SKAT(Avg) are ranked by the $-\log_{10}$(p-values). The pathways highlighted in red are neurodegenerative disease related pathways. The red horizontal dashed line in each bar plot indicates the threshold used by each model for fair comparison (see Materials and Methods). The pie charts illustrate the proportion of the selected pathways (above model's thresholds) that belongs to the neurodegenerative disease pathways. BayesKAT notably exhibits enhanced prioritization of neurodegenerative disease pathways. Due to the issue of inflated p-values in mKU, pathways with p-values of 0 are assigned $-\log_{10}$(p-values)=30 for visualizations.

Figure 2.5 (cont'd)



Neurodegenerative disease pathways (KEGG)
1. Pathways of neurodegeneration - multiple diseases
2. Alzheimer disease
3. Huntington disease
4. Amyotrophic lateral sclerosis
5. Parkinson disease
6. Spinocerebellar ataxia
7. Prion disease

2.4.3 Biological pathways linked to neurodegenerative diseases are top-ranked by BayesKAT

To identify biological pathways that potentially modulate the whole brain volume trait, BayesKAT is used to analyze pathway-level SNP groups (see Materials and Methods, Figure 2.5A). The top 50 ranked KEGG pathways by each model are summarized in Figure 2.5B, where BayesKAT ranks the pathways based on decreasing posterior probability of association ($P(H_1|Data)$) while the mKU and SKAT(Avg) methods rank the pathways based on decreasing -log$_{10}$(p-value). Interestingly, BayesKAT successfully prioritized most of the neurodegenerative disease related pathways with top ranks (Figure 2.5B). This is a strong mechanistic support to BayesKAT's results, because the neurodegenerative diseases, including Alzheimer's disease, Huntington's disease, Amyotrophic lateral sclerosis and Parkinson's disease, have been found to be strongly related to brain volume loss [52], [53], [54], [55]. Based on a reasonable posterior probability threshold 0.7, there are 21 pathways identified by BayesKAT (Figure 2.5B), Supplementary Table E.2), among which five pathways are associated with neurodegenerative diseases. In comparison, the mKU and SKAT(Avg) models prioritized large numbers of pathways (242 and 72 respectively), while only a small fraction of them are neurodegenerative disease related pathways (6 and 5, respectively). For SKAT(Avg), these functionally related pathways are not even the top-ranked ones. As summarized in the corresponding pie charts in Figure 2.5B, BayesKAT achieves the highest efficiency in prioritizing important pathways and resulting in fewer false discoveries than the other methods. These results are also consistent with the larger type 1 errors of mKU and SKAT(Avg) observed from simulation analyses described above. Note that, setting the posterior probability threshold is subjective, same as selecting type 1 error threshold or FDR threshold (0.05 or 0.01 or 0.1). Opting for a threshold greater than 0.7 (such as 0.9, 0.95, or 0.99) is also effective, as BayesKAT efficiently prioritizes the top pathways. Overall, the highly prioritized functional relevant pathways imply the novel biological insights that can be generated by using BayesKAT.

To further evaluate the performance of BayesKAT in determining the optimal kernel weights, 10 randomly chosen pathways are used to test the pathway-level associations. The resulting composite kernels are compared to the results of using SKAT based on individual kernels separately. The corresponding -$\log_{10}$(p-values) metrics from SKAT using individual kernels are compared to the inferred kernel weights in the composite kernels from BayesKAT. As shown in Figure 2.6A, the high similarity between the two heatmaps indicates that BayesKAT can efficiently select the optimal composite kernel automatically from the data, without relying on prior knowledge or repetitively trying different individual kernels.

2.4.4 BayesKAT identifies trait-associated gene modules and protein complexes

To further demonstrate BayesKAT's capability of revealing novel group-level associations to traits from specific sets of cooperative SNPs, two additional SNP grouping strategies are applied: 1) SNPs from co-expression gene modules; and 2) SNPs from protein-protein interaction (PPI) modules (see Materials and Methods). Applied on the SNP groups aggregated from co-expression gene modules, BayesKAT is able to pinpoint specific modules as significantly associated with the whole brain volume trait (Figure 2.6B Left). On the other hand, SKAT(Avg) and mKU identify a large number of modules (Figure 2.6B Right), which are consistent to the inflated type 1 errors of these two methods as observed previously, and failed to provide specific prioritizations of the gene modules. Remarkably, by comparing to the significant GWAS SNPs identified from another genome-wide meta-analysis of brain volume study [56], two out of the four selected modules by BayesKAT (50%) contain significant GWAS SNPs (Figure 2.6C). In contrast, only 26% (7/27) and 43% (6/14) modules selected by mKU and SKAT(Avg) contain significant GWAS SNPs. These results further provide orthogonal support for the superior performance of BayesKAT in identifying new collective associations to the complex traits for SNP groups of functionally related genes.

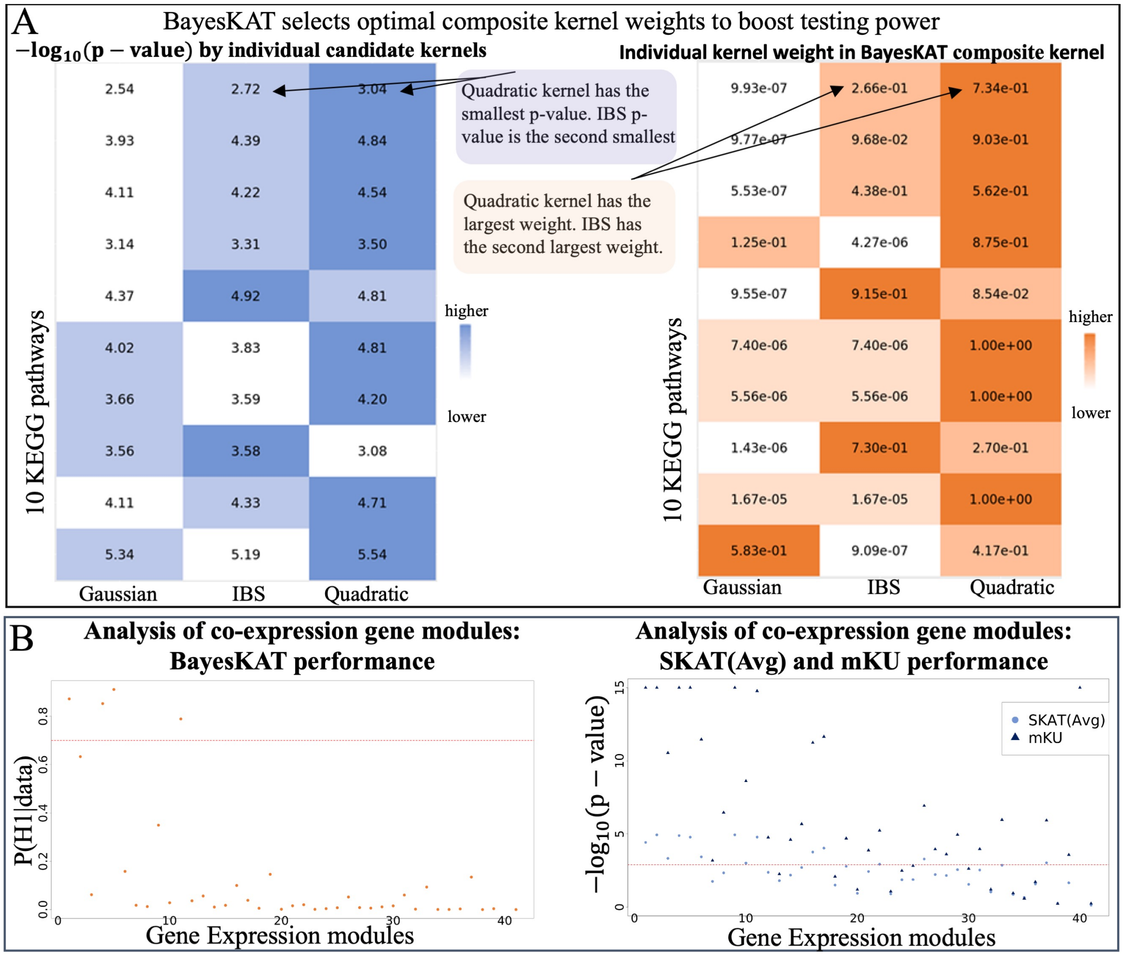
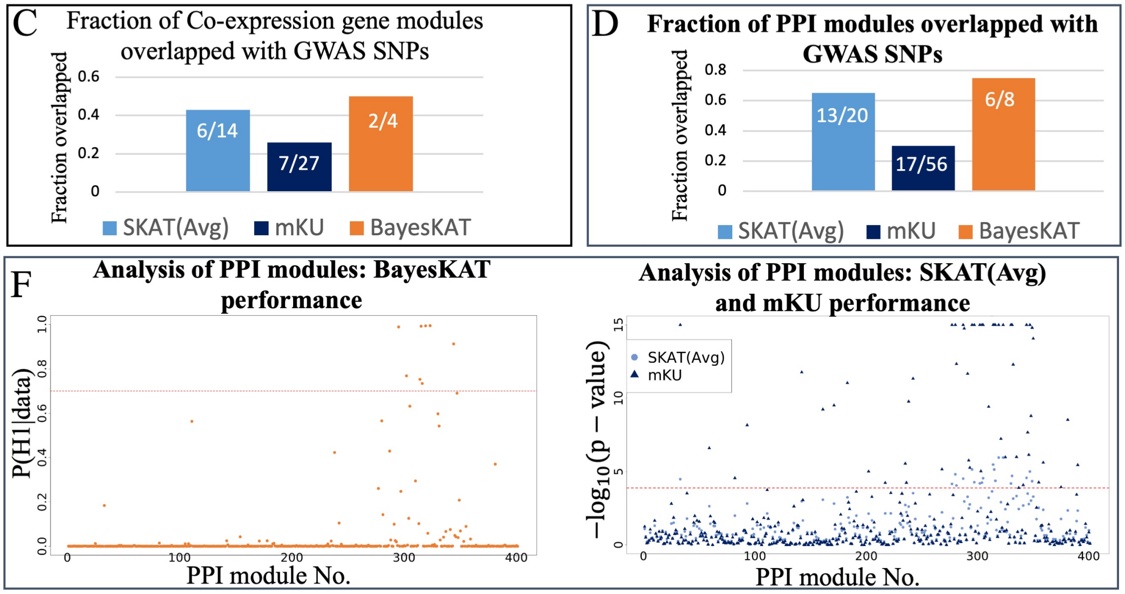


Figure 2.6 Boosted association tests based on BayesKAT's composite kernels reveal novel modules of genes and proteins linked to brain volume. (A) The inferred weights of individual kernels in BayesKAT's composite kernels (right) recapitulate the strength of each kernel (-$\log_{10}$(p-values)) when each kernel is incorporated separately (left). Without relying on prior knowledge or repetitively testing different kernels separately, BayesKAT automatically infers the appropriate composite kernels to boost the group-level tests for different pathways. (B) Prioritized co-expression gene modules by BayesKAT (left) vs. SKAT(Avg) and mKU (right). The significance threshold of selection for each model is represented by the horizontal red dashed lines (see Materials and Methods). (C) The selected significant co-expression gene modules by BayesKAT demonstrate higher fractions of overlapping with significant SNPs from orthogonal GWAS studies, compared to the results from SKAT(Avg) and mKU. (D) The selected significant PPI modules by BayesKAT demonstrate higher fractions of overlapping with significant SNPs from orthogonal GWAS studies, compared to the results from SKAT(Avg) and mKU. (E) Prioritized PPI modules by BayesKAT (left) vs. SKAT(Avg) and mKU (right). The significance threshold of selection for each model is represented by the horizontal red dashed lines (see Materials and Methods)

Figure 2.6 (cont'd)



When applied to SNP groups organized according to PPI modules that largely represent protein complexes, BayesKAT distinctly prioritizes eight PPI modules as significant, six of which contain significant GWAS SNPs (75%, Figure 2.6D, Figure 2.6E Left). On the contrary, only 30%(17/56) and 65%(13/20) of the modules selected by mKU and SKAT(Avg) respectively contain significant GWAS SNPs (Figure 2.6D, Figure 2.6E Right). Taken together, the highly specific prioritizations of potential gene modules and protein complexes, along with the substantially improved justification from other GWAS SNPs, suggest that BayesKAT can facilitate novel discoveries of molecular components involved in complex traits and may pave the way for innovative approaches to disease treatments.

2.5 Discussion

BayesKAT (https://github.com/wangjr03/BayesKAT) is a data adaptive methodology that automatically selects the appropriate composite kernel using the MCMC algorithm (BayesKAT-MCMC) or the optimization technique (BayesKAT-MAP) and conducts hypothesis testing on the presence of group-level genetic associations for complex traits. The Bayesian framework and the inferred posterior probabilities are more interpretable and informative, compared to p-values from frequentist methods. Based on extensive benchmark analyses, BayesKAT demonstrates consistent superior performance than other methods across different settings. Moreover, evaluated on a series of biologically inspired SNP groups based on a real genetic dataset, BayesKAT not only achieves improved prioritization of functionally relevant and justified group-level SNP associations, but also enables novel discoveries with respect to the underlying molecular mechanisms of complex traits. By revealing the collective effects of functionally cooperative SNPs without relying on the prior knowledge of specific kernels, BayesKAT represents one important step forward towards the goal of deciphering the intricate genetic basis of human diseases.

Although some methods based on the Gaussian process [57] or supervised learning technique [58] attempt to select the best kernel using training data for prediction purposes, BayesKAT is the first Bayesian KBT methodology that simultaneously selects the optimal composite kernel while testing for the associations, without requiring the training data. In addition, the data-adaptive strategy of composite kernel selections also facilitates the description of more complicated interdependence structures that can not be fully captured by individual kernels. Furthermore, BayesKAT provides the flexibility of incorporating multiple testing corrections, integrating prior biological knowledge, and modeling various data types. To complement the MCMC strategy, BayesKAT-MAP is highly scalable and can be efficiently implemented for large-scale genome-wide studies.

BayesKAT utilizes the Metropolis-Hastings MCMC algorithm in combination with a derivative-free grid-search-based optimization approach to choose the composite kernel for specific datasets. Nevertheless, the BayesKAT framework is not restricted to these techniques. Other efficient MCMC sampling algorithms or reliable optimization techniques can be incorporated. A variety of sampling techniques have been reviewed and compared for different purposes of modeling [59], [60], [61], [62]. Integrating these techniques, especially the variational Bayes techniques, into the BayesKAT framework and systematically evaluating their performance for different types of applications will be an important step for future developments (More discussion on this topic is included in the supplementary files).

2.6 Data and code avilability

BayesKAT is an open source infrastructure available in the GitHub repository https:// github.com/ wangjr03/BayesKAT. The repository includes R codes for BayesKAT, along with comprehensive instructions, sample testing data, and code for pre-processing real data. Please note that the Alzheimer's Disease Neuroimaging Initiative (ADNI) data used for this manuscript is not publicly available, and interested users can request for access through the official portal: https://adni.loni.usc.edu/data-samples/access-data/. Detailed information on data download and processing procedures can be found in the supplementary material.

2.7 Supplementary materials

2.7.1 Kernel functions

A kernel function is defined as a function $K : \mathcal{X}\times\mathcal{X} \to \mathbb{R}$, where the kernel matrix $K = (k_{i,i'})_{i,i'=1}^{n}$ is symmetric and positive semidefinite with $k_{i,i'} = k(Z_i, Z_{i'})$. In this setting, $k(Z_i, Z_{i'})$ is a measure of similarity between the ith and the i'th subject. There are a variety of kernel functions to choose from, and the most widely used ones include the Linear kernel, the Quadratic kernel and the Gaussian kernel. For genetic SNP data, identity by state (IBS) kernel is a popular candidate kernel function suggested by various studies [19], [10], [12]. The functional forms of these kernels are summarized below:

- Linear kernel: $K(Z_i, Z_{i'}) = Z_i^T Z_{i'}$
- Quadratic kernel: $K(Z_i, Z_{i'}) = (Z_i^T Z_{i'} + 1)^2$
- Gaussian kernel: $K(Z_i, Z_{i'}) = exp\{-\|Z_i - Z_{i'}\|^2/l\}$, where $\|Z_i - Z_{i'}\|^2 = \sum_{j=1}^{p}(Z_{ij} - Z_{i'j})^2$, l is a tuning parameter.
- IBS kernel: $K(Z_i, Z_{i'}) = (2p)^{-1} \sum_{j=1}^{p} IBS(Z_{ij}, Z_{i'j}) = (2p)^{-1} \sum_{j=1}^{p}(2 - |Z_{ij} - Z_{i'j}|)$

2.7.2 Importance of choosing appropriate Kernel

Although a variety of different kernels are available, for a given dataset, it is practically impossible to know a priori which kernel will fit the dataset best and maximize the testing power.

Genetic data related to complex phenotypes pose particular challenges, primarily stemming from our limited understanding of how the interplay among genetic or molecular features influences their collective association with a phenotype. Therefore, choosing a kernel randomly can lead to a less powerful testing procedure for genome-wide applications. For example, if the outcome variable Y is related to the features through a Quadratic function, using a Linear kernel in the model will lead to weak tests that are not able to reject the null hypothesis even when the association is strong. On the other hand, by repeatedly applying KBT models based on different candidate kernels and choosing the one resulting in the minimum p-value, there is a high chance of making a false discovery, i.e., rejecting the null hypothesis when there is no association.

Combining a panel of candidate kernels together to create a composite kernel is thus a natural and effective strategy to overcome this issue. While a straightforward strategy of averaging kernels to form a composite kernel, i.e., a linear combination of candidate kernels with equal weights, can perform better than the worst-performing kernel function, it usually cannot perform as efficiently as the best kernel to accurately represent the association between the trait and features for a given dataset, thus, is not guaranteed to increase the statistical power. As shown in main Figure 1 (A)(B), evaluated on both real and synthetic datasets, the average kernel strategy can often lead to incorrect and inconsistent results in practice. Thus, a systematic data-adaptive approach of optimal kernel selection is highly desirable for high-dimensional genome-wide association tests, especially for complex human disease phenotypes that are genetically modulated by multiple inter-dependent genetic variants.

2.7.3 Comaparison between BayesKAT-MCMC and BayesKAT-MAP

The performance and runtime of BayesKAT-MCMC and BayesKAT-MAP under different settings are systematically compared. Supplementary Table 2.1 summarizes the empirical type-1 error and empirical power for different simulated functional dependencies. The simulations were based on the proposed settings used by previous studies [21]. As shown in Table 1, both MCMC and MAP strategies achieve nearly equal statistical power in detecting associations. While BayesKAT-MCMC provides more information on the posterior distributions of parameters, it is more computationally expensive compared to BayesKAT-MAP. When the number of samples or features, i.e. n or p, increases beyond 500, BayesKAT-MCMC is not sufficiently scalable without requesting more high-performance computing resources. On the other hand, Main Figure 1(D) shows the superior computational scalability of BayesKAT-MAP based on the same level of computational resource support. Therefore, these two alternative strategies provide similar accuracy and complementary signatures for genetic association tests of complex traits, with BayesKAT-MAP being more flexible and conservative. For the rest of the paper, results of BayesKAT-MAP are presented due to its scalability. The code for implementing BayesKAT-MCMC and BayesKAT-MAP are both made publicly available via GitHub: https://github.com/wangjr03/BayesKAT.

Table 2.1 Empirical power of BayesKAT-MCMC and BayesKAT-MAP.

h(Z)	n	p	BayesKAT-MCMC	BayesKAT-MAP
$h(Z) = 0$	500	500	0.018	0
$h(Z) = 2 \times Z_1 Z_3$	500	500	0.946	0.946
$h(Z) = 2 \times Z_1 Z_3 +$ $0.04 \times Z_i + 0.04 \times Z_3$	500	500	0.948	0.958
$h(Z) = 0.4 \times (Z_1 - Z_3) +$ $0.4 \times cos(Z_3) exp(-Z_3^2/5)$	500	500	0.976	0.998

The functional forms in simulation scenarios are taken from [21] and coefficients are adjusted based on n (sample size) and p (no. of features)

2.7.4 Multiple testing correction

When m_1 multiple groups are simultaneously tested, multiplicity corrections are needed. Multiple testing corrections on p-values from frequentist models are carried out using methods such as the Bonferroni correction [63], which controls the family-wise type 1 error to α by setting the individual test's type 1 error at α/m_1. Other methods like FDR control have also been popular ones. For multiple testing corrections on Bayesian models, the multiplicity control is achieved by setting a high prior probability of the individual null hypothesis, as suggested by previous studies [64] [65]. For each test, the prior probability is set as $P(H_0) = 0.99$, which is equivalent to assuming that, on average, one in 100 tests is believed to have a true association. Considering the number of SNP groups in genetic applications, such as the number of biological pathways or co-expression modules, this choice of prior probability setting is regarded as rather conservative and ensures fair performance comparisons.

2.7.5 ADNI Data Handling and Pre-processing steps

The Alzheimer's Disease Neuroimaging Initiative(ADNI) data referenced in this manuscript is not publicly accessible. Interested users can request access through the official portal at https://adni .loni.usc.edu/data-samples/access-data/. Once granted access, users can navigate to the "downloads" section and download the genotype data. For this study, "ADNI 1 SNP genotype data - PLINK" file containing the .bed, .fam and .bim files was downloaded. Initially, these files contain information on 757 individuals and 620901 SNPs. The standard quality control steps are performed using Plink[41] software, which resulted in the removal of SNPs and individuals failing standard missingness thresholds, Minor Allele Frequency (MAF), and Hardy-Weinberg Equilibrium (HWE) criteria. Subsequent to quality control, a simple imputation technique addresses the remaining missing values, resulting in a final genotype matrix featuring 531086 SNPs and 756 individuals meeting the established quality criteria. Demographic and phenotypic information for these individuals is retrieved using the publicly available ADNIMERGE package (detailed description: https://adni.bitbucket.io/). Covariates, specifically Age, gender, and education levels, which exhibit significant linear relationships with the phenotype of interest (whole brain volume), are integrated into the model. By aligning individual IDs across response, genotype, and covariate data, a final dataset comprising 755 individual-level data points is obtained. The GitHub repository https://github.com/wangjr03/BayesKAT contains the specific codes for preprocessing real individual-level genotype datasets.

2.7.6 Further Discussion

In recent years, different deep learning models have been developed and applied in genomics studies to predict molecular features, such as gene expression, histone marks, chromatin accessibility and transcription factor binding, using DNA sequences as features [66], [67], [68], [69]. These models allow for in silico mutations of DNA sequences and the prediction of perturbed molecular features for each mutation individually. Compared to these models, the power of BayesKAT in delineating the collective group-level genetic associations based on automatic composite kernel selection opens up a new level of analytical capability of dissecting the genetic complexity. Combined together, the complementary advantages of BayesKAT and deep learning models are expected to facilitate novel mechanistic insights into human diseases.

The implementation of BayesKAT is not limited to genetic studies based on features of SNPs or gene expressions. Any group of continuous or discrete features that are functionally related can be tested for associations with an outcome using the BayesKAT methodology. For instance, BayeskAT can be used to test if a group of images is associated with a particular disease trait by adopting properly defined kernels. As another direction for future developments, non-linear functions of candidate kernels, instead of the linear combinations, will be explored as the composite kernel, which may lead to improved power for kernel-based testing.

CHAPTER 3

SPATIAL TRANSCRIPTOMICS - SVG DETECTION TECHNIQUES AND SCOPES FOR IMPROVEMENT

A modified version of part of this chapter was previously published (Das Adhikari,Sikta et al, 2024): Das Adhikari S., Yang J., Wang J. and Cui, Y. (2024) Recent advances in spatially variable gene detection in spatial transcriptomic. https://doi.org/10.1016/j.csbj.2024.01.016

3.1 Introduction: Analysis of spatial transcriptomic data

Recent advancements in Spatially-resolved transcriptomics (SRT) technology have provided comprehensive gene expression data for thousands of genes across multiple samples or spatial spots, accompanied by their respective spatial coordinates across a tissue which refers to a collection of cells that are organized in a specific manner and perform a particular function or set of functions within an organism. It is a complex and dynamic landscape where the spatial arrangement of cells is integral to understanding gene expression patterns and their implications for health and disease. Depending on the specific technology utilized, a sample could represent a single cell (as in the case of STARmap technology), a cell-sized local region (as with HDST technology[70]), or a localized region comprising dozens of cells (as seen in Slide-seq[71, 72] and Visium technologies). The latest SRT platforms, such as 10x Genomics Visium and Slide-seqV2, encompass thousands of spatial locations within each tissue sample, with future developments poised to achieve even higher resolutions. As technology progresses, the demand for more robust statistical frameworks to effectively analyze spatial data intensifies.

Although spatial transcriptomic (ST) data permit addressing a range of distinct questions, a fundamental initial step in the downstream analysis of spatial data is the identification of spatially variable genes (SVGs). These are genes that exhibit variations in expression levels either across the entire tissue or within predefined spatial domains. These genes can potentially unveil tissue heterogeneity and the underlying structural factors that drive distinct expression patterns across spatial locations, thus offering valuable insights into biology.

Numerous methods have been developed for the identification of SVGs. These methods encompass a spectrum of approaches, including the utilization of standard spatial statistics measures like Moran's I statistic[73] and Geary's C statistic[74] to rank genes based on their spatial autocorrelation. More advanced methods employ model-based approaches such as SpatialDE[75], SpatialDE2[76], SPARK and its extensions[77], nnSVG[78], BOOST-GP[79], marked point process frameworks like Trendsceek[80] and scGCO[81], or model-free frameworks like sepal[82] and GLISS[83]. Additionally, there are toolboxes, such as MERINGUE[84], Giotto[85], Seurat[86], Squidpy[87] that integrate some of these methods into comprehensive end-to-end analysis frameworks.

Downstream analysis involving SVGs encompasses various tasks, such as spatial clustering, deciphering spatial domains, and identifying spatial domain-specific SVGs. Additionally, there are numerous other downstream analyses that leverage additional information like scRNASeq data, histological images, and more, for tasks such as spatial decomposition of spots, gene imputation, the inference of cell-cell and gene-gene interactions and spatial location reconstruction for scRNA-seq data. However, this review till section 3.7 primarily concentrates on SVG detection frameworks and does not delve into the details of other downstream analyses in this section.

Therefore, the primary focus of this chapter is to discuss selected frameworks for SVG identification. This serves as a literature review aimed at providing a comprehensive overview of the field of spatial transcriptomic studies. The goal is to become acquainted with existing SVG identification frameworks, including their unique characteristics, novelty, as well as their pros and cons.

3.2 An overview of SVG detection techniques

Generally, in a spatial transcriptomics setup, the available spatial dataset contains gene expression measures/counts for m genes distributed across N known spatial coordinates or spots. This section establishes the key symbols that will be frequently utilized. Specifically, $y = (y_1, y_2, ..., y_N)$ is defined as the gene expression profiles/counts for a given gene across spatial coordinates (referred to as samples or spots), denoted by $s = (s_1, ..., s_N)$. The coordinates of the spatial locations are typically two-dimensional, i.e., $s_i = (s_{i1}, s_{i2})$, but any dimensional coordinates can be employed. The primary objective of these SVG detection models is to ascertain which genes, out of the m genes, are spatially variable across the tissue. In other words, the main goal is to determine whether the gene expression measure y depends on or relates to the spatial locations where the gene expression measures are collected.

Here, we classify SVG detection methods based on two primary categories: (1) based on input data type and (2) based on the computational framework. The initial categorization focuses on input data type, representing the foundational step in SVG detection. Therefore, we first discuss the input data pre-processing step in Section 3.2.1. Subsequently, Sections 3.2.2 and 3.2.3 delve into the detailed exploration of model-based and model-free approaches, respectively, aligning with the later categorization based on the computational framework. Table 1 is then presented in this sequential order to reflect the dual categorization process.

3.2.1 Gene expression data and pre-processing step

The gene expression measure y are generally of count data type (originated from sequence based or image based technology). Various SVG detection models have been developed to specifically use count data as input following some mandatory filtering and quality control steps. Some examples of these models include SPARK-X[88] ,BOOST-GP[79], SINFONIA[89], and GPcounts[90]. The gene expression count data often exhibit over-dispersion and contain numerous zero values, mainly due to the technology employed for data generation or simply because many genes are poorly expressed for biological reasons. These particular issues in count data are generally taken care of by using negative binomial models which handle over-dispersion well. For the issue of zero-inflation, Zhao et al, 2022 [91] showed that modeling zero inflation is not necessary in spatial transcriptomics, thus is not a concern in many method development. On the other hand, some methods, for example SpatialDE[75], nnSVG[78], and BOOST-MI[92], use normalized gene expression data as input in the framework for easy implementation, where in most of cases, the data is modeled using multivariate normal distribution after transformation. Authors in SPARK[77] proposed two different data models, SPARK and SPARK-G which uses count data and normalized data, respectively. The data normalization method is not unique for these methods. The normalization step generally removes the bias due to differences in sequencing depth using size factors and normalizes the data using log transformation(log10 or log2 transformations after adding a pseudocount value c, preferably 1). The method sepal[82] uses a slightly different normalization procedure which involves mapping the log-transformed values to the interval [0,1] and using a pseudocount 2. Other normalization methods, such as scran, scuttle, and scater R/Bioconductor packages[93, 94], can also be applied. Table 3.1 provides information on some selective methods together with their required input data type and the implemented model:

Table 3.1 A selective list of methods for SVG detection in ST data analysis categorized based on required input data type and the implemented computational framework.

Method	Input data type	Computational framework	Data model
SpatialDE2[76]	Count	model-based	Poisson
SPARK[77]	Count	model-based	Overdispersed poisson
BOOST-GP[79]	Count	model-based	Zero-inflated negative binomial
CTSV[95]	Count	model-based	Zero-inflated negative binomial
GPcounts[90]	Count	model-based	Negative binomial
SPARK-X[88]	Count	model-free	-
SINFONIA[89]	Count	Model-free	-
HEARTSVG[96]	Count	Model-free	-
SpatialDE[75]	Normalized	model-based	Multivariate Normal
SPARK-G[77]	Normalized	model-based	Multivariate Normal
nnSVG[78]	Normalized	model-based	Multivariate Normal
SOMDE[97]	Normalized	model-based	Multivariate Normal
BOOST-MI[92]	Normalized	model-based	Modified Ising model
Trendsceek[80]	Normalized	model-based	Marked point process
scGCO[81]	Normalized	model-based	Marked point process
sepal[82]	Normalized	Model-free	-
GLISS[83]	Normalized	Model-free	-
MULTILAYER[98]	Normalized	Model-free	-
BSP[99]	Normalized	Model-free	-

3.2.2 Overview of model-based frameworks

3.2.2.1 Gaussian process(GP) regression based and similar models

The majority of the methods, including some of the state-of-the-art algorithms to detect SVG, are based on Gaussian process (GP) regression models. For example, one of the first published SVG detection methods, SpatialDE[75], models the normalized gene expression y for a given gene using the following multivariate normal model:

$$p(y|\mu, \sigma_s^2, \delta, K) \sim N(y|\mu 1, \sigma_s^2 K + \delta I), \tag{3.1}$$

where the covariance term is decomposed into a spatial and a non-spatial part, where δI represents the non-spatial part and $\sigma_s^2 K$ is the spatial covariance matrix, whose $(i, j)^{th}$ element in the kernel matrix K denotes the spatial similarity between the i^{th} and j^{th} spot calculated based on the corresponding coordinates s_i and s_j. The choice of the kernel function plays a very important role in detecting the spatial correlation present in the gene expressions. More discussion about kernel function is provided in the next subsection.

Other methods like SPARK-G [77] (the Gaussian version of SPARK), nnSVG [78], and SOMDE [97] implement similar GP models for modeling normalized gene expression data with some extra features or added level of complexity. SPARK-G and nnSVG provide the option to include extra covariate terms in the model. The covariates or the explanatory variables could contain batch information, cell-cycle information, or other information that is important to adjust for during the analysis. SOMDE is a two-step procedure. This approach involves first utilizing a self-organizing map to cluster neighboring cells into nodes. Subsequently, it employs a Gaussian process to model and analyze the spatial gene expression patterns at the node level.

Table 3.1 shows that methods like SPARK[77], SpatialDE2[76], BOOST-GP[79], CTSV[95], and GPcounts[90] model count data directly. SPARK models the count data using an overdispersed poisson model where the logarithm of the unknown Poisson rate parameter is assumed to follow a stationary Gaussian process with similar spatial and non-spatial covariance components. BOOST-GP presents a novel Bayesian hierarchical model to analyze spatial transcriptomic data, which models the count data using a zero-inflated negative binomial(ZINB) model. The logarithm of the normalized expression level, which is included in the expectation term in NB, can be seen as a GP with a spatial covariance term representing the spatial variability in case there is a spatial pattern. GPcounts also uses negative binomial distribution to model the UMI(Unique Molecular Identifier) data. SpatialDE2 employs a Generalized Linear Mixed Model (GLMM) for count data modeling. In contrast to GP-based techniques that typically separate covariance into a spatial and a non-spatial component, SpatialDE2 dissects the covariance into several spatial components along with a non-spatial random component. CTSV implements a slightly different technique and does not use the GP model. In CTSV, the gene specific, spot specific and cell-type specific relative mean expression level in the ZINB model is a linear combination of $h_1(s_{.1})$ and $h_2(s_{.2})$ where the functions $h_1(\cdot)$ and $h_2(\cdot)$ represents the underlying true spatial effect modeled with the kernel function in GP model.

3.2.2.2 Statistical inference and selecting kernel function in GP-based frameworks

Typically, when evaluating the existence of spatial patterns within the data, an assessment is made by testing the alternative hypothesis, which suggests the presence of a spatial covariance term in the model, against the null hypothesis, where the spatial covariance term is set to zero, indicating the absence of spatial variability. This comparison between the model fitted under the alternative hypothesis and the null model forms the basis of a significance testing procedure. This often involves conducting significance tests and drawing conclusions based on p-values in frequentist approaches. For example, in model (3.1), testing SVG is equivalent to testing $H_0 : \sigma_s^2 = 0$.

As previously mentioned, selecting the appropriate kernel function for computing the spatial covariance matrix is a critically important step in identifying spatial patterns within the data. Ideally, the kernel function should accurately capture the true underlying relationship between the y values and the spatial coordinates s. In practice, the actual underlying function remains unknown, and the closer the chosen kernel function approximates the true functional form, the more precise the model specification becomes, rendering the test more robust and powerful.

SpatialDE employs a squared exponential covariance function(a.k.a Gaussian kernel function or radial basis kernel function) to compute the spatial covariance matrix:

$$K_{i,j} = k(s_i, s_j) = exp(-\frac{|s_i - s_j|^2}{2l^2})$$

The hyperparameter l, recognized as the characteristic length scale or bandwidth, determines how rapidly the covariance decays as a function of distance and is typically chosen by grid search. SOMDE also uses the squared exponential (Gaussian) kernel in their model with ten different length scales and chooses the one that achieves the highest log-likelihood ratio value. GPcounts uses linear or periodic kernel based on BIC values. SPARK asserts that relying on a single kernel restricts the ability to robustly identify spatially variable genes across diverse spatial patterns. Therefore, SPARK (and SPARK-G) adopts an approach involving a total of ten distinct spatial kernels. These comprise five periodic kernels (e.g., Cosine kernels) with varying periodicity parameters and five Gaussian kernels with different smoothness parameters. SPARK proceeds to compute ten p-values, each derived from a different test employing these various kernel functions. These p-values are subsequently combined using the Cauchy combination rule [100, 101]. Similar to SPARK, SpatialDE2 incorporates a variety of pre-defined kernels with varying structures and length scales. It also offers the flexibility to conduct an omnibus test as an alternative to independently testing each kernel and subsequently merging the p-values. nnSVG posits that genes can potentially display a vast spectrum of spatial patterns, and using the same set of kernel functions for all of the genes might lead to less powerful tests. They consider the use of an exponential covariance function as a kernel function where the length scale parameter of the kernel function is fitted for each gene, which allows capturing the unique spatial variability pattern of the gene. CTSV uses five different sets of functional forms for $h_1(s_{.1})$ and $h_2(s_{.2})$, which includes linear functions, squared exponential functions, and periodic functions with different sets of scaling parameters and the five p-values calculated from five different forms are combined using the Cauchy combination method.

Although different models discussed here have some similarities in testing procedures, the model fitting techniques implemented and the testing procedures utilized are different and are summarized in Table 3.2.

Table 3.2 List of some popular SVG detection methods with model-fitting and testing information.

Method	Bayesian/ Frequentist	Model fitting and parameter estimation	Hypothesis testing method
SpatialDE	Frequentist	Maximizing marginal log likelihood	Likelihood ratio test
SpatialDE2	Frequentist	Only null model parameters needs to be estimated by BLUP	Score test based on Zhang and Lin[102]
SPARK	Frequentist	Approximate-inference algorithm based on the PQL approach	Satterthwaite method on the basis of score statistics
SPARK-G	Frequentist	Maximum likelihood	Score test
nnSVG	Frequentist	Fast optimization algorithms for NNGP models (BRISC R package)	Likelihood ratio test
SOMDE	Frequentist	Gradient optimization	Likelihood ratio test
CTSV	Frequentist	Approx. maximum likelihood using conjugate gradient(CG) algorithm	Wald tests (R package pscl)
GPcounts	Frequentist	Optimization of log marginal likelihood by variational approximation	Likelihood ratio test
BOOST-GP	Bayesian	Sampling from posterior using MCMC	Bayes Factor or posterior probabilities of inclusion (PPI)

The statistical power of GP-based methods hinges on the selection of kernel functions[77], which can complicate the model selection and limit SVG detection power. To address this challenge, the authors in [92] introduced BOOST-MI. This novel approach employs Bayesian modeling of spatial transcriptomics data via a modified Ising model to identify SV genes. As an initial step, BOOST-MI takes normalized gene expression data as input and dichotomizes the normalized expression levels into a binary spatial pattern. Subsequently, BOOST-MI proceeds to identify a wide spectrum of spatial patterns displayed by the genes by inferring the Ising model interaction parameter within a Bayesian framework. It achieves this by generating samples from the posterior distribution of the parameters through a double Metropolis-Hastings (DMH) algorithm[103]. Subsequently, it computes the Bayes factor based on these posterior samples, which are then used for selecting SV genes.

Trendsceek[80], one of the earliest published SVG detection methods, models data as marked point processes, where they assign points to represent the spatial locations of spots and marks on each point to represent expression levels. The pivotal objective of Trendsceek revolves around evaluating the dependency between the spatial distribution of points and their respective marks through pairwise analyses as a function of the inter-point distances. The underlying premise is that if there exists no dependency between marks and point locations, the resulting scores should remain constant across various distances. A resampling procedure is executed to gauge the significance of a gene's spatial expression pattern, involving permutations of expression values that create a null model with no spatial expression dependency.

Similar to Trendsceek, ScGCO[81](single-cell graph cuts optimization) method also models gene expression data as a marked point process where points represent the spatial locations of measured cells or spots, and marks are discrete gene expression states (such as, down-regulated or up-regulated) associated with points. It analyzes the dependency of points with a specific mark on spatial locations using a hypothesis test. Under the null hypothesis (i.e., no spatial dependency), it assumes that points with a specific mark in a 2D space are distributed in a completely random fashion and can be described by a homogeneous spatial Poisson process. Genes with spatial regions whose number of cells/spots of specific marks are associated with statistically significant low probabilities under the null model are selected as SVG.

3.2.3 Overview of model-free frameworks

There are other SVG detection methods such as SPARK-X[88], sepal[82], GLISS[83], and SINFONIA[89] which do not attempt to model the data generation process or rely on distributional assumptions. Instead, they use model-free techniques to detect SVGs. The authors introduced sepal[82] (Spatial Expression Pattern Locator), an innovative method that leverages transcript diffusion simulations to identify genes exhibiting spatial patterns. It simulates transcript diffusion within the spatial domain and measures the time required for convergence. The core idea is that transcripts with random spatial distributions will converge more quickly or reach a homogeneous state faster compared to those with distinct spatial patterns. Consequently, the diffusion time serves as an indicator of a gene's degree of spatial variability. Genes with longer diffusion times exhibit less spatial randomness. Therefore, ranking genes based on this indicator and selecting the top-ranked genes as SVGs is a logical approach.

SINFONIA[89] offers a scalable approach to initially identify spatially variable genes through ensemble strategies as part of its spatial transcriptomic data analysis, with the ultimate goal of deciphering spatial domains. SINFONIA initially identifies the k nearest neighbors in Euclidean space for each spot and builds a Spatial Neighbor Graph (SNG) using the weight matrix where the (i, j)th element is determined by a function of the distance between the ith and jth spot. Next, SINFONIA calculates Moran's I and Geary's C statistics for each gene based on the weight matrix W to assess spatial autocorrelation. The underlying concept is that genes with more pronounced spatial autocorrelation exhibit more organized spatial expression patterns.

HEARTSVG[96] utilizes a unique, distribution-free, test-based approach that focuses on identifying non-SVGs first and then infers the presence of SVGs using this information. The process involves assessing serial autocorrelations within the marginal expressions across the global spatial context to pinpoint non-SVGs. This, in turn, enables the automatic recognition of all other genes as SVGs, regardless of their spatial patterns. HEARTSVG asserts its superiority in terms of robustness and computational efficiency by abstaining from assumptions about specific underlying spatial patterns for these variable genes.

SPARK-X[88] is a nonparametric method grounded in the following insight: if y is independent of s, then the spatial distance between two locations i and j would also be unrelated to the gene-expression difference between those two locations. SPARK-X constructs two $N \times N$ projection covariance matrices: (1) The expression covariance matrix based on gene expression levels; and (2) the distance covariance matrix based on all spatial locations. It employs a test statistic derived from the product of these two covariance matrices to evaluate the independence between the gene expression (y) and the spatial coordinates (s). In simpler terms, if gene expressions are indeed independent of spatial coordinates, the product of these covariance matrices will yield a small value. Conversely, if gene expressions are dependent on the spatial coordinates, the product of the matrices will yield a large value.

Similar to the kernel matrix used in methods like SpatialDE or SPARK, the statistical power of the SPARK-X test inevitably hinges on how the distance covariance matrix is constructed and how well it aligns with the true underlying spatial patterns exhibited by the gene of interest. To ensure robust identification of spatially varying genes across diverse spatial expression patterns, SPARK-X explores various transformations of the spatial coordinates (s) and subsequently generates distinct distance covariance matrices. Specifically, the algorithm applies five Gaussian transformations with varying smoothness parameters and five cosine transformations to the spatial coordinates (s). This process results in the creation of eleven distinct p-values, corresponding to the ten transformed distance covariance matrices and the original one constructed using the original coordinates. These individual p-values are then combined using the Cauchy combination method. MULTILAYER[98] treats spatially transcriptomics data as a raster image and uses digital image strategies to resolve tissue substructures. The basic unit in MULTILAYER is the "gexel", gene expression element analogous to a pixel in a digital image. The gene expression levels per gexel relative to the average gene expression are computed within the tissue. Genes are considered upregulated or downregulated when their normalized read counts per gene are above or below the average behavior, respectively. Differentially expressed genes are ranked based on the number of related gexels, providing a rapid view of genes that are overrepresented on the digital map based on their relative expression.

GLISS[83] (Graph Laplacian-based Integrative Single-cell Spatial Analysis) utilizes a graph-based feature learning framework to detect and discover SVGs and recover cell locations in scRNA-seq data by leveraging spatial transcriptomic and scRNA-seq data. The workflow involves multiple steps. First, SV genes are identified from ST data using graph-based feature selection. Next, it determines the cells of interest in the scRNA-seq data based on unsupervised learning methods and leverage these selected SVGs to discover new SVGs in scRNA-seq data. The final goal of this workflow is to cluster genes based on their spatial patterns.

The BSP (Big-Small Patch)[99] method, introduced in a recent publication, utilizes a non-parametric model for the identification of spatially variable genes in 2D or 3D spatial transcriptomics data. The approach involves taking normalized spatial transcriptomics data as input. It defines big and small patches for each spatial spot based on neighboring spots with larger or smaller radii, respectively. The method then calculates local means of gene expression for both big and small patches. Following this, it calculates the ratio between the variances of local means for each gene, approximating a log-normal distribution for the distribution of these ratios. Subsequently, a p-value is determined for each gene based on this approximated distribution.

3.3 Statistical Inference with Multiple Testing Control

We have previously discussed both model-based and model-free methods for detecting SVGs. The mathematical models employed for capturing the data generation process and the innovative model-free SVG detection technique have proven valuable for uncovering significant SVGs that offer critical biological insights. However, from a statistical perspective, concerns arise regarding the potential for false discoveries of genes that lack genuine spatial variability. This concern becomes more pronounced when a large number of genes are simultaneously tested across most frameworks. If the false discovery rate or type 1 error is not adequately controlled, it may lead to incorrect conclusions and the selection of numerous genes that exhibit false spatial variability.

Various methods have been developed for multiplicity correction (MC) to address this concern. Some methods analytically constrain the false discovery rate (FDR) to remain below a predetermined threshold, while others do not analytically control the FDR and simply select a user-specified number of top genes as SVGs. Researchers may choose a method that aligns better with their research goals and the type of downstream analysis they intend to perform. In Table 3.3, we present an overview of these methods, organized around these critical questions. The permutation-based method is usually considered as the golden standard method as it is purely data-driven and distribution free. However, it is the least scalable one since it is computationally more demanding. The FDR-based methods have been the commonly applied ones since they offer type I error control while maintaining high power compared to the Bonferroni method. Nevertheless, depending on the downstream analysis goal, it is not necessary to strictly enforce the MC rule. For example, when the goal is to find the low dimensional embedding of genes, such as in spatial PCA analysis [104], people usually choose top ranked genes for further analysis. In such cases, strictly enforcing MC is not needed.

3.4 Exploring Performance, Advantages, and Limitations

In the preceding sections, we have explored the complexities associated with spatial count data. In many instances, these count data are characterized by sparsity and overdispersion. Section 2 of this review classifies modeling frameworks based on whether they directly model the count data or opt for modeling the normalized data. Some literatures [88, 79] argue against modeling normalized data with a Gaussian distribution due to concerns that such a parametric approximation may result in reduced statistical power and difficulties in controlling type 1 errors, especially when dealing with small p-values.

On the other hand, methods that employ normalized count data, such as SpatialDE, SPARK-G, and nnSVG, offer advantages, including simpler model structures and reduced computational challenges. Notably, SPARK employs a dual modeling approach, encompassing both an overdispersed Poisson model (SPARK) and a Gaussian model (SPARK-G) for count data analysis. They declare that SPARK-G exhibits significantly improved computational efficiency compared to the Poisson version of SPARK. Moreover, SPARK-G may demonstrate greater resilience to model misspecification, potentially enhancing its effectiveness in specific data applications.

Although many researchers prefer to model count data directly, there is no consensus on the preferred approach for directly modeling count data either. While some opt for Poisson distribution models, others argue that it may be insufficient to address issues of overdispersion, suggesting that a negative binomial distribution is more suitable in such cases. Furthermore, when data exhibit extreme sparsity, the utilization of a zero-inflated Poisson or negative binomial model may be more logical, although it tends to introduce greater complexity into the model. But we need to note that direct modeling of sparse count data with a negative binomial distribution or other over-dispersed Poisson distributions incurs algorithm stability issues [88, 108, 90].

With the continuous evolution of spatial transcriptomic technologies, researchers now have access to increasingly vast and high-resolution spatial datasets. Analyzing these extensive datasets demands the use of efficient and scalable methods for downstream analysis. Notably, approaches like Trendsceek and BOOST-GP impose substantial computational demands. In a study referenced from SRTsim[109], it was observed that when applying these methods to synthetic data, Trendsceek (v.1.0.0) required approximately 10 hours, while BOOST-GP needed about 8 hours to analyze a single synthetic dataset containing 1000 genes and 673 locations. In the same research context, SOMDE (v.0.1.8) struggled, failing to process nearly 90 percent of the genes and yielding NA values.

Another comprehensive comparison, outlined in a review paper[110], assessed the performance of various SVG detection methods. The evaluation considered computational time and memory usage across 20 diverse spatial datasets, each varying in the number of spots or samples. Among the methods examined, including SpatialDE, SPARK-X, nnSVG, SOMDE, Giotto KM, and Giotto Rank (both are implemented in the Giotto package), SPARK-X emerged as the swiftest, with SOMDE following as the second-best option, albeit notably slower than SPARK-X. SpatialDE exhibited poorer performance in larger datasets, while nnSVG proved faster than SpatialDE for larger datasets but relatively slower for datasets with fewer spatial locations. In particular, SPARK-X [88] scales linearly with the number of spatial locations, while other methods scale cubically (e.g., SpatialDE) or quadratically (SpatialDE2, SPARK).

In terms of peak memory usage, study [110] revealed that SOMDE consumed the least memory, with SPARK-X ranking second. Conversely, SpatialDE demonstrated high peak memory consumption. Considering the trade-off between speed and memory usage, SPARK-X and SOMDE emerged as the two most efficient methods, as determined by the experiment. Furthermore, the evaluation included other methods such as Giotto KM, Giotto Rank, and Moran's I , but none of these alternatives matched the efficiency of SPARK-X or SOMDE based on the experimental findings.

In summary, each modeling framework comes with its own set of pros and cons, necessitating careful consideration of the trade-off between computational efficiency/cost and performance when selecting the most suitable approach for analyzing spatial count data. The model-free or nonparametric approaches do not try to capture the data generation process and offer alternative frameworks to detect SVG. Most of the method frameworks are very intuitive but each comes with its own sets of restrictions or assumptions. For example, Trendsceek is a resampling-based method, which incurs a substantial computational load, rendering its application impractical for extensive ST datasets. SPARK-X exhibits impressive performance for high dimensional data, but the authors recommend using it with large sample (e.g., spot) size, say 3,000 or more.

3.5 Assessing Input Data and Model Outputs

For the various methodologies we reivewed so far, some of these approaches primarily focus on identifying genes that exhibit spatial variability across the entire tissue, exemplified by methods like SpatialDE and SPARK. In contrast, others are additionally equipped to detect genes with spatial variability within predefined spatial domains, as seen in nnSVG. Other methods like SpaGCN[111] and STAMarker[112] are designed to identify Spatial domains and detect SVGs within spatial domains.

Additionally, certain methods aim to identify SVGs to facilitate downstream analysis. For instance, SINFONIA, as cited in this work, provides a scalable approach for the initial identification of spatially variable genes using ensemble strategies within the context of spatial transcriptomic data analysis. The ultimate objective of this method is to decipher distinct spatial domains within the tissue.

Furthermore, some of these methods leverage additional information as input, such as single-cell RNA sequencing (scRNA-seq) data, spatial domain information, or tissue-specific markers, in conjunction with spatial transcriptomic data. For instance, CTSV requires scRNA-seq data and a set of marker genes as input alongside the spatial transcriptomic data. It employs deconvolution techniques like SPOTlight[113], RCTD[114], or SpatialDWLS[115] to estimate cell-type proportions for each spatial spot. Ultimately, this approach identifies spatially variable genes specific to different cell types. Similarly, Trendsceek identifies genes with significant spatial trends and subsequently determines the subset of cells occupying spatial regions of interest.

Given the distinct ultimate objectives and input criteria for each method, it would be unfair to evaluate their performance solely based on a single parameter. Rather, the utility or superiority of these frameworks depends on the researcher's specific goals and the nature of their research inquiries. In this context, we present a table that combines these selective frameworks, including details about their typical inputs and primary research objectives (see Table 3.4).

3.6 Publicly Accessible Code for Major SVG Detection Methods

Every prominent SVG detection method featured in this paper has made its code publicly available. Certain frameworks have packages published in CRAN or available as Python modules, while others have shared their code on Github, and the package can be installed directly from Github. Here, we have compiled a list of the packages and repositories associated with these techniques, along with the coding language they have used (see Table 3.5). This compilation aims to facilitate convenient access to their respective code bases, making it easier for researchers to choose a method based on their preferred programming language.

3.7 Summary and Outlook

We systematically reviewed recently developed frameworks for identifying spatially variable genes and grouped them into different categories and delved into the unique aspects of their models and underlying principles. Here, we provide a brief discussion encompassing various facets, including pre-processing steps, modeling frameworks, inference techniques, scalability, and practical applicability of these frameworks. We explored the performance of select methods as reported in previously published papers. Nevertheless, it is essential to note that we refrained from conducting evaluations based solely on the number of SVGs detected or the trade-off between statistical power and FDR. This decision arises from the fact that the methods discussed in this paper often serve different research objectives, each tailored to specific research questions. For example, a method primarily focused on spatial clustering may yield similar outcomes when considering the top 100 genes versus the top 110 genes. In contrast, a method geared toward accurately identifying genuine SVGs and scrutinizing individual SVGs to glean deeper insights into biological mechanisms may prioritize stringent control of false discovery rates, making it a pivotal concern in their evaluation. The evaluation criteria must align with the unique goals and nuances of each method, akin to comparing apples to oranges when attempting to gauge their performance solely based on the number of SVGs selected.

This paper[110] has previously investigated several methods for detecting spatial gene expression variations and benchmarked their performance based on different measures. It reported that, although each SVG detection method successfully identifies a significant number of SVGs, there is limited overlap in the SVGs detected when a significance cutoff is applied to filter the SVGs. The study's simulation analysis revealed that, in most cases, the estimated FDRs do not accurately reflect the true FDRs. These findings indicate that there is room for improvement in the commonly used methods for SVG detection and their associated FDR control approaches.

In the context of Gaussian process based methods, one potential issue could be related to the selection of kernel function. For instance, as an improvement to spatialDE, approaches like SPARK and SPARK-X employ a variety of different kernels to robustly identify various traits. However, they apply the same set of parameter values to all genes, even when these genes may exhibit vastly different spatial patterns. While nnSVG offers improvements by allowing gene-specific kernel function parameter selection, it relies on a single type of kernel function. This opens room for further methodological development for optimal kernel selection when kernel-based methods are applied for SVG detection.

Furthermore, model-free techniques, in many cases, do not analytically control FDR, making it challenging to establish a specific cutoff for selecting SVGs. Many methods claim to detect more SVGs than others, often undetected by alternative methods. However, the mere detection of more SVGs does not necessarily indicate the superiority of a framework if it does not effectively control the FDR. If the goal is to pinpoint the top k (say 1000) SVGs for subsequent analysis without the necessity of precisely quantifying detection uncertainty, these methods can be employed. However, for a more rigorous approach, it is crucial to implement stringent FDR control measures to prevent false discoveries. In our empirical analysis, we observed that numerous methods exhibit elevated false positive rates with inflated p-values (data not shown). There is an urgent demand for the development of more rigorous statistical approaches to enhance false positive control.

Model-based SVG detection techniques frequently incorporate covariate variables, such as cell type information or domain structure information, into the model. However, the unavailability of this information alongside spatial transcriptomics data poses a challenge. It remains unclear how to obtain covariate information without utilizing the same data twice—once for identifying covariates and again for detecting SVG. Addressing this issue represents an ongoing challenge within the framework of SVG detection techniques. Hopefully, future methods will be developed to effectively bridge this gap.

Finally, we acknowledge that benchmarking existing methods is essential to determine their efficiency in terms of scalability and accurate selection of SVGs. This is crucial for ensuring proper downstream analysis. To address this need, we present a foundational outline of a benchmarking design. For data generation, they can be simulated through methods such as SRTsim [109] and scDesign3 [116]. Both methods are capable of simulating datasets that emulate the structure of a real spatial dataset by learning their parameters. SRTsim offers a ShinyR platform where spatial patterns can be visualized, and parameters can be configured to generate count data for spatially variable gene expression. Diverse datasets containing both spatial and non-spatial genes can be simulated, with various spatial effect strengths, sparsity levels, distinct spatial patterns. scDesign3 is another model-based simulation machinery where users can use real data to estimate parameters which allow for a wide range of simulation scenarios, from homogenous cell populations to complex tissues with diverse cell types. Benchmarking involves assessing the performance of implemented methods by checking their power and false discovery rate (FDR) in detecting SVGs, their scalability, as well as the impact on specific downstream analysis such as spatial domain detection.

In summary, this chapter provides a selective survey of recently published and archived literature on SVG detection, offering an analysis of their practical utility, adaptability, innovation, and constraints from various practical perspectives. This effort aims to facilitate new researchers in gaining a holistic understanding of the available methods and assist them in selecting a framework aligned with their specific research needs and questions.

Table 3.3 Compilation of SVG detection techniques Grouped by the method's control of False Discovery Rate (FDR).

Method	If framework analytically controls FDR	How SVGs are selected
Trendsceek	Yes	Permutation based p-values, Benjamini–Hochberg procedure[105] for MC
SpatialDE	Yes	Analytically estimated p-values, q-value method[106] for MC
SpatialDE2	Yes	Analytically estimated p-values, Benjamini–Yekutieli procedure[107] for MC
SPARK	Yes	Analytically estimated p-values, Benjamini–Yekutieli procedure for MC
SPARK-G	Yes	same as SPARK
SPARK-X	Yes	same as SPARK
nnSVG	Yes	Analytical approximate p-values, Benjamini–Hochberg method for MC
BOOST-GP	Yes	Based on Bayesian FDR controlled PPI threshold
GLISS	Yes	Permutation based p-values, Benjamini–Hochberg procedure for MC
scGCO	Yes	Analytically estimated p-values, Benjamini–Hochberg procedure for MC
CTSV	Yes	Analytically estimated p-values, q-value method for MC
HEARTSVG	Yes	Analytically estimated p-values, MC by Bonferroni/Holm/Hochberg
GPcounts	Yes	Analytical or permuted p-values, q-value method for MC
BSP	yes	Analytically estimated p-values, q-value method[106] for MC
SOMDE	No	Top ranked genes based on spatial variability score
sepal	No	Top k genes with highest ranks
SINFONIA	No	Top k genes with highest score and an ensemble technique
BOOST-MI	No	Based on specific Bayes Factor threshold
MULTILAYER	No	Based on the two-fold threshold of a test statistic

Table 3.4 List of selective methods with input data type and main goal.

Method (Publication)	Input data	Main goal
SpatialDE(2018)	ST data	Finding SVG Spatial gene-clustering
SpatialDE2(Archived,2021)	ST data	Tissue region segmentation Finding SVG Spatial gene-clustering
SPARK(2020)	ST data	Finding SVG
SPARK-X(2021)	ST data	Finding SVG
nnSVG(2023)	ST data	Finding SVGs across tissue or within spatial domains
BOOST-GP(2021)	ST data	Finding SVG
SOMDE(2021)	ST data	Finding SVG
sepal(2021)	ST data	Finding SVG Spatial gene-clustering
SINFONIA(2023)	ST data	Finding SVG for deciphering spatial domains
BOOST-MI(2022)	ST data	Finding SVG
scGCO(2022)	ST data	Finding SVG
BSP	ST data	Finding SVG
HEARTSVG (Archived,2023)	ST data	Detecting SVG and spatial domain
MULTILAYER(2021)	ST data	Detecting SVG, dimensionality reduction, spatial clustering and more
STAMarker(Archived,2022)	ST data	Spatial domain-specific variable genes
GPcounts(2021)	ST data scRNA-seq data	Finding SVG, identifying gene-specific branching locations and more
SpaGCN(2021)	ST data histology image data	Identifying spatial domains and SVG in domain
Trendsceek(2018)	ST data scRNA-Seq data	Finding SVG Identifying cells in spatially significant gene expression regions
GLISS(Archived,2020)	ST data scRNA-seq data	Finding SVG, recovering cell locations in scRNA-seq data and gene-clustering
CTSV(2022)	ST data scRNA-seq set of marker genes	Detecting cell-type-specific SVG

Table 3.5 List of methods with implementing code language and package site.

Method	Code language	Package or GitHub or vignette
SpatialDE	Python	https://github.com/Teichlab/SpatialDE
SpatialDE2	Python	https://github.com/PMBio/SpatialDE
SOMDE	Python	https://pypi.org/project/somde https://github.com/XuegongLab/somde
sepal	Python	https://github.com/almaan/sepal 10.5281/zenodo.4573237
GLISS	Python	https://github.com/junjiezhujason/gliss
SINFONIA	Python	https://github.com/BioX-NKU/SINFONIA
ScGCO	Python	https://github.com/WangPeng-Lab/scGCO
MULTILAYER	Python	https://github.com/SysFate/MULTILAYER
GPcounts	Python	https://github.com/ManchesterBioinference/GPcounts
BSP	Python	https://github.com/juexinwang/BSP/
Trendsceek	R	https://github.com/edsgard/trendsceek
SPARK SPARK-G SPARK-X	R	https://github.com/xzhoulab/SPARK https://xzhoulab.github.io/SPARK/01_about/
nnSVG	R	https://github.com/lmweber/nnSVG
BOOST-MI	R	https://github.com/Xijiang1997/BOOST-MI
CTSV	R	https://bioconductor.org/packages/devel/bioc/html/CTSV.html
HEARTSVG	R	https://github.com/cz0316/HEARTSVG.git
BOOST-GP	R/C++	https://github.com/Minzhe/BOOST-GP

3.8 Downstream analysis: Spatial domain detection

In spatially resolved RNA-seq data, researchers aim to identify spatially variable genes (SVGs) as an initial step in the analysis, although detecting SVGs is not the primary objective. The main goal lies in the downstream analysis, which relies on the SVGs identified in the initial step. Key downstream analyses include detecting spatial domains, spatial trajectory inference on the tissue, and high-resolution spatial map reconstruction. etc. Here, our primary focus is on spatial domain detection. Various algorithms have been proposed by authors that are equipped to perform this task, such as SpaGCN[111], SpatialPCA[104], BayesSpace[117] etc. Among these, we opt for SpatialPCA due to its superior performance compared to other existing algorithms. In this section, we elaborate on the method and domain detection steps conducted by SpatialPCA in greater detail, as we have employed this efficient algorithm in our framework, which will be described in the next chapter.

3.8.1 SpatialPCA

Spatial transcriptomic datasets are typically vast and high dimensional, with a significantly higher number of genes (m) than spots (N). Handling the entire dataset directly poses challenges. Moreover, disregarding spatial information while seeking low-dimensional embeddings of the data from gene expression values results in information loss. To address this issue, the Authors propose SpatialPCA, a method for spatially-aware dimension reduction. SpatialPCA effectively extracts a low-dimensional representation of spatial transcriptomic data while preserving both biological signals and spatial correlation structures. This condensed representation of spatial transcriptomic data can be further utilized for many downstream analysis, for example spatial domain detection, which we are interested in.

3.8.1.1 Method to obtain Spatial PCs

Y is denoted as the available $m \times N$ normalized gene expression matrix. The ji^{th} element of Y, $y_{ji}(s_i)$ represents the gene expression measure for j^{th} gene on i^{th} location. SpatialPCA aims to perform dimension reduction on the gene expression matrix and infer a $d \times N$ factor matrix Z that represents a low-dimensional embedding of Y.For dimension reduction, consider the following latent factor model

$$Y = (XB)^T + WZ + E$$

where X is the covariate matrix, B be the corresponding coefficients. W is a $m \times d$ factor loading matrix and E is the $m \times N$ residual matrix, where $E_{ji} \sim N(0, \sigma_0^2)$. The model is currently unidentifiable, so following the probabilistic principal component analysis model (PPCA)[118] an orthonormality constraint is imposed on $W : W^T W = I_d$.

Unlike the independent assumption on Z ($Z \sim N(0, I)$) in PPCA and similar models, here the elements of Z are not independent because in spatial transcriptomic data, the spots spatially close to each other are likely to be more similar than the ones far from each other. This is because the spots closer to each other might share similar cell type composition and display similar gene expression measures. In order to promote consistency among neighborhood factor values and enhance the exchange of information among adjacent areas for factor estimation, it is assumed that the l^{th} factor values $Z_{l.}^{1 \times N}$ across N locations follows a multivariate normal distribution:

$$Z_{l.} \sim MVN(0, \Sigma_l)$$

where the covariance matrix Σ_l is constructed using the Gaussian kernel

$$\Sigma_l = \sigma_0^2 \tau_l K$$
$$\text{where } K(s_i, s_j) = exp(-|s_i - s_j|^2/\gamma)$$

where γ being the bandwidth parameter and $\sigma_0^2 \tau_l$ is the variance component that is scaled with respect to the residual error variance σ_0^2.

With the model specifications provided above, the factor loading matrix W and the factor matrix Z, along with the hyperparameters (τ, σ_0^2), are inferred through maximum likelihood-based optimization. Specifically, both B and Z are integrated out initially to obtain a marginal likelihood, based on which τ, σ_0^2, and W are inferred. Subsequently, Z is estimated by computing their posterior mean conditional on the estimated τ, σ_0^2, and W. The rows of the final matrix Z are called Spatial PCs.

3.8.1.2 Spatial domain detection with spatial PCs

The spatial PCs Z inferred from SpatialPCA can be combined with various methods already developed in the scRNA-seq literature to enable a range of downstream applications in spatial transcriptomics. Spatial domain detection, aimed at segmenting the tissue into multiple structures or microenvironments, each characterized by a distinct transcriptomic profile, is facilitated. For spatial domain detection, it is formulated as a clustering problem on the inferred spatial PCs Z. Specifically, standard clustering algorithms like walktrap or Louvain are applied on Z to categorize spatial locations into different spatial domains. Due to the critical spatial correlation information present across locations in Z, clustering based on the spatial PCs would result in similar cluster assignments in neighboring locations, leading to smooth boundaries in the detected tissue structures. This advantageous property of SpatialPCA is harnessed by our framework, to be described in the next chapter, to enhance the accuracy of spatial domain detection.

CHAPTER 4

CSVG: IMPROVED SPATIAL DOMAIN DETECTION BY SPATIALLY VARIABLE GENE CLUSTERING ADJUSTING FOR CELL TYPE EFFECT

4.1 Introduction

Recent advancements in spatially-resolved transcriptomics (SRT) technology have revolutionized our ability to acquire comprehensive gene expression data for thousands of genes across tissue locations in multiple samples. The number of genes and spatial resolution vary depending on the specific technology employed. However, regardless of the technology and resolution, spatial transcriptomic data facilitate the exploration of various biological questions.

Often a fundamental initial step in the analysis of SRT data involves identifying spatially variable genes (SVGs). These genes exhibit expression level variations either across the entire tissue or within predefined spatial domains. In recent years, there has been an abundance of research and the development of new methods to address the challenge of detecting SVGs[119][120]. Although the detection of SVGs lets us visualize the spatial patterns in the tissue which might offer some level of biological insights about the tissue of interest, the main use of SVGs lies in downstream analysis, specifically for spatial domain detection. Spatial domains are distinct and functionally specialized anatomical structures within tissue, each distinguished by unique local characteristics including cell-type composition, transcriptome heterogeneity, and cell-cell interactions[121][122][123]. Detecting these domains is crucial for understanding their collaborative role in tissue functions and development stages. To achieve this, a set number of top SVGs is typically selected, and spatial domains are identified using these top SVGs.

However, using an arbitrary number of top SVGs might not represent all the spatial patterns exhibited by the SVGs. As previously argued [82], Some dominant patterns may overshadow less pronounced yet relevant patterns. Previous methods, like SpatialDE[75], SPARK [77] and Sepal[82], attempted to classify SVGs into groups with similar spatial patterns, aiding in a more holistic representation of results. It's important to highlight that classifying spatially variable genes (SVGs) is a challenging task requiring specialized methods. Simple clustering approaches are inadequate as they overlook spatial information [75]. However, in existing methods challenges arise regarding the selection of the unknown number of spatial pattern groups and selection of other parameters, as well as the unclear impact of classification on downstream analysis. Here, we propose an efficient method cSVG to classify SVGs into clusters and explore the benefits of this clustering step in the final goal of spatial domain detection.

The concept of clustering SVGs carries a significant biological rationale. Researchers have already identified many SVGs as the markers for specific cell types [124],[125],[126],[127], yet detecting these cell type-specific SVGs presents a formidable challenge. As distinct spatial patterns are associated with distinct cell type compositions, it follows that different cell type-specific SVGs would manifest distinct spatial patterns. Consequently, clustering SVGs with similar patterns can be viewed as a means of segregating distinct cell type-specific SVGs.

cSVG is a Gaussian process-based method that initially identifies SVGs and then establishes a dependency map among these SVGs using an intuitive approach (see methods 4.2). This map links each SVG with other SVGs exhibiting similar spatial patterns, ultimately clustering similarly expressed SVGs together (see Figure 4.1A). The resulting SVG-clusters from the cSVG algorithm can serve as inputs for further downstream analysis.

For performing downstream analysis, we employ a well-known dimension reduction technique tailored for spatial data, SpatialPCA[104], to derive low-dimensional embeddings specific to each SVG-cluster. These embeddings are subsequently utilized for spatial domain detection. In each example considered in this study, whether through simulation setups or real data analysis, we compare our findings with those obtained from the SpatialPCA framework, which generates low-dimensional embeddings for all top SVGs collectively and subsequently performs the same spatial domain detection step. This comparison aims to elucidate the advantages of the gene clustering step facilitated by cSVG.

To evaluate the performance of cSVG, we conducted a comparison using a synthetic dataset derived from real human DLPFC cortex data, with known annotations of its 5 layers (4 prefrontal cortex layers and the white matter). See Supplementary Figure F.1 for more details on the synthetic data generation. Figure 4.1B illustrates the domain-based SVG clusters within the synthetic dataset: cluster 1 represents genes predominantly expressed in layer 1 cluster 2 includes genes from cortex layers 2, 3, or 4, while cluster 3 comprises genes overexpressed in the white matter region. The distribution of the number of genes across clusters is uneven in this scenario as shown in figure 4.1C, as frequently evident in real datasets. Upon repeating the analysis on 10 simulated synthetic datasets, and considering ARI scores (Adjusted Rand Index, higher the better) and PAS scores (Percentage of abnormal spots, lower the better, see method section 4.2) (see Figure 4.1D), we observe that our framework provides better domain detection results compared to the SpatialPCA framework. Additionally, Figure 4.1E displays a t-SNE plot [128] representing genes from a randomly chosen simulation outcome. The genes are color-coded according to the gene cluster label identified by cSVG. This visualization demonstrates that genes within each cluster are packed together, indicating the accuracy of gene classification by cSVG.

In this paper, we conducted two types of simulation studies: firstly, to assess the accuracy of the cSVG framework for SVG classification, and secondly, to evaluate the accuracy of domain detection based on the detected SVG-clusters by cSVG using synthetic datasets mimicking real-world scenarios. In addition, we analyzed three publicly available datasets and one newly acquired pancreatic cancer SRT dataset. The publicly available datasets include: 1). The DLPFC human cortex annotated dataset[129], comprising 12 samples; 2). The HER2 human breast tumor annotated dataset[130]; and 3). The dataset from study of human breast cancer biopsies[131]. The newly acquired dataset is the pancreatic cancer dataset, which comes with rudimentary annotation. The application of cSVG and the detection of domains aligns well with the rough annotation. Overall, findings from simulation studies and real data analyses across multiple datasets affirm that utilizing SVG clusters and, consequently, the cSVG framework can markedly enhance the performance of spatial domain detection. This approach holds significant promise to unveil hidden spatial patterns which provide novel insights into spatial heterogeneity of tissue samples.

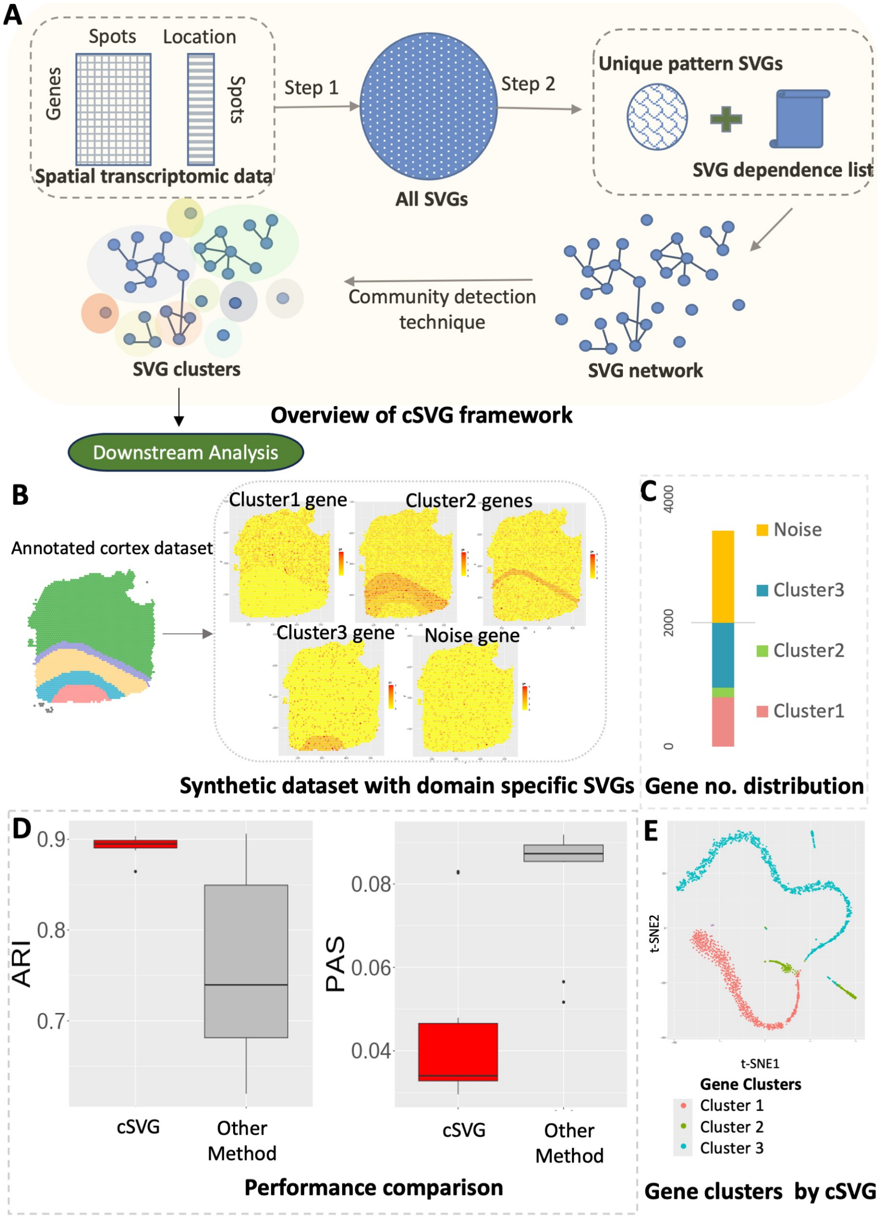


Figure 4.1 (A) Schematic overview of the cSVG framework. (B-E) An example simulation based on a synthetic dataset: (B) The synthetic dataset is generated based on a real dataset with annotated spatial domains. Three types of domain-specific SVGs and noise genes are created. (C) The distribution of the number of each type of gene is presented. (D) Evaluation based on ARI (higher is better) and PAS score (lower is better). (E) Visualization by the t-SNE plot: shown is for a randomly selected simulation result. The gene clusters identified by cSVG are highlighted with distinct colors. The plot illustrates that genes within the clusters are densely grouped together and separated from genes belonging to other clusters.

4.2 Method

In a typical spatial transcriptomics setup, the dataset comprises gene expression measures or counts for m genes distributed across N known spatial coordinates or spots. Suppose $y = (y_1, y_2, ..., y_N)$ represents the gene expression profiles or counts for a particular gene across spatial coordinates (referred to as samples or spots), denoted by $s = (s_1, ..., s_N)$. The spatial locations are typically represented as two-dimensional coordinates, i.e., $s_i = (s_{i1}, s_{i2})$, although coordinates of any dimensionality can be employed. The primary objective of spatially variable gene (SVG) detection models is to identify which genes, among the m genes, exhibit spatial variability across the tissue. In essence, the key goal is to determine whether the gene expression measure y is dependent on or related to the spatial locations where the gene expression measures are sampled.

Consider a scenario where there are k_d spatial domains within the tissue of interest. A spatial domain represents a distinct region or area within the tissue characterized by unique molecular signatures or gene expression patterns. These patterns may arise from specific factors such as cellular composition, anatomical organization, spatial arrangement, or functional attributes.

In reality, the specific spatial domains are typically unknown. However, each domain may be defined with a set of spatially variable genes (SVGs) exhibiting characteristic gene expression patterns in proximity to these domains. To accurately reconstruct the underlying domain structure, it is beneficial to group SVGs based on their spatial patterns and utilize all SVG groups in downstream analysis, rather than relying solely on an arbitrary subset of all SVGs.

With this objective in mind, we present the cSVG framework, organized into two primary steps. The first step involves the selection of SVGs, while the second step utilizes the SVGs selected in the first step to generate an SVG dependency list and subsequently create clusters of SVGs based on this information (see 4.1A). It is worth noting that there exists a plethora of SVG detection techniques in the literature [119] [120], and any of these methods can substitute for the first step in the framework. However, methods that rigorously control the False Discovery Rate (FDR) are preferable, as they enhance accuracy in the subsequent stages of analysis.

4.2.1 Step 1: Selecting SVGs

Like the majority of the SVG detection methods this step is based on the Gaussian process (GP) regression model which models the normalized gene expression y for a given gene assuming the following multivariate normal model:

$$p(y|\mu, \sigma_s^2, \delta, K) \sim MVN(X\beta, \sigma_s^2 K + \delta I), \tag{4.1}$$

where the covariance term is decomposed into a spatial and a non-spatial part, with δI and $\sigma_s^2 K$ representing the non-spatial and spatial covariance matrix, respectively. The $(i, j)^{th}$ element in the kernel matrix K denotes the spatial similarity between the i^{th} and j^{th} spot calculated based on the corresponding coordinates s_i and s_j. The choice of the kernel function plays a very important role in detecting the spatial correlation presented in the gene expressions. $X^{N\times k}$ represents the covariate matrix, while $\beta^{k\times 1}$ denotes the array of corresponding coefficients. This model can incorporate up to $k - 1$ covariates, such as cell type information or domain structure information. However, often such information is either unavailable or deemed untrustworthy. Hence, in practice, we typically employ X solely as the intercept.

When evaluating the existence of spatial patterns within the data, an assessment is made by testing the alternative hypothesis, which suggests the presence of spatial variance in the model, against the null hypothesis, where the spatial variance component is zero, indicating the absence of spatial variability. This comparison between the model fitted under the null and alternative hypotheses forms the basis of a significance testing procedure. This often involves conducting significance tests and drawing conclusions based on p-values in frequentist approaches. In model (4.1), testing if a gene is spatially variable is equivalent to testing $H_0 : \sigma_s^2 = 0$.

Within this framework, we use a straightforward score test to test the underlying hypothesis and a p-value is calculated. More information regarding the test is provided in the supplementary material. Prior studies [77, 132] indicate that Gaussian and Cosine kernels are adept at capturing a wide spectrum of distinct spatial gene expression patterns. Hence, we utilize 10 different kernels (5 Cosine and 5 Gaussian kernels with varying parameter values) following the approach established by SPARK [77]. Denote the number of detected SVGs by m_1 in step 1.

4.2.2 Step 2: Classifying SVGs

This stage categorizes the SVGs identifed in step 1 into cohesive groups. While conventional clustering algorithms can segregate genes into distinct groups based on gene expression, they invariably disregard spatial information. Hence, we require more sophisticated algorithms to classify spatially variable genes more precisely [75].

Intuitively, if two spatially variable genes y_1 and y_2 exhibit similar spatial patterns, they should be correlated which is equivalent to assume $y_1 = y_2 + \epsilon$, where $\epsilon \sim (0, \sigma_\epsilon^2 I)$ is a random noise term with arbitrary variance σ_ϵ^2. Hence, while testing gene y_1, if we use gene y_2 as a covariate in model (4.1) and test the same alternative hypothesis described in step 1, we would expect gene $y1$ to be less or even not significant depending on how strong the correlation between y_1 and y_2 is. On the other hand, if after using another gene y_3 as a covariate in the model we still find gene y_1 to be significant, that would imply that gene y_1 and gene y_3 have different spatial patterns. Given that many SVGs are cell-type marker genes [124][125][126][127], we would expect gene y_1 and y_2 belong to the same cell type while gene y_3 belongs to a different cell type. However, our algorithm does not require such knowledge (often not available in spot-level SRT data) as detailed below.

With this intuition, for each SVG $j, j = 1, \cdots, m_1$, we can select a list of genes S_j which are correlated to gene j and use them as covariates in the model 4.1. If the no. of genes in S_j exceeds 3, we choose the top k_j principal components and include them as covariates in the model. Typically, k_j is chosen such that at least 80% of the total variance is explained by the top k_j principal components. With this, model (4.1) becomes:

$$y(s) = \sum_{l=1}^{k_j} \beta_l PC_l + \alpha(s) + \epsilon(s) \tag{4.2}$$

where PC_l represents the l^{th} principal component, $\alpha(s) \sim N(0, \sigma_s^2 K)$ and $\epsilon(s) \sim N(0, \delta I)$.

There might be different ways of selecting related genes. It could be done by choosing a threshold based on a correlation measure, either linear or nonlinear, such as Pearson correlation, Spearman correlation, distance correlation or kernel correlation. Alternatively, one can run a penalized regression with LASSO[133], Elastic net [134][135] or MCP penalty[136] or perform sure independence screening[137] to select genes correlated with gene j.

In real applications, many SVGs represent marker genes for cell types, exhibiting spatial expression patterns that reflect the distribution of various cell types. Given that marker genes for a given cell type often exhibit similar or strongly correlated gene expression distributions, employing this model adeptly controls for the cell-type specific effect, a factor typically challenging to ascertain or quantify in real-world settings as the cell type information is typically unknown in spot-level SRT data.

Employing this model on all the genes, we can get a set of genes with unique patterns (showing significance under model (4.2)) and a gene dependency list. From the list, we come up with a weighted graph structure, where nodes are SVGs and two nodes are connected if they are correlated. By applying a clustering algorithm such as leiden [138], groups of genes are determined and the unique genes (which are not part of any gene group) create singleton sets. The full algorithm steps are provided in the supplementary file.

4.2.3 Downstream analysis: spatial domain detection

Spatially resolved transcriptomics serve a crucial role in identifying tissue or region substructures through domain detection analysis. Numerous frameworks have been developed for this purpose, to name a few, SpaGCN[111], SpatialPCA [104] and BayesSpace[117]. In our analysis, we opted for SpatialPCA[104] due to its proven superiority in performance over other available algorithms. Furthermore, to conduct domain detection post identification of SVG clusters using our framework, we require an effective low-dimensional representation of the dataset, a task efficiently facilitated by SpatialPCA. SpatialPCA effectively extracts a low-dimensional representation of spatial transcriptomic data while preserving both biological signals and spatial correlation structures. This condensed representation can serve as an input for efficient clustering algorithms, such as the Louvain [139] or Walktrap algorithm [140], facilitating the clustering of spots and thereby identifying spatial domains. The steps for obtaining spatial domains in the SpatialPCA workflow include: 1) Select the top 3000 SVGs and calculate spatial PCs based on these genes, and 2) Use the top 20-30 spatial PCs for spatial clustering using algorithms like Louvein or walktrap.

In our approach, we utilize our framework to identify SVG groups with similar spatial patterns. Within each SVG group, we compute the spatial PCs and aggregate the top spatial PCs from each group to form the final embedding. These aggregated spatial PCs are then used as input in the clustering algorithm to detect spatial domains. The differences between our method and the existing ones are that existing methods use low-dimensional embeddings based on all top SVGs, while our method gets low-dimensional embeddings within each cluster. The within-cluster embeddings can capture unique spatial structures and hence lead to improved spatial domain detection.

4.2.4 Measuring accuracy of domain detection

Detecting domains essentially involves assigning a cluster label to each of the N spots in the tissue sample. Once a framework is implemented for detecting spatial domains, it becomes crucial to measure its accuracy against the ground truth domain labels. We primarily employ a standard clustering evaluation metric, the Adjusted Rand Index (ARI), to assess the similarity between the predicted domain labels and the true labels. Additionally, we utilize the PAS (Percentage of Abnormal Spots) score to quantify the clustering performance of spatial domain detection, following the approach outlined in [104]. This score gauges the randomness of spots located outside their clustered spatial domain computed as the proportion of spots with a cluster label differing from at least six of their ten neighboring spots. A lower PAS score reflects greater homogeneity within spatial clusters.

4.3 Simulation Study

We conducted two simulation studies: one to demonstrate the effectiveness of the SVG clustering performance and another to assess whether the SVG clusters identified by cSVG are beneficial for enhancing domain detection accuracy.

4.3.1 Evaluation of the clustering performance

To illustrate cSVG's capability to accurately classify SVGs, we devise the simulation scenario I in which 100 normalized gene expression datasets were simulated, each consisting of 53 genes (labeled g1-g53) and 2000 spots following model 4.1 with no covariates. The first 10 genes (g1-g10)

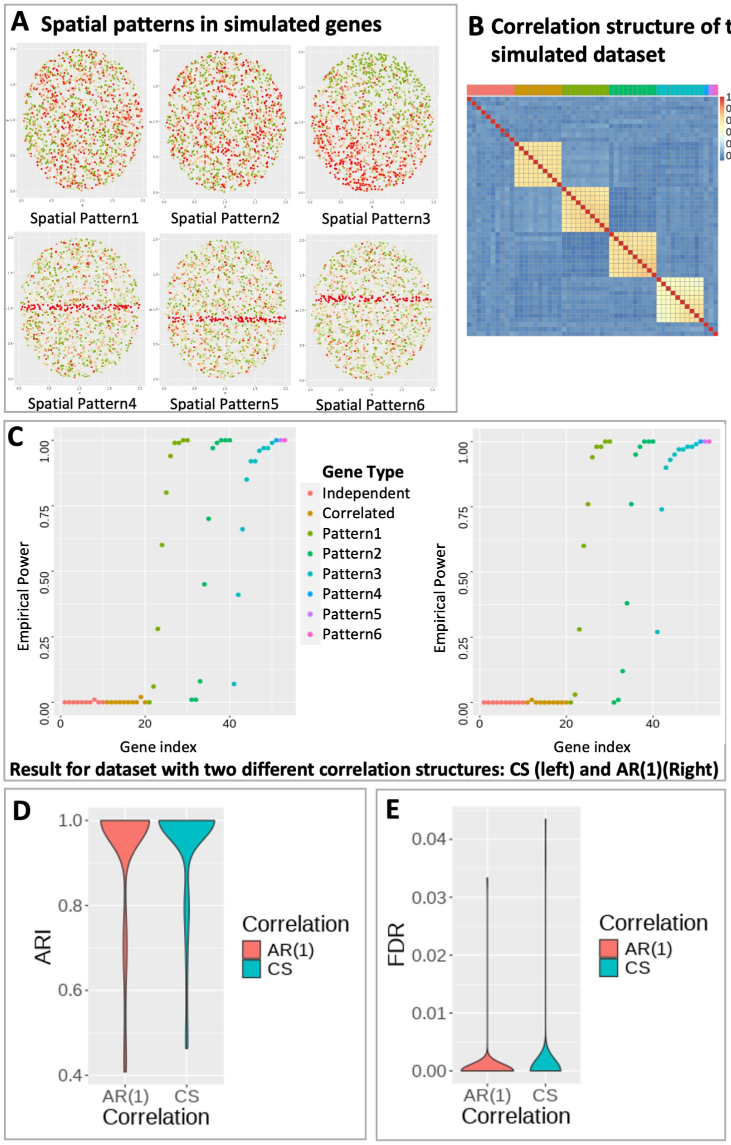


Figure 4.2 Simulation setting for cSVG. A) Six representative genes each with a distinct spatial pattern in the simulated dataset. Green and Red color represents low and high gene expression. B) Correlation heatmap of the simulated dataset with compound symmetry correlation structure within gene groups. Independent: uncorrelated gene group for genes without any spatial pattern; Correlated: correlated gene group for genes without any spatial pattern. Pattern 1-3: correlated gene group for genes with spatial pattern 1-3. Pattern 4-6: single gene with spatial pattern 4-6. C) Empirical power of the SVG detection step at detecting SVGs for the simulated datasets with compound symmetry (Left) and AR(1) (Right) correlation structure within each gene group. D) The distribution of ARI values based on predicted gene clusters for simulation under each correlation structure. E) Empirical FDR distribution for simulation under each correlation structure. Here, false discovery occurs when Correlated or Independent genes show up in any of the final gene clusters.

represent independent noise genes, not correlated with any other gene in the dataset. The next set of 10 genes (g11-g20) exhibit strong correlations among themselves but do not show any spatial pattern. The third (g21-g30), fourth (g31-g40), and fifth (g41-g50) sets of genes represent spatially variable genes with three distinct spatial patterns (Spatial Pattern 1, Spatial Pattern 2, and Spatial Pattern 3, respectively). Figure 4.2A exhibits the six sets of representative genes with distinct spatial patterns. Genes are correlated within the spatial pattern 1-3 as exhibited in Figure 4.2B, and the spatial effect strength increases with the gene index. For example, g21 and g30 are correlated and share the same spatial pattern, but the spatial effect is stronger in g30 compared to g21. The correlation between the genes within a gene group could have different structures. They could display Compound Symmetry (CS) if any pair of genes within a group have the same correlation, i.e., $\rho_{ij} = \rho$. Alternatively, they could demonstrate a first order Autoregressive (AR(1)) pattern if the correlation between two genes decays as their distance increases, i.e., $\rho_{ij} = \rho^{|i-j|}$. Figure 4.2B exhibits CS correlation structure. The last three genes, g51, g52, and g53, each exhibit a unique spatial pattern (Spatial Pattern 4, Spatial Pattern 5, and Spatial Pattern 6, respectively). As the spatial pattern strength increases within each spatial gene group (pattern1-pattern3), the SVG detection power converges towards 1, as expected (see Figure 4.2C). Furthermore, upon comparing this outcome with the performance of the most efficient comparable SVG detection method, nnSVG[78] (where both nnSVG and cSVG use normalized gene expressions), we observed that cSVG detects spatial genes slightly more effectively (see Supplementary Figure F.3). The cSVG framework not only demonstrates efficacy in identifying the true SVGs (g21-g53) but also adeptly categorizes them. The Adjusted Rand Index (ARI) scores, calculated for each simulated dataset, which are computed based on the cluster labels of detected SVGs and their true cluster labels - cluster near 1 in the violin plot in Figure 4.2D, indicating high accuracy of gene clustering. The false discovery rates were also monitored in this study. Here false discovery happens when any of the non-spatial genes (g1-g20) appears in any of the final gene clusters. FDR is calculated as the number of false discoveries divided by the total number of SVG discoveries. As indicated in Figure 4.2E, the FDR values are predominantly distributed near 0, indicating high accuracy of the results.

4.3.2 Evaluation of the spatial domain detection performance

To showcase how the outputs of the cSVG algorithm aid in the downstream analysis of spatial domain detection and enhance its accuracy, we generated synthetic datasets based on the annotated human DLPFC data with sample ID 151670. Supplementary Figure F.1 details the steps involved in generating the synthetic data, ensuring that key features of the original dataset are preserved, such as the distribution of means and variances of all genes. The dataset is annotated with 5 layers (4 prefrontal cortex layers and the white matter layer). After filtering out sparse genes, we ended up with 4,865 genes whose expression were measured in 3484 spots. We first randomly selected 2,000 genes which were converted to SVGs in the generated dataset (See supplementary Figure F.1). The rest of the genes were converted to random noise genes with no specific pattern. Among the 2,000 SVGs, three distinct spatial domain structures were represented (see 4.1B and 4.1C): 800 SVGs correspond to the first cluster, wherein genes are predominantly expressed in the cortex layer 1; 150 SVGs exhibit overexpression in cortex layers 2, 3, or 4; and the remaining 1,050 SVGs from cluster 3 predominantly display expression in the white matter domain region. In such 10 simulated synthetic datasets, we applied our framework, cSVG, to identify gene clusters which were further leveraged for domain detection. We compared the domain detection results of cSVG with the default one by SpatialPCA without an additional clustering step. As we mentioned previously in the introduction 4.1, our framework significantly improves the spatial domain detection accuracy, as evidenced by ARI scores and PAS scores (see 4.1D. Additionally, t-SNE plot[128] from a randomly chosen simulation result for the genes displays the accuracy of gene clustering performance of cSVG. The t-SNE plots for all 10 simulation results were given in Supplementary Figure F.2.

4.4 Real data analysis

4.4.1 Human DLPFC 10x Genomics Visium dataset

We applied cSVG algorithm to the human dorsolateral prefrontal cortex (DLPFC) data[129] generated by Visium from 10x Genomics. Publicly available datasets from 12 human DLPFC tissue samples, obtained from three individuals, can be accessed and downloaded from the link: http://spatial.libd.org/spatialLIBD/. We directly downloaded the processed datasets from the SpatialPCA

Github repository, available at: https://github.com/shangll123/SpatialPCA_analysis_codes. These samples, on average, encompassed 3,973 spots, each manually annotated to one of the six prefrontal cortex layers or white matter.

Our primary analysis is focused on two samples with ID 151670 and 151673, which contain expression measurements of 33,538 genes across 3,498 spots and 33,538 genes across 36,39 spots, respectively. We also analysed the other 10 samples and the results are available in the supplementary files.

We applied our framework to detect the spatial domains (see methods 4.2) for each of the samples. We also followed the SpatialPCA framework provided in https://github.com/shangll123/SpatialPCA_analysis_codes without separating genes into clusters to detect spatial domains to compare with our results. The comparison is based on the ARI scores, utilizing the provided annotations for each sample. We also compared the PAS scores for all the samples. In Figure 4.3A, sample 151670 is presented with annotated spatial domains as the ground truth (left), spatial domains detected by SpatialPCA (middle), and spatial domains detected by our framework (right). The ARI values for the detected domains by SpatialPCA and our framework (0.34 and 0.69, respectively, as shown in Figure 4.3E), along with the PAS scores (0.013722 and 0.006289, respectively, depicted in Figure 4.3F), highlight the superior performance of our framework over SpatialPCA.

Similarly, in Figure 4.3C, the spatial domains for sample 151673 are displayed in the same order: annotated domains, spatial domains by SpatialPCA, and spatial domains by our framework. The ARIs for SpatialPCA and our framework are 0.58 and 0.65, respectively and the PAS scores are 0.028579 and 0.023083, respectively. Combining the results from all the samples, our framework significantly enhances spatial domain detection compared to SpatialPCA (see Figure 4.3G, 4.3H) as confirmed by the increase in the median ARI score.

We visualized the SVG clusters identified by cSVG through the t-SNE plots in Figure 3B and 3D for the two samples. These visualizations highlight the resemblance among genes within the same cluster and the disparity between genes from separate clusters, emphasizing the efficacy of cSVG in delineating SVG clusters.

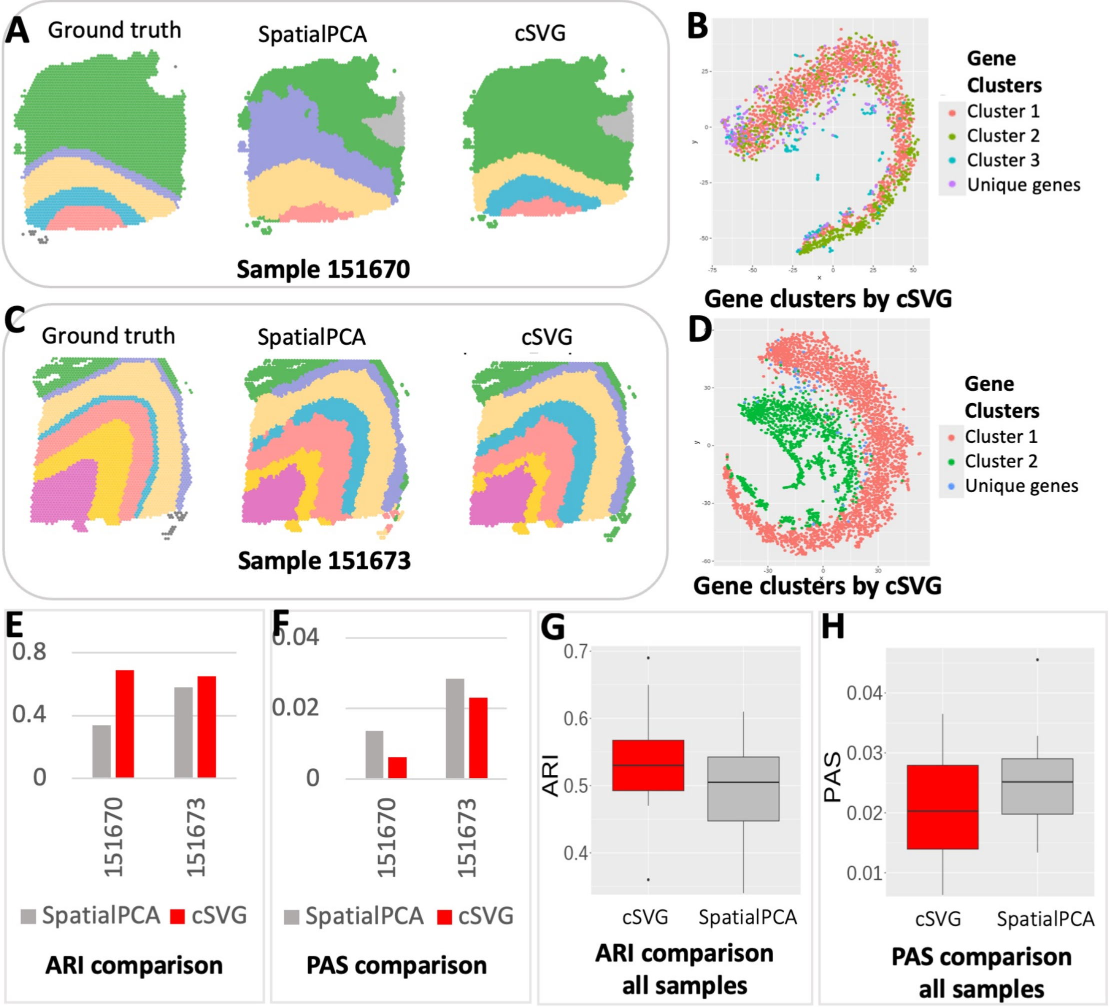


Figure 4.3 Spatial domain detection analysis results of the Human cortex data from DLPFC: (A) The detected spatial domains for Sample 151670: Annotated domain used as the ground truth (left), by SpatialPCA (middle), and by our framework (Right). (B) The t-SNE plot for all the SVGs detected by cSVG for Sample 151670. SVGs are colored based on cluster label calculated by cSVG. (C) The spatial domains detected for Sample 151673: Annotated domain used as the ground truth (left), by SpatialPCA (middle), and by our framework (Right). (D) The t-SNE plot for all the SVGs detected by cSVG for Sample 151673. The fact that SVGs with similar color are close to each other but are away from those with different colors indicates accurate gene clustering. (E) Comparison of ARI score (higher the better) between SpatialPCA and our framework based on these two samples. (F) Comparison of PAS score (lower the better) between SpatialPCA and our framework based on these two samples. (G) Comparison of ARI score between SpatialPCA and our framework based on all 12 samples. (H) Comparison of PAS score between SpatialPCA and our framework based on all 12 samples.

4.4.2 Analysis of HER2 breast tumor data

We applied cSVG to another dataset, the HER2-positive breast tumor data[130], initially comprising 36 tumor datasets from eight individuals (patients A-H), each consisting of 3 or 6 sections. Following SpatialPCA analysis, we selected the H1 dataset, encompassing 15,030 genes measured across 613 spatial locations. We utilized the pre-processed dataset available in the SpatialPCA repository, containing 10,053 genes across 607 spots, and omitted datasets from other samples due to sparse gene expressions or minimal spot coverage. cSVG identified 268 SVGs grouped into three main clusters, along with a few unique pattern genes. The three main gene clusters consist of 84, 48, and 117 genes respectively. The t-SNE plot of the SVGs (see Figure 4.4B) exhibits a similar pattern for genes within the same cluster in close proximity to each other. Comparing spatial domains detected by SpatialPCA and cSVG based on the ARI value, our framework shows better performance (ARI=0.48) than SpatialPCA (ARI=0.44), thereby improving domain detection accuracy (see 4.4C).

4.4.3 Analysis of the pancreatic cancer data

Our final analysis was conducted on a new dataset concerning Pancreatic cancer, obtained from the Henry Ford Health System. This dataset comprises gene expression measurements for 17,943 genes across 3,142 spots, collected from a pancreatic tumor-infested tissue.

Following standard filtering and normalization procedures, we applied our cSVG framework to identify SVG clusters and detect spatial domains. Three primary SVG clusters were identified, each showcasing differential expressions in distinct tissue regions. Representative genes from these clusters are depicted in supplementary Figure F.8.

For each SVG cluster, we extracted low-dimensional embeddings (top SpatialPCs from SpatialPCA), combined them, and applied the Leiden algorithm to identify spatial domains. We repeated this process for the SpatialPCA framework, utilizing the top 20 Spatial PCs from the top 3,000 SVGs. Due to the absence of spot annotations, the exact number of spatial domains is unknown. Therefore, we employed the same Leiden algorithm for the SpatialPCA framework, rather than using algorithms typically utilized by SpatialPCA that require a predetermined number

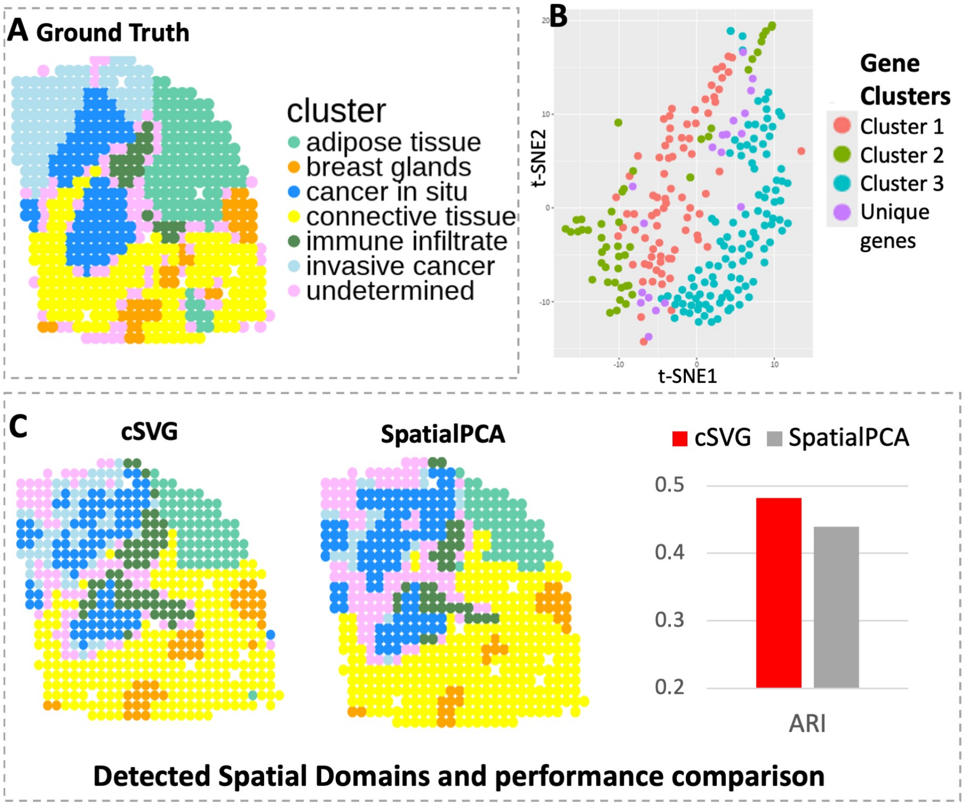


Figure 4.4 Analysis of the HER2 data. (A) Annotated spatial layers considered as ground truth, showcasing 6 known tissue components including cancer-related spots in blue shades. (B) The t-SNE plot illustrating SVGs detected by cSVG, with distinct colors representing distinct SVG-clusters. (C) Spatial domains detected by cSVG and SpatialPCA, with respective ARI scores of 0.48 and 0.44.

of domains.

The dataset includes a rough annotation (see 4.5A) highlighting the tumor (marked in red) and non-tumor (marked in yellow) regions of interest. Figures 4.5B and 4.5C display the predicted domains by SpatialPCA and cSVG frameworks, respectively.

As the precise annotation labels for each spot are unavailable, we cannot compute scores like ARI to compare spatial domain detection accuracy between the two methods. However, through visual inspection, the domains detected by cSVG appear more accurate, effectively capturing the most important regions. Tumor-containing regions identified by cSVG are notably smaller and more accurate compared to those identified by SpatialPCA.

The cSVG algorithm identified three primary clusters of spatially variable genes (comprising 3931, 2751, and 1781 genes, respectively) in the pancreatic cancer dataset, each providing significant biological insights. Supplementary Figure F.8 demonstrates that genes in Cluster 1 are predominantly overexpressed in non-tumor regions, while genes in Clusters 2 and 3 are overexpressed around tumor regions. Consequently, pathway enrichment analysis reveals that genes in Clusters 2 and 3 are associated with cancer-related pathways, whereas genes in Cluster 1 are not. Notably, Cluster 3 genes are enriched with several T cell-related pathways, a feature not observed in Cluster 2. This suggests that Cluster 3 gene expressions exhibit immune cells spatially located around cancer regions, offering intriguing insights into the biology of cancer.

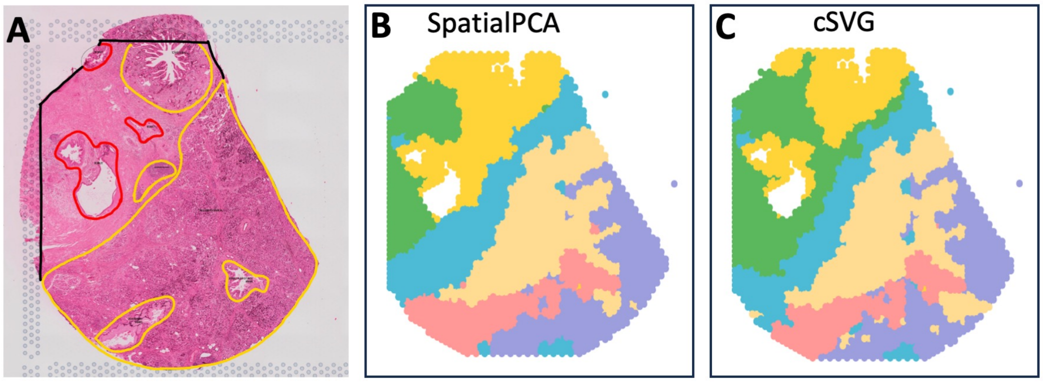


Figure 4.5 Analysis of the Pancreatic cancer data: (A) Rough annotation of the tissue depicting different important regions. Tumor regions are highlighted in red, while non-tumor yet significant regions are marked in yellow. (B) Spatial domains detected by SpatialPCA, utilizing the calculation of spatial PCs and employing the Leiden algorithm for clustering without presupposing the number of clusters. (C) Spatial domain detection by cSVG, involving the aggregation of SVG-cluster specific spatial PCs and utilizing the Leiden algorithm for clustering without presupposing the number of clusters.

4.5 Discussion

In recent years, the exploration and analysis of spatial data have reached unprecedented heights, offering diverse insights into biological systems. Central to this endeavor is the identification of SVGs, which serve as pivotal components in understanding tissue organization and function. However, merely detecting SVGs does not inherently yield substantial biological insights. Rather, their significance lies in their dual role: 1) SVGs are used for spatial domain detection and 2) Some SVGs serve as markers for specific cell types. Traditional approaches often struggle to achieve precise spatial domain detection, and discerning cell type-specific SVGs amidst the data noise poses a formidable task. Our proposed framework, cSVG, addresses these challenges to improve spatial domain detection without requiring cell type information.

By implementing cSVG, we effectively detect SVG clusters that can be interpreted as clusters of cell type SVGs. Remarkably, this identification is accomplished without the need for complex cell type deconvolution techniques, streamlining the analysis process while providing biologically meaningful insights. Through extensive real data analysis and simulation studies, we have demonstrated the efficacy of cSVG in enhancing spatial domain detection accuracy, validating its utility in spatial transcriptomic analysis.

The implementation of cSVG involves two sequential steps, with the initial phase focusing on SVG detection. This step can be performed using any existing SVG detection technique and the alteration can be done without changing any code for cSVG. The detailed step-by-step code and results are provided on our GitHub repository https://github.com/wangjr03/cSVG. Looking ahead, there are ample opportunities to refine and expand upon the cSVG framework. Future enhancements may involve simplifying and scaling up the SVG detection and cluster detection methods to accommodate larger and more complex datasets. Additionally, continued refinement of the framework will enable researchers to extract deeper insights from spatial transcriptomic data, advancing our understanding of tissue biology and disease mechanisms.

In conclusion, the use of SVG clusters generated by cSVG represents a crucial advancement in spatial transcriptomic analysis. As we continue to refine and evolve this methodology, it is poised to become an indispensable tool for dissecting the spatial complexity of biological systems and unraveling the intricate interplay between genes, cells, and tissues.

Data and code availability

All relevant codes for reproducing each step of the real data analysis and simulation study results are available on our GitHub repository: https://github.com/wangjr03/cSVG. The publicly accessible datasets and their sources are provided in the data folder. Please note that the pancreatic cancer data used in this study was received from our collaborators Dr. Nina Steele and Dr. Brian Theisen from the Henry Ford Health System. The Institutional and Review Board approval is maintained for 16150 at Henry Ford Hospital for The Translational and Clinical Research Center Biorepository. The data has not been published and is not publicly available at this time. We received a de-identified dataset for the analysis described in this chapter.

4.6 Supplementary materials

4.6.1 Score test for SVG detection

The Gaussian process (GP) regression model which models the normalized gene expression y for a given gene using the following multivariate normal model:

$$p(y|\mu, \sigma_s^2, \delta, K) \sim N(y|X\beta, \sigma_s^2 K + \delta I),$$

where the covariance term is decomposed into a spatial and a non-spatial part, where δI represents the non-spatial part and $\sigma_s^2 K$ is the spatial covariance matrix, whose $(i, j)^{th}$ element in the kernel matrix K denotes the spatial similarity between the ith and jth spot calculated based on the corresponding coordinates s_i and s_j. The choice of the kernel function plays a very important role in detecting the spatial correlation present in the gene expressions. $X^{N\times k}$ represents the covariate matrix, while $\beta^{k\times 1}$ denotes the array of corresponding coefficients.

As we mentioned in the main text, testing if a gene is a SVG is equivalent to testing $H_0 : \sigma_s^2 = 0$. The null hypothesis $H_0 : \sigma_s^2$ =0 can be tested using the variance-component score test which is the locally most powerful test[141]. The variance-covariance score statistic is:

$$Q = (y - X\hat{\beta})^T K(y - X\hat{\beta})$$

where $\hat{\beta}$ is the MLEs under the null model. Under the null hypothesis, the score statistic Q follows a mixture of chi-square distributions [142], which can be closely approximated with the computationally efficient Davies' method[143]. More details about the test is provided in the appendix B.

In this chapter, we utilize part of the code provided along with the SKAT paper [142] which uses the same score test for the purpose of rare-variant association testing in genetic data.

Kernel functions: A kernel function is defined as a function $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, where the kernel matrix $K = (k_{i,i'})_{i,i'=1}^{n}$ is symmetric and positive semidefinite with $k_{i,i'} = k(s_i, s_{i'})$. In this setting, $k(s_i, s_{i'})$ is a measure of similarity between the ith and the i'th subject. There are a variety of kernel functions to choose from, and the most simple one is the Linear kernel. The other useful kernels are Polynomial kernel, the Gaussian kernel and the cosine kernel. The functional forms of these kernels are summarized below:

- Linear kernel: $K(s_i, s_{i'}) = s_i^T s_{i'}$
- Polynomial kernel: $K(s_i, s_{i'}) = (s_i^T s_{i'} + c)^d$, where c,d are the free parameters.
- Gaussian kernel: $K(s_i, s_{i'}) = exp\{-\|s_i - s_{i'}\|^2/l\}$, where $\|s_i - s_{i'}\|^2 = \sum_{j=1}^{p}(s_{ij} - s_{i'j})^2$ is the Euclidean distance, l is a length scale parameter.
- Cosine kernel: $K(s_i, s_{i'}) = cos(2\pi \frac{\|s_i - s_{i'}\|^2}{\phi})$, where ϕ is the periodicity parameter.

Choices of kernel functions: We must define the kernel function in order to proceed with the hypothesis testing. As it is unknown which kernel will be best for the test, we employ the score test to evaluate the null hypothesis across various kernel functions with distinct kernel parameters. Gaussian and cosine kernels are typically effective in capturing spatial patterns. Following the method outlined in the SPARK paper [77], we compute five different length scale parameter values for the Gaussian Kernel and five different periodic parameter values for the cosine kernel. We conduct the test across ten different kernels and aggregate the resulting p-values using the Cauchy combination rule[144].

4.6.2 cSVG algorithm

Our model is built upon the Gaussian process model. Thus, we require normalized count matrix data with m rows (genes) and N columns (spots). We also need the spatial location matrix L with N rows, 2 columns (X and Y coordinates of spots).

Step 1: Detect SVGs based on model 4.1 defined in the main text. Suppose there are m_1 SVGs and denote the SVG list as S_y with $|S_y| = m_1$ where $|\cdot|$ denotes the cardinality of a set.

Step 2: Start with the subset matrix denoted by $M_{SVG}^{m_1 \times N}$.

Repeat for $j = 1, \cdots, m_1$:

1. For the jth gene in S_y, find genes correlated with it using methods such as (1) SIS[137], (2) marginal correlation test (+ve correlation), (3) Elastic net[134][135], (4) SIS+Enet, or other methods. Denote the correlated gene list as S_j.

2. If $|S_j| = 0$, then the jth gene has an unique spatial pattern.

If $|S_j| <= 3$, then fit all genes in S_j as the covariates in model (2) in the main text.

If $|S_j| > 3$, then get the k_j PCs of genes in S_j and fit them as covariates in model (2) in the main text.

3.. Using the model 4.2 in the main text, conduct a score test to compute the p-value under 10 different kernels following the SPARK idea, then integrate these 10 p-values using the Cauchy combination rule[144],[145] to get the final p-value. The output includes: a) For each SVG j, a list of correlated genes in S_j; and b) A list of unique SVGs as defined in 2.

Step 3:

1). Based on the output in step 2, a weighted graph structure is created where each SVG is a node. Each SVG j in the output list in 3.a) has a common edge with all its dependent genes specified in list S_j in the graph.

2). Clusters are determined from the weighted graph structure in 1) using the Leiden community detection algorithm[138] (see Appendix D).

3). The unique genes in output list 3.b) not connected with other nodes in the graph structure are allocated to the singleton set.

4.6.3 Multiplicity correction

At each stage of cSVG, our model is applied to each of the m genes. To control the false discovery rate (FDR), multiplicity correction is required. We employ the Benjamini–Yekutieli (BY) procedure, known for its effectiveness under arbitrary correlation conditions, to obtain adjusted p-values to claim significant genes [146].

4.6.4 Analysis of human breast cancer dataset

We analyzed another spatial transcriptomics dataset of human breast cancer. We obtained the data from the SPARK[77] GitHub repository at https://github.com/xzhoulab/SPARK-Analysis (specifically the 'Layer2_BC_count_matrix-1.tsv' file). The dataset can also be downloaded from Spatial Transcriptomics Research (http://www.spatialtranscriptomicsresearch.org). This dataset contains 14,789 genes measured across 251 spots. Our preprocessing involved filtering out genes with expression levels below 10% across the array spots and retaining spots with a total read count > 10. Following these criteria, our analysis focused on a final set of 5,262 genes observed across 250 spots within the breast cancer dataset.

Using our framework cSVG, we identified 724 spatially relevant genes in the initial step. In comparison, SPARK[77] detected 290 genes, SPARK-G detected 244 genes, nnSVG[78] detected 592 genes, and SPARK-X[88] detected 901 genes. The 724 genes were subsequently classified into three main clusters(containing 369, 320, and 14 genes respectively) in the second step of cSVG framework, along with a few unique singleton genes. The representative genes from each cluster are visualized in Figure 4.6. Genes in cluster 1 exhibited higher expression levels in the lower tissue region, while cluster 2 genes were overexpressed in the middle tissue region, and cluster 3 genes highlighted the upper tissue region. Additionally, the unique genes showcased distinct expression patterns in the third row. This figure confirms the superior performance of our cSVG method.

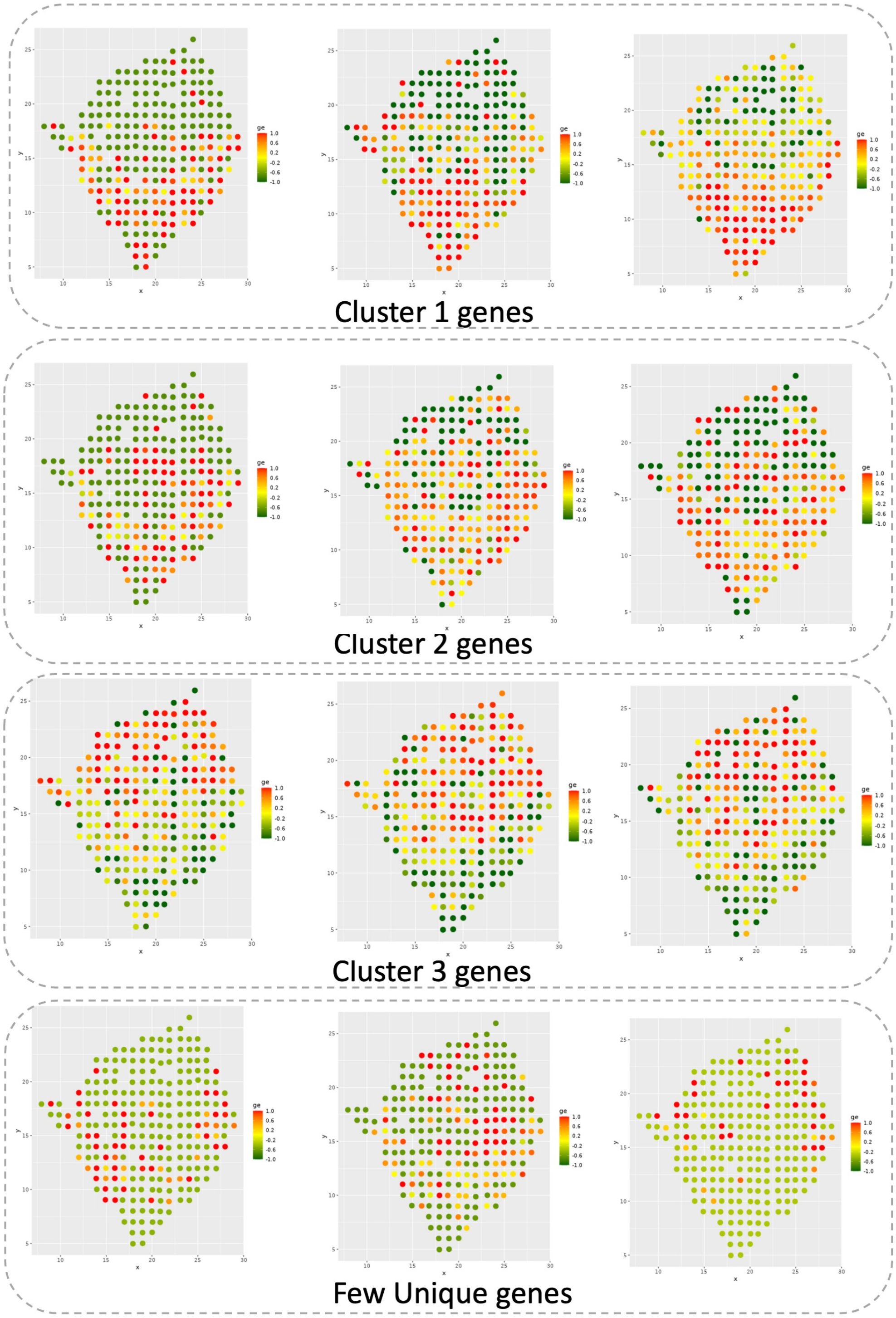


Figure 4.6 The representative spatial genes in the human breast cancer dataset from SVG clusters detected by cSVG.

CHAPTER 5

3D-TF-IMPUTE: A COMPUTATIONAL MODEL TO IMPUTE TF BINDING SITES

5.1 Introduction

Typically, the initial phase in understanding gene regulation involves pinpointing the in vivo binding sites across the genome for various transcription factors (TFs). Unfortunately, ChIP-seq data of TF binding are only available for a small subset of TFs in limited cellular contexts. For instance, within the human genome, despite hundreds of recognized TFs, ChIP-seq data is available for fewer than 10 TFs in most cell types cataloged in ENCODE [147]. This limitation is exacerbated in other organisms, such as plants [148].

Hence, there arises an urgent demand for innovative algorithms to predict TF binding sites on a genome-wide scale across diverse cell types. While several methods have been proposed [149],[150],[151] to address this TF binding imputation challenge, The most recent algorithm, Avocado [152], proposes to leverage the information from 6,870 epigenomics and TF binding data together and do imputation for the human genome. Although it can generate a rich model, it is limited for biological applications since it is usually impossible for biologists to gather the huge amounts of data first, especially for studies in other species.

As chromatin accessibility data is becoming widely available [153] , other algorithms, with 'J-Team' and 'Yuanfang Guan' as the winning methods [149],[150] , all share similar ideas: i.e. combine TF binding motif information with in vivo chromatin accessibility to impute TF binding for specific genomic locations, based on models learned from another cell type. To improve the performance, co-occurring motifs in local flanking sequences, which biologically represent the potential co-factors, are integrated into the models as important features.

The fundamental restriction of these methods (e.g. 'J-Team') is that only local co-occurring motif information along the 1D genome (< 1KB) is used. Since TF bindings happen in 3D chromatin, we need to explore the co-occurring motif patterns of potential co-factors and the TF itself in 3D neighborhoods. This idea has big potential to improve cell type specific TF imputation because: 1) It align with the real biology better. TFs and their co-factors usually form complexes and co-regulate genes in 3D chromatin space, i.e. the well-known transcription factories [154], [155]. 2) Preliminary observation: We grouped 3D interactions (in GM12878 cells) by genes and looked at TF binding frequency in 3D neighborhood for two TFs: CTCF and it's known co-factor YY1. A substantial number of 3D neighborhoods show binding for both the TFs (See 5.1A), suggesting the TF co-factors patterns in 3D space are critical. 3)Leveraging 3D chromatin information can significantly expand useful features from local flanking sequences to long-range interacting sequences. The percentages of binding sites for TF cofactors CTCF and YY1 are much higher in 3D neighborhood compared to in 1D flanking region (See 5.1B). 4) Based on 3D interactions to genes, we are enabled to explore specific TF co-occurring information in different gene groups.

There are many challenges in this work. To efficiently leverage the 3D chromatin information, we need to overcome a series of computational hurdles: 1) Co-factors of TFs in 3D interacting neighborhood are largely unknown ; 2) Combinations of TFs and corresponding co-factors may vary for different gene groups; (See 5.1C) 3) Distributions of chromatin accessibility (e.g. DNase-seq) are unknown in 3D interacting regions vs. non-interacting regions.

We propose 3D-TF-IMPUTE, a novel probabilistic model designed to concurrently infer cell type-specific TF binding and uncover all pertinent unknown parameters, as depicted in Figure 5.1C,D, employing an unsupervised approach. The algorithm takes as input: 1)Chromatin accessibility data, 2)Motif hit information for the entire genome across various TFs, and 3)Chromatin interaction data. The first key feature of the model is to explore 3D chromatin neighborhoods, instead of 1D flanking regions. The second key feature of the model is to explore gene group specific co-factor association patterns, so that we can borrow information across different genes. Notably, TF-ChIP-Seq data is omitted during model computation and only utilized post hoc to validate the model's outcomes.

5.2 Input data pre-processing

3D chromatin interaction data is now widely available for different cell types and species [153],[156],[157]. Both Capture Hi-C data and ChIA-PET data based on general factors (e.g.

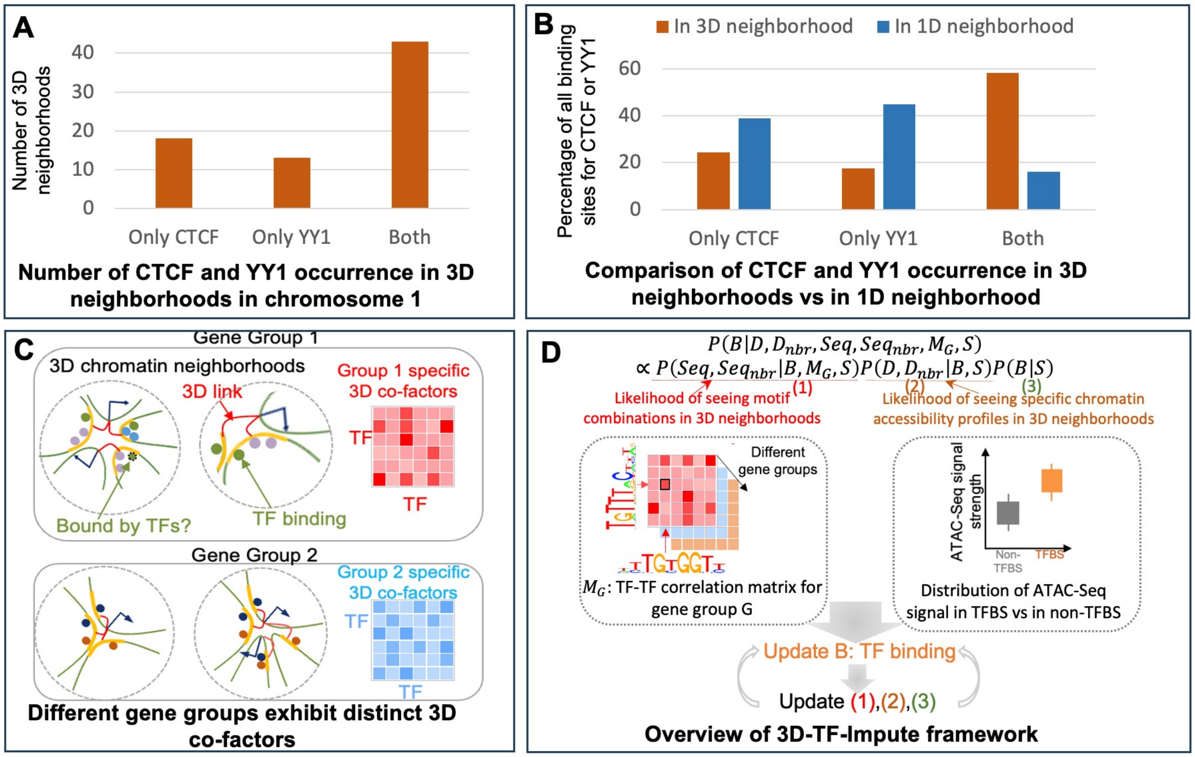


Figure 5.1 Motivation and Overview of 3D-TF-IMPUTE. A) Often TF pairs exhibit tendency to bind simultaneously in 3D neighborhood. Barplot shows frequency of CTCF and YY1 binding sites in chromosome 1 within 3D neighborhood regions, comparing individual binding occurrences versus simultaneous binding. B) Pair-wise binding tendency of CTCF, YY1 in 3D neighborhoods vs. in 1D neighborhood regions. C) Schematic of TF binding imputation. 3D neighborhoods are defined for each genes (doted circle). Genes are grouped based on similar 3D co-factor profile (rectangular line). D) Modeling and inference of TF binding by borrowing information from 3D neighbors and gene groups. Subscript nbr denotes features of 3D neighborhoods. B: binary label represents TF binding. D: continuous ATAC-Seq signal at the location. Seq: binary sequence featured by matching with known TF motifs. S: 3D chromatin structure. M_G: gene group-specific 3D co-factor profile. Posterior probability of B is calculated based on observed ATAC-Seq signal and sequence feature of local and 3D neighborhoods given 3D chromatin structure and group-wise co-factor profile. After updating B, each probability component is recalculated based on updated B until convergence.

p300, Pol II or H3K4me2) can be used as inputs to the model since they have better resolution. For Hi-C data, we use ATAC-Seq peaks to significantly increase the resolution and cell-type specificity of chromatin contacts, as suggested by previous studies [158].

To enhance scalability, we partitioned the entire genome into consecutive 100-base pair (BP) bins and assessed the likelihood of a binding site occurrence within each bin for every TF. If a bin is predicted to contain a binding site, the specific binding site location can be identified by the presence of an 8mer MOTIF within that bin.

5.3 Method Overview

Here we describe our advanced novel sampling framework 3D-TF-IMPUTE, which uses ATAC-Seq, Hi-C, and TF binding motif information datasets as input and predicts the TF binding sites for our TF of interest. First, the whole chromosome is divided into consecutive 100 Base Pairs length segments or genomic bins and 3D-TF-IMPUTE predicts which of these bins contains the potential binding sites for the TF of interest. It narrows down the search to active bins, which serve as the potential region for TF binding sites. Each of these active bins create a 3D neighborhood, which is defined as genomic regions in chromatin interacting hubs(indicated by 3D chromatin interaction data) where at least one gene promoter region is involved. We refine the selection of active bins further by identifying those that overlap with any TF MOTIF. These bins, characterized by such overlap, emerge as highly active and promising candidates for TF binding sites.

3D-TF-IMPUTE starts with this subset of active bins and calculates the probability $P(B = 1 \mid D, D_{nbr}, Seq, Seq_{nbr}, M_G, S)$ for all the bins in the active set, where B is the latent variable indicating whether the genomic bin location belongs to a TF binding site or not (B=1 or 0), D represents the ATAC-seq signal at the location, D_{nbr} represents the ATAC-seq signals in the 3D neighborhood corresponding to the location, Seq represents the occurrences of motif hits for all available TFs at the location, Seq_{nbr} represents the same in the 3D neighborhood, M_G represents the TF co-occurrence in 3D neighborhoods for a specific gene-group G, and S represents the 3D chromatin structure. A higher value of $P(B = 1 \mid D, D_{nbr}, Seq, Seq_{nbr}, M_G, S)$ indicates higher chances of the TF binding at the active bin of interest. Directly this posterior probability is complex, but it can be decomposed into several parts: $P(B \mid D, D_{nbr}, Seq, Seq_{nbr}, M_G, S) \propto P(Seq, Seq_{nbr} \mid B, M_G, S)P(D, D_{nbr} \mid B, S)P(B \mid S)$. The first probability term infers the likelihood of the event based on 3D neighborhood TFs interdependence, the second probability term incorporates the information based on chromatin accecibility and the third probability term simply includes the odds of having a binding site.

The first probability term can be further decomposed into $P(Seq, Seq_{nbr} \mid B, M_G, S)$= $P(Seq_{nbr} \mid Seq, B = 1, M_G, S)$ $P(Seq \mid B = 1, M_G, S) = P(Seq_{nbr} \mid Seq, B = 1, M_G, S)$ $P(Seq_{TF_1} \mid B = 1, M_G, S)$ $P(Seq_{TF_2}|B = 1, M_G, S) \cdots P(Seq_{TF_t} \mid B = 1, M_G, S)$, where $Seq^{t\times1}$ array contains information about t known TFs MOTIF binding at the location, $Seq^{t\times1} = [Seq_{TF_1}, Seq_{TF_2}, \cdots, Seq_{TF_t}]$. The second "=" holds with the assumption that given the binding status at the location and the TF-TF co-occurrence information provided by M_G, $Seq_{TF_i}, i = 1, 2, ..t$ are independently distributed. While predicting TF binding sites for TF_j, 3D-TF-IMPUTE algorithm estimates $\prod_{i=1 i\neq j}^{t} P(Seq_{TF_i}|B = 1, M_G, S)$ by $\frac{\sum_{i=1}^{t} Seq_{TF_i} \times M_{G_{ji}}}{\sum_{i=1^t} M_{G_{ji}}}$, where $M_{G_{ji}}$ is the i^{th} component of the row in M_G corresponding to the TF of interest TF_j.

The second probability term is calculated based on a kernel density estimation(KDE) of the joint distribution of D, D_{nbr}. The third probability term is calculated based on the proportion of active bins with predicted binding sites.

In the iterative process of 3D-TF-IMPUTE, the B values are initialized for all active bins using TF MOTIF data. Subsequently, utilizing these initialized B values, the probability terms 1, 2, and 3 are computed. Based on the resulting final probability $P(B \mid D, D_{nbr}, Seq, Seq_{nbr}, M_G, S)$ values, the B values for the active bins are sampled, initiating the next iteration. This iterative cycle continues until the probability values converge across all active bins. The schematic overview of the sampling framework is provided in figure 5.1D and the detailed algorithm is provided here:

```
Set ch=chromosome1, TF=TF1,num$_{iter}$=1000
Prepare ch specific ATAC-seq data
Prepare ch specific MOTIF data.
Prepare ch specific 3D neighbourhood data
Prepare TF specific MOTIF data
iter 1:num$_{iter}$:
    Initialize latent variable values b1 specific to the TF for all active bins.
    Prepare $P_2$ function input based on ATAC and initialized latent variable values.
    b1 in ActiveBinList:
        calculate $P_1(1), P_1(0)$
        calculate $P_2(1), P_2(0)$
        calculate $P_3(1), P_3(0)$
        calculate $P(B = 1|D, D_{nbr}, Seq, Seq_{nbr}, s) = \frac{P_1(1)*P_2(1)*P_3(1)}{P_1(1)*P_2(1)*P_3(1)+P_1(0)*P_2(0)*P_3(0)}$
        Sample B based on $P(B = 1|D, D_{nbr}, Seq, Seq_{nbr}, s)$, update b1
```

where, $P_1(1) = P(Seq, Seq_{nbr}|B = 1, M_G, S)$, $\quad P_1(0) = P(Seq, Seq_{nbr}|B = 0, M_G, S)$

$P_2(1) = P(D, D_{nbr}|B = 1, S)$, $\quad P_2(0) = P(D, D_{nbr}|B = 0, S)$

$P_3(1) = P(B = 1|S)$, $\quad P_3(0) = P(B = 0|S)$

5.4 Results

In cell types where TF-ChIP-seq data is available, the data is often noisy or incomplete. Since there is no ground truth available, we evaluate the performance of our predictions by examining the number of overlaps with TF ChIP-seq binding sites. Specifically, we compare the predicted binding sites of CTCF, YY1, and RUNX3 in GM12878 cell type with the ChIP-Seq data. Figure 5.2 showcases real examples on the UCSC genome browser, where black tracks represent TF ChIP-Seq peaks and predicted binding sites are marked in red. These examples illustrate the degree of overlap between predicted binding sites and TF ChIP-Seq peaks.

Figure 5.3 presents the ROC and Precision-Recall curves for these three key TFs: CTCF, YY1, and RUNX3. The high quality of TF MOTIF data for CTCF leads to highly accurate final predictions as evident by the high area under the ROC and PR curve. On the other hand, the MOTIF data quality for YY1 and RUNX3 is extremely low. Nonetheless, their predictions remain reasonable.

Overall, the results demonstrate the effectiveness of the 3D-TF-IMPUTE algorithm in accurately predicting TF binding sites.

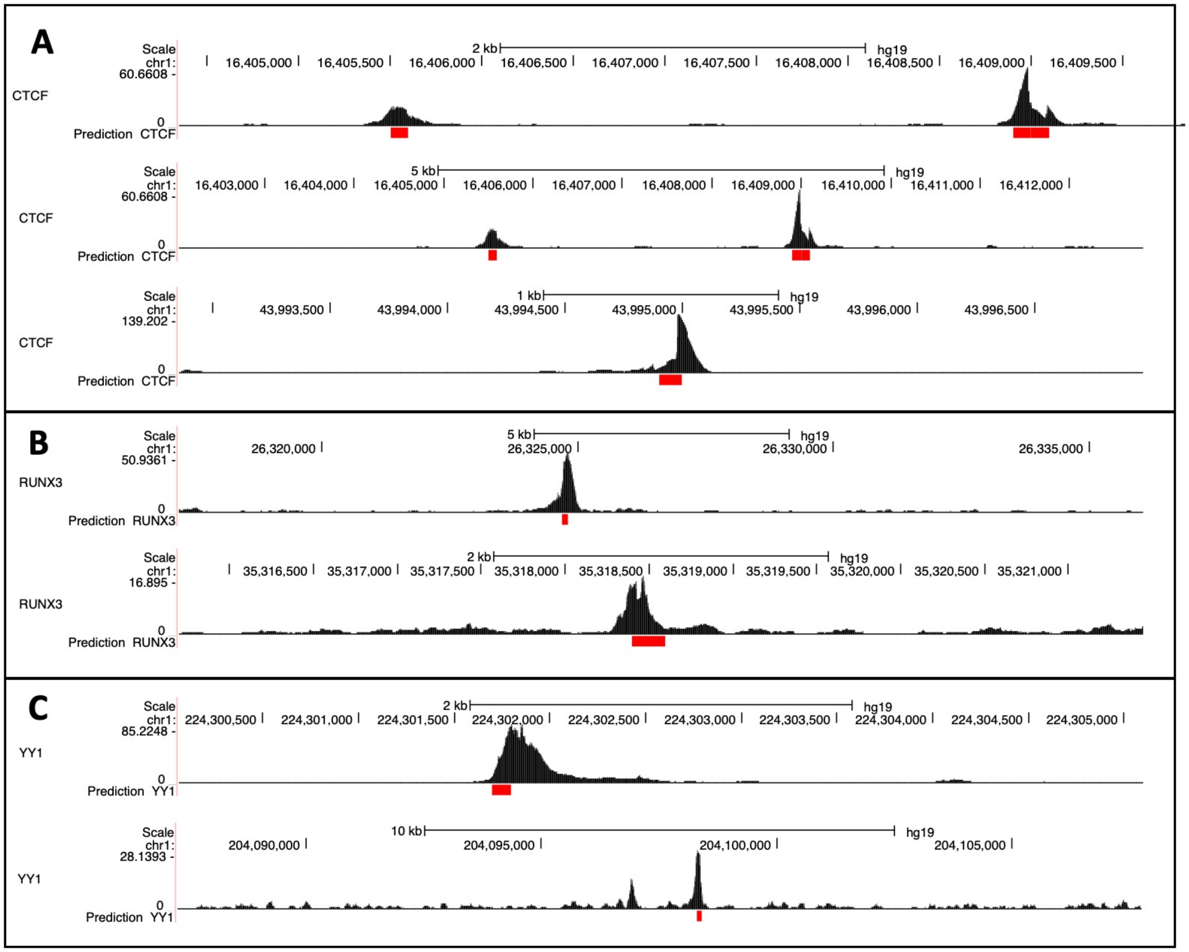


Figure 5.2 Examples of predicted TF binding sites by 3D-TF-IMPUTE overlapped with significant TF peaks shown in the genome browser. The tracks in black correspond to the transcription factors A) CTCF, B) RUNX3, C) YY1. The predictions by 3D-TF-IMPUTE are highlighted in red.

5.5 Discussion

In conclusion, while existing computational methods for TF binding have predominantly relied on supervised setups, our proposed approach marks a significant departure by introducing an unsupervised probabilistic model. Unlike traditional methods, which often encounter challenges when training data from one cell type fails to accurately represent TF binding in others or data quality is low, our method circumvents such issues.

Moreover, in contrast to existing methods that suffer from lengthy execution times, our approach

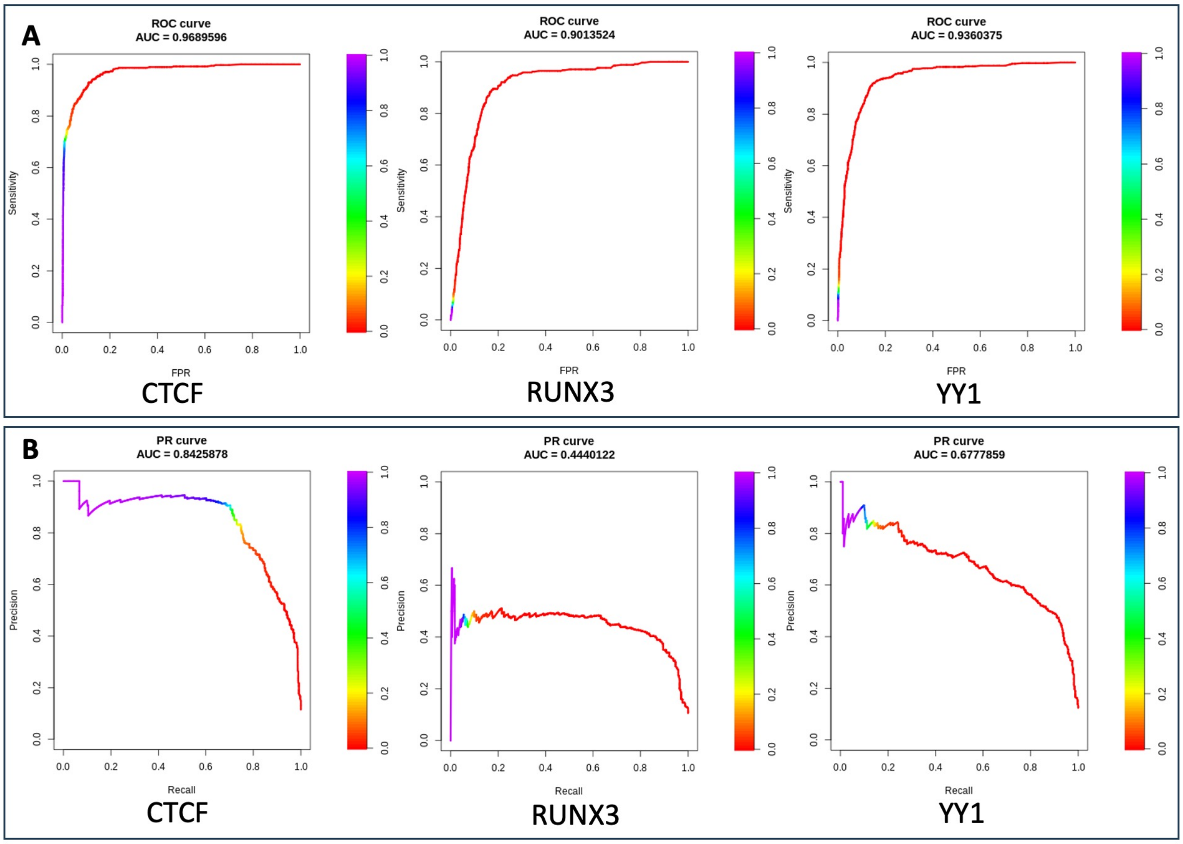


Figure 5.3 Evaluation of 3D-TF-IMPUTE performance. A) ROC curve demonstrating the prediction accuracy for significant TFs. B) Precision-Recall curve demonstrating the prediction accuracy of the same TFs.

showcases remarkable efficiency by running in parallel, thus substantially decreasing preprocessing and computation time.

Remarkably, 3D-TF-IMPUTE attains a notably low false discovery rate, particularly in cases where input data quality is high. Even in instances of subpar motif data quality, 3D-TF-IMPUTE adeptly utilizes information from co-factors to predict high-quality TF binding sites, thereby augmenting prediction reliability, as illustrated by the example of YY1(see figure 5.3).

While our approach represents a pioneering advancement, there remains room for enhancement. Addressing the impact of low-quality motif data on prediction accuracy could be a promising avenue for future refinement. Incorporating mechanisms to update motif information based on co-factor interactions may prove instrumental in bolstering prediction power.

Importantly, our work underscores the pivotal role of three-dimensional chromatin neighborhoods in enhancing TF binding prediction accuracy. Looking ahead, this concept holds significant promise for inspiring further methodological developments, with potential applications across diverse research endeavors focused on predicting TF binding sites.

5.6 Data and Code availability

The code to run this model is available at GitHub: https://github.com/wangjr03/3DTFImpute. The input dataset ATAC-Seq and ChIP-Seq for GM12878 cell type is downloaded from ENCODE [147]. The Motif data downloaded from [159] https://www.internationalgenome.org and the Hi-C data from [160] https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE63525.

CHAPTER 6

CONCLUSION AND FUTURE WORK

The cSVG model introduced in Chapter 4 represents a significant advancement in spatial domain detection, substantially improving accuracy compared to existing methods. However, there are opportunities to further enhance its performance.

One area for improvement is addressing the challenge of gene separability observed in t-SNE plots of LIBD data samples 5, 6, 7, and 8 (see F.6). In these instances, genes may form large clusters with spatial patterns overlapping multiple distinct clusters. This suggests the presence of genes with mixed spatial features, requiring a more nuanced approach to modeling spatial variability. The integration of composite kernel functions, as facilitated by the BayesKAT model, could offer a more efficient means of capturing the spatial complexity of these genes.

Furthermore, enhancing the scalability of cSVG cluster identification represents a significant area for improvement in future research. While we have demonstrated the utility of SVG clusters in improving downstream analysis, the current approach may be resource-intensive for larger datasets. Leveraging the BayesKAT model to identify gene clusters based on selected composite kernel weights offers a promising solution to this challenge, streamlining the process of SVG cluster identification and enabling more scalable analysis.

These future directions hold the potential to address key questions in the field of spatial transcriptomics and further enhance our understanding of spatial gene expression patterns. By refining our methodologies and leveraging advanced computational techniques, we aim to unlock new insights into the spatial organization of gene expression and its implications for biological processes.

The primary objective behind developing the BayesKAT algorithm was to assess the joint association of a single group of SNPs or gene expressions with the phenotype of interest. While the effectiveness and efficiency of BayesKAT in prioritizing and ranking significant SNP groups were demonstrated through pathway-wise or gene-wise analyses in the chapter, from a statistical perspective, determining a threshold that effectively controls false discovery rates poses challenges. Additionally, the current version of BayesKAT relies on approximations, which may not perform optimally with small sample sizes. As a response, I propose an updated version of the BayesKAT model which improves the model structure. Also, I intend to enhance its speed and performance by implementing variational inference techniques[161].

As defined in 2.3 in chapter 2, a slight reparametrization of the BayesKAT model has the data distribution:

$$y|\theta \sim N(X\beta, \tau K_c + \sigma^2 I)), \tag{6.1}$$

where the composite kernel $K_c = \sum_{i=1}^{3} \rho_i K_i$. Now we use one indicator variable δ to indicate whether or not τ is 0. That means, if $\delta = 0$, it indicates that $\tau = 0$, i.e there is no association between the group of SNPs and the phenotype. We treat the indicator variable δ as Bernoulli random trials with success rate $p(\delta = 1) = \pi$. τ follows an exponential(λ_s) prior distribution when $\delta = 1$ and equals to 0 when $\delta = 0$. This is similar to the spike and slab prior [162], where the slab distribution is generally normal as it is normally used for variable selection. We might set specific values for the hyperparameters of the model π, λ_s or different prior distributions can be assumed on them. The posterior distribution for all the parameters can be estimated using the variational inference technique[161] assuming a mean-field variational family and finally, the posterior inclusion probability (PIP) can be defined as: $PIP = p(\delta = 1 \mid X, y)$. Based on the PIP value it would be possible to infer the strength of the association. Furthermore, in the case of multiple testing(same phenotype, different sets of SNPs), the problem can be written as a variable selection problem and solved following the sparse Bayesian variable selection technique outlined in literature [163], [164].

BIBLIOGRAPHY

[1] Lars Bertram and Rudolph E. Tanzi. Thirty years of alzheimer's disease genetics: the implications of systematic meta-analyses. *Nature Reviews Neuroscience*, 9(10):768–778, Oct 2008.

[2] Timothy J. Vyse and John A. Todd. Genetic analysis of autoimmune disease. *Cell*, 85(3):311–318, May 1996.

[3] Emil Uffelmann, Qin Qin Huang, Nchangwi Syntia Munung, Jantina de Vries, Yukinori Okada, Alicia R. Martin, Hilary C. Martin, Tuuli Lappalainen, and Danielle Posthuma. Genome-wide association studies. *Nature Reviews Methods Primers*, 1(1):59, Aug 2021.

[4] Evan A Boyle, Yang I Li, and Jonathan K Pritchard. An expanded view of complex traits: from polygenic to omnigenic. *Cell*, 169(7):1177–1186, 2017.

[5] Eric S Lander. Initial impact of the sequencing of the human genome. *Nature*, 470(7333):187–197, 2011.

[6] Andre Franke, Dermot PB McGovern, Jeffrey C Barrett, Kai Wang, Graham L Radford-Smith, Tariq Ahmad, Charlie W Lees, Tobias Balschun, James Lee, Rebecca Roberts, et al. Genome-wide meta-analysis increases to 71 the number of confirmed crohn's disease susceptibility loci. *Nature genetics*, 42(12):1118–1125, 2010.

[7] Benjamin F Voight, Laura J Scott, Valgerdur Steinthorsdottir, Andrew P Morris, Christian Dina, Ryan P Welch, Eleftheria Zeggini, Cornelia Huth, Yurii S Aulchenko, Gudmar Thorleifsson, et al. Twelve type 2 diabetes susceptibility loci identified through large-scale association analysis. *Nature genetics*, 42(7):579–589, 2010.

[8] Laura I Furlong. Human diseases through the lens of network biology. *Trends Genet*, 29(3):150–159, December 2012.

[9] Aravinda Chakravarti and Tychele N Turner. Revealing rate-limiting steps in complex disease biology: The crucial importance of studying rare, extreme-phenotype families. *Bioessays*, 38(6):578–586, April 2016.

[10] Lydia Coulter Kwee, Dawei Liu, Xihong Lin, Debashis Ghosh, and Michael P Epstein. A powerful and flexible multilocus association test for quantitative traits. *The American Journal of Human Genetics*, 82(2):386–397, 2008.

[11] Dawei Liu, Xihong Lin, and Debashis Ghosh. Semiparametric regression of multidimensional genetic pathway data: Least-squares kernel machines and linear mixed models. *Biometrics*, 63(4):1079–1088, 2007.

[12] Michael C. Wu, Seunggeun Lee, Tianxi Cai, Yun Li, Michael Boehnke, and Xihong Lin. Rare-variant association testing for sequencing data with the sequence kernel association test. *The American Journal of Human Genetics*, 89(1):82–93, 2011.

[13] Shaoyu Li and Yuehua Cui. Gene-centric gene–gene interaction: A model-based kernel machine method. *The Annals of Applied Statistics*, 6(3):1134 – 1161, 2012.

[14] Dawei Liu, Debashis Ghosh, and Xihong Lin. Estimation and testing for the effect of a genetic pathway on a disease outcome using logistic kernel machine regression via logistic mixed models. *BMC bioinformatics*, 9:1–11, 2008.

[15] Rachel Marceau, Wenbin Lu, Shannon Holloway, Michèle M Sale, Bradford B Worrall, Stephen R Williams, Fang-Chi Hsu, and Jung-Ying Tzeng. A fast multiple-kernel method with applications to detect gene-environment interaction. *Genetic epidemiology*, 39(6):456–468, 2015.

[16] Michael C Wu, Peter Kraft, Michael P Epstein, Deanne M Taylor, Stephen J Chanock, David J Hunter, and Xihong Lin. Powerful snp-set analysis for case-control genome-wide association studies. *The American Journal of Human Genetics*, 86(6):929–942, 2010.

[17] Seunggeun Lee, Zhangchen Zhao, with contributions from Larisa Miropolsky, and Michael Wu. *SKAT: SNP-Set (Sequence) Kernel Association Test*, 2023. R package version 2.2.5.

[18] Michael C Wu, Arnab Maity, Seunggeun Lee, Elizabeth M Simmons, Quaker E Harmon, Xinyi Lin, Stephanie M Engel, Jeffrey J Molldrem, and Paul M Armistead. Kernel machine snp-set testing under multiple candidate kernels. *Genetic epidemiology*, 37(3):267–275, 2013.

[19] Jennifer Wessel and Nicholas J Schork. Generalized genomic distance–based regression methodology for multilocus association analysis. *The American Journal of Human Genetics*, 79(5):792–806, 2006.

[20] Xinyi Lin, Tianxi Cai, Michael C Wu, Qian Zhou, Geoffrey Liu, David C Christiani, and Xihong Lin. Kernel machine snp-set analysis for censored survival outcomes in genome-wide association studies. *Genetic epidemiology*, 35(7):620–631, 2011.

[21] Tao He, Shaoyu Li, Ping-Shou Zhong, and Yuehua Cui. An optimal kernel-based u-statistic method for quantitative gene-set association analysis. *Genetic epidemiology*, 43(2):137–149, 2019.

[22] the ADNI team. *ADNIMERGE: Alzheimer's Disease Neuroimaging Initiative*, 2023. R package version 0.0.1.

[23] Minoru Kanehisa and Susumu Goto. Kegg: kyoto encyclopedia of genes and genomes. *Nucleic acids research*, 28(1):27–30, 2000.

[24] Minoru Kanehisa. Toward understanding the origin and evolution of cellular organisms. *Protein Science*, 28(11):1947–1951, 2019.

[25] Minoru Kanehisa, Miho Furumichi, Yoko Sato, Masayuki Kawashima, and Mari Ishiguro-Watanabe. Kegg for taxonomy-based analysis of pathways and genomes. *Nucleic Acids Research*, 51(D1):D587–D592, 2023.

[26] Robert E Kass and Adrian E Raftery. Bayes factors. *Journal of the american statistical association*, 90(430):773–795, 1995.

[27] Nicholas Metropolis, Arianna W. Rosenbluth, Marshall N. Rosenbluth, Augusta H. Teller, and Edward Teller. Equation of State Calculations by Fast Computing Machines. *The Journal of Chemical Physics*, 21(6):1087–1092, 12 2004.

[28] W. K. Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109, 04 1970.

[29] Andrew Gelman, John B Carlin, Hal S Stern, and Donald B Rubin. *Bayesian data analysis*. Chapman and Hall/CRC, 1995.

[30] Christian P Robert, George Casella, and George Casella. *Monte Carlo statistical methods*, volume 2. Springer, 1999.

[31] Siddhartha Chib and Ivan Jeliazkov. Marginal likelihood from the metropolis–hastings output. *Journal of the American statistical association*, 96(453):270–281, 2001.

[32] Florian Hartig, Francesco Minunno, and Stefan Paul. *BayesianTools: General-Purpose MCMC and SMC Samplers and Tools for Bayesian Statistics*, 2023. R package version 0.1.8.

[33] Heikki Haario, Eero Saksman, and Johanna Tamminen. An adaptive metropolis algorithm. *Bernoulli*, pages 223–242, 2001.

[34] Vivekananda Roy. Convergence diagnostics for markov chain monte carlo. *Annual Review of Statistics and Its Application*, 7:387–412, 2020.

[35] Andrew Gelman and Donald B Rubin. Inference from iterative simulation using multiple sequences. *Statistical science*, pages 457–472, 1992.

[36] Douglas Bates, Katharine M. Mullen, John C. Nash, and Ravi Varadhan. *minqa: Derivative-Free Optimization Algorithms by Quadratic Approximation*, 2022. R package version 1.2.5.

[37] Michael Evans and Tim Swartz. Methods for approximating integrals in statistics with special emphasis on bayesian integration problems. *Statistical science*, pages 254–272, 1995.

[38] Luke Tierney and Joseph B Kadane. Accurate approximations for posterior moments and marginal densities. *Journal of the american statistical association*, 81(393):82–86, 1986.

[39] Robert E Kass. The validity of posterior expansions based on laplace's method. *Bayesian and likelihood methods in statistics and econometrics*, pages 473–487, 1990.

[40] Donna K Pauler, Jonathan C Wakefield, and Robert E Kass. Bayes factors and approximations for variance component models. *Journal of the American Statistical Association*, 94(448):1242–1253, 1999.

[41] Shaun Purcell, Benjamin Neale, Kathe Todd-Brown, Lori Thomas, Manuel A R Ferreira, David Bender, Julian Maller, Pamela Sklar, Paul I W de Bakker, Mark J Daly, and Pak C Sham. PLINK: a tool set for whole-genome association and population-based linkage analyses. *Am J Hum Genet*, 81(3):559–575, July 2007.

[42] Matteo Sesia, Chiara Sabatti, and Emmanuel J Candès. Gene hunting with hidden markov model knockoffs. *Biometrika*, 106(1):1–18, 2019.

[43] Andrew J Saykin, Li Shen, Tatiana M Foroud, Steven G Potkin, Shanker Swaminathan, Sungeun Kim, Shannon L Risacher, Kwangsik Nho, Matthew J Huentelman, David W Craig, et al. Alzheimer's disease neuroimaging initiative biomarkers as quantitative phenotypes: Genetics core aims, progress, and plans. *Alzheimer's & Dementia*, 6(3):265–273, 2010.

[44] Jennifer Harrow, Adam Frankish, Jose M Gonzalez, Electra Tapanari, Mark Diekhans, Felix Kokocinski, Bronwen L Aken, Daniel Barrell, Amonida Zadissa, Stephen Searle, If Barnes, Alexandra Bignell, Veronika Boychenko, Toby Hunt, Mike Kay, Gaurab Mukherjee, Jeena Rajan, Gloria Despacio-Reyes, Gary Saunders, Charles Steward, Rachel Harte, Michael Lin, Cédric Howald, Andrea Tanzer, Thomas Derrien, Jacqueline Chrast, Nathalie Walters, Suganthi Balasubramanian, Baikang Pei, Michael Tress, Jose Manuel Rodriguez, Iakes Ezkurdia, Jeltje van Baren, Michael Brent, David Haussler, Manolis Kellis, Alfonso Valencia, Alexandre Reymond, Mark Gerstein, Roderic Guigó, and Tim J Hubbard. GENCODE: the reference human genome annotation for the ENCODE project. *Genome Res*, 22(9):1760–1774, September 2012.

[45] Bin Zhang and Steve Horvath. A general framework for weighted gene co-expression network analysis. *Statistical applications in genetics and molecular biology*, 4(1), 2005.

[46] Peter Langfelder and Steve Horvath. Wgcna: an r package for weighted correlation network analysis. *BMC bioinformatics*, 9(1):1–13, 2008.

[47] Hao Wang, Binbin Huang, and Jianrong Wang. Predict long-range enhancer regulation based on protein–protein interactions between transcription factors. *Nucleic Acids Research*, 49(18):10347–10368, 2021.

[48] Pinki Munot, Dawn E Saunders, Dianna M Milewicz, Ellen S Regalado, John R Ostergaard, Kees P Braun, Timothy Kerr, Klaske D Lichtenbelt, Sunny Philip, Christopher Rittey, et al. A novel distinctive cerebrovascular phenotype is associated with heterozygous arg179 acta2 mutations. *Brain*, 135(8):2506–2514, 2012.

[49] Kwangsik Nho, Jason Corneveaux, Sungeun Kim, Hai Lin, Shannon Risacher, Li Shen, S Swaminathan, V Ramanan, Y Liu, T Foroud, Mark Inlow, Ashley Siniard, Rebecca Reiman, P Aisen, Ronald Petersen, R Green, Clifford Jack, Michael Weiner, C Baldwin, and Andrew Saykin. Whole-exome sequencing and imaging genetics identify functional variants for rate of change in hippocampal volume in mild cognitive impairment. *Molecular Psychiatry*, 18:781–787, 04 2013.

[50] Michelle C do Rosario, Guillermo Rodriguez Bey, Bruce Nmezi, Fang Liu, Talia Oranburg, Ana SA Cohen, Keith A Coffman, Maya R Brown, Kirill Kiselyov, Quinten Waisfisz, et al. Variants in the zinc transporter tmem163 cause a hypomyelinating leukodystrophy. *Brain*, 145(12):4202–4209, 2022.

[51] Bernard Ng, Charles C White, Hans-Ulrich Klein, Solveig K Sieberts, Cristin McCabe, Ellis Patrick, Jishu Xu, Lei Yu, Chris Gaiteri, David A Bennett, et al. An xqtl map integrates the

genetic architecture of the human brain's transcriptome and epigenome. *Nature neuroscience*, 20(10):1418–1426, 2017.

[52] Jasper D Sluimer, Wiesje M van der Flier, Giorgos B Karas, Nick C Fox, Philip Scheltens, Frederik Barkhof, and Hugo Vrenken. Whole-brain atrophy rate and cognitive decline: longitudinal mr study of memory clinic patients. *Radiology*, 248(2):590–598, 2008.

[53] Ferdinando Squitieri, Milena Cannella, Maria Simonelli, Jenny Sassone, Tiziana Martino, Eugenio Venditti, Andrea Ciammola, Claudio Colonnese, Luigi Frati, and Andrea Ciarmiello. Distinct brain volume changes correlating with clinical stage, disease progression rate, mutation size, and age at onset prediction as early biomarkers of brain atrophy in huntington's disease. *CNS neuroscience & therapeutics*, 15(1):1–11, 2009.

[54] DM Mezzapesa, A Ceccarelli, F Dicuonzo, A Carella, MF De Caro, M Lopez, V Samarelli, P Livrea, and IL Simone. Whole-brain and regional brain atrophy in amyotrophic lateral sclerosis. *American Journal of Neuroradiology*, 28(2):255–259, 2007.

[55] Emma J Burton, Ian G McKeith, David J Burn, E David Williams, and John T O'Brien. Cerebral atrophy in parkinson's disease with and without dementia: a comparison with alzheimer's disease, dementia with lewy bodies and controls. *Brain*, 127(4):791–800, 2004.

[56] Philip R Jansen, Mats Nagel, Kyoko Watanabe, Yongbin Wei, Jeanne E Savage, Christiaan A de Leeuw, Martijn P van den Heuvel, Sophie van der Sluis, and Danielle Posthuma. Genome-wide meta-analysis of brain volume identifies genomic loci and genes shared with intelligence. *Nature communications*, 11(1):5606, 2020.

[57] Tong Teng, Jie Chen, Yehong Zhang, and Bryan Kian Hsiang Low. Scalable variational bayesian kernel selection for sparse gaussian process regression. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(04):5997–6004, Apr. 2020.

[58] Mehmet Gönen and Ethem Alpaydın. Multiple kernel learning algorithms. *The Journal of Machine Learning Research*, 12:2211–2268, 2011.

[59] Cong Han and Bradley P Carlin. Markov chain monte carlo methods for computing bayes factors: A comparative review. *Journal of the American Statistical Association*, 96(455):1122–1132, 2001.

[60] Thomas J DiCiccio, Robert E Kass, Adrian Raftery, and Larry Wasserman. Computing bayes factors by combining simulation and asymptotic approximations. *Journal of the American Statistical Association*, 92(439):903–915, 1997.

[61] Sandip Sinharay and Hal S Stern. An empirical comparison of methods for computing bayes factors in generalized linear mixed models. *Journal of Computational and Graphical Statistics*, 14(2):415–435, 2005.

[62] Nial Friel and Anthony Pettitt. Marginal likelihood estimation via power posteriors. *Journal of the Royal Statistical Society Series B*, 70:589–607, 07 2008.

[63] Bradley Efron. *Large-scale inference: empirical Bayes methods for estimation, testing, and prediction*, volume 1. Cambridge University Press, 2012.

[64] Jon Wakefield. Bayes factors for genome-wide association studies: comparison with p-values. *Genetic Epidemiology*, 33(1):79–86, 2009.

[65] Matthew Stephens and David J Balding. Bayesian statistical methods for genetic association studies. *Nature Reviews Genetics*, 10(10):681–690, 2009.

[66] Jian Zhou and Olga G Troyanskaya. Predicting effects of noncoding variants with deep learning–based sequence model. *Nature methods*, 12(10):931–934, 2015.

[67] Jian Zhou, Chandra L Theesfeld, Kevin Yao, Kathleen M Chen, Aaron K Wong, and Olga G Troyanskaya. Deep learning sequence-based ab initio prediction of variant effects on expression and disease risk. *Nature genetics*, 50(8):1171–1179, 2018.

[68] David R Kelley, Jasper Snoek, and John L Rinn. Basset: learning the regulatory code of the accessible genome with deep convolutional neural networks. *Genome research*, 26(7):990–999, 2016.

[69] Žiga Avsec, Vikram Agarwal, Daniel Visentin, Joseph R Ledsam, Agnieszka Grabska-Barwinska, Kyle R Taylor, Yannis Assael, John Jumper, Pushmeet Kohli, and David R Kelley. Effective gene expression prediction from sequence by integrating long-range interactions. *Nature methods*, 18(10):1196–1203, 2021.

[70] Sanja Vickovic, Gökcen Eraslan, Fredrik Salmén, Johanna Klughammer, Linnea Stenbeck, Denis Schapiro, Tarmo Äijö, Richard Bonneau, Ludvig Bergenstråhle, José Fernandéz Navarro, et al. High-definition spatial transcriptomics for in situ tissue profiling. *Nature methods*, 16(10):987–990, 2019.

[71] Samuel G Rodriques, Robert R Stickels, Aleksandrina Goeva, Carly A Martin, Evan Murray, Charles R Vanderburg, Joshua Welch, Linlin M Chen, Fei Chen, and Evan Z Macosko. Slide-seq: A scalable technology for measuring genome-wide expression at high spatial resolution. *Science*, 363(6434):1463–1467, 2019.

[72] Robert R Stickels, Evan Murray, Pawan Kumar, Jilong Li, Jamie L Marshall, Daniela J Di Bella, Paola Arlotta, Evan Z Macosko, and Fei Chen. Highly sensitive spatial transcriptomics at near-cellular resolution with slide-seqv2. *Nature biotechnology*, 39(3):313–319, 2021.

[73] Patrick AP Moran. Notes on continuous stochastic phenomena. *Biometrika*, 37(1/2):17–23, 1950.

[74] Robert C Geary. The contiguity ratio and statistical mapping. *The incorporated statistician*, 5(3):115–146, 1954.

[75] Valentine Svensson, Sarah A. Teichmann, and Oliver Stegle. Spatialde: identification of spatially variable genes. *Nature Methods*, 15(5):343–346, May 2018.

[76] Ilia Kats, Roser Vento-Tormo, and Oliver Stegle. Spatialde2: Fast and localized variance component analysis of spatial transcriptomics. *bioRxiv*, 2021.

[77] Shiquan Sun, Jiaqiang Zhu, and Xiang Zhou. Statistical analysis of spatial expression patterns for spatially resolved transcriptomic studies. *Nature Methods*, 17(2):193–200, Feb 2020.

[78] Lukas M. Weber, Arkajyoti Saha, Abhirup Datta, Kasper D. Hansen, and Stephanie C. Hicks. nnsvg for the scalable identification of spatially variable genes using nearest-neighbor gaussian processes. *Nature Communications*, 14(1):4059, Jul 2023.

[79] Qiwei Li, Minzhe Zhang, Yang Xie, and Guanghua Xiao. Bayesian modeling of spatial molecular profiling data via Gaussian process. *Bioinformatics*, 37(22):4129–4136, 06 2021.

[80] Daniel Edsgärd, Per Johnsson, and Rickard Sandberg. Identification of spatial expression trends in single-cell gene expression data. *Nature Methods*, 15(5):339–342, May 2018.

[81] Ke Zhang, Wanwan Feng, and Peng Wang. Identification of spatially variable genes with graph cuts. *Nature Communications*, 13(1):5488, Sep 2022.

[82] Alma Andersson and Joakim Lundeberg. sepal: identifying transcript profiles with spatial patterns by diffusion-based modeling. *Bioinformatics*, 37(17):2644–2650, 03 2021.

[83] Junjie Zhu and Chiara Sabatti. Integrative spatial single-cell analysis with graph-based feature learning. *bioRxiv*, 2020.

[84] Brendan F Miller, Dhananjay Bambah-Mukku, Catherine Dulac, Xiaowei Zhuang, and Jean Fan. Characterizing spatial gene expression heterogeneity in spatially resolved single-cell transcriptomic data with nonuniform cellular densities. *Genome Res*, 31(10):1843–1855, May 2021.

[85] Ruben Dries, Qian Zhu, Rui Dong, Chee-Huat Linus Eng, Huipeng Li, Kan Liu, Yuntian Fu, Tianxiao Zhao, Arpan Sarkar, Feng Bao, et al. Giotto: a toolbox for integrative analysis and visualization of spatial expression data. *Genome biology*, 22:1–31, 2021.

[86] Yuhan Hao, Stephanie Hao, Erica Andersen-Nissen, William M Mauck, Shiwei Zheng, Andrew Butler, Maddie J Lee, Aaron J Wilk, Charlotte Darby, Michael Zager, et al. Integrated analysis of multimodal single-cell data. *Cell*, 184(13):3573–3587, 2021.

[87] Giovanni Palla, Hannah Spitzer, Michal Klein, David Fischer, Anna Christina Schaar, Louis Benedikt Kuemmerle, Sergei Rybakov, Ignacio L. Ibarra, Olle Holmberg, Isaac Virshup, Mohammad Lotfollahi, Sabrina Richter, and Fabian J. Theis. Squidpy: a scalable framework for spatial omics analysis. *Nature Methods*, 19(2):171–178, February 2022.

[88] Jiaqiang Zhu, Shiquan Sun, and Xiang Zhou. Spark-x: non-parametric modeling enables scalable and robust detection of spatial expression patterns for large spatial transcriptomic studies. *Genome Biology*, 22(1):184, Jun 2021.

[89] Rui Jiang, Zhen Li, Yuhang Jia, Siyu Li, and Shengquan Chen. SINFONIA: Scalable identification of spatially variable genes for deciphering spatial domains. *Cells*, 12(4), February 2023.

[90] Nuha BinTayyash, Sokratia Georgaka, S T John, Sumon Ahmed, Alexis Boukouvalas, James Hensman, and Magnus Rattray. Non-parametric modelling of temporal and spatial counts data from RNA-seq experiments. *Bioinformatics*, 37(21):3788–3795, November 2021.

[91] Peiyao Zhao, Jiaqiang Zhu, Ying Ma, and Xiang Zhou. Modeling zero inflation is not necessary for spatial transcriptomics. *Genome Biology*, 23(1):118, May 2022.

[92] Xi Jiang, Guanghua Xiao, and Qiwei Li. A bayesian modified ising model for identifying spatially variable genes from spatial transcriptomics data. *Statistics in Medicine*, 41(23):4647–4665, 2022.

[93] Aaron TL Lun, Davis J McCarthy, and John C Marioni. A step-by-step workflow for low-level analysis of single-cell rna-seq data with bioconductor. *F1000Research*, 5, 2016.

[94] Davis J McCarthy, Kieran R Campbell, Aaron TL Lun, and Quin F Wills. Scater: pre-processing, quality control, normalization and visualization of single-cell rna-seq data in r. *Bioinformatics*, 33(8):1179–1186, 2017.

[95] Jinge Yu and Xiangyu Luo. Identification of cell-type-specific spatially variable genes accounting for excess zeros. *Bioinformatics*, 38(17):4135–4144, 07 2022.

[96] Xin Yuan, Yanran Ma, Ruitian Gao, Shuya Cui, Yifan Wang, Botao Fa, Shiyang Ma, Ting Wei, Shuangge Ma, and Zhangsheng Yu. Heartsvg: a fast and accurate method for spatially variable gene identification in large-scale spatial transcriptomic data. *bioRxiv*, 2023.

[97] Minsheng Hao, Kui Hua, and Xuegong Zhang. SOMDE: a scalable method for identifying spatially variable genes with self-organizing map. *Bioinformatics*, 37(23):4392–4398, 06 2021.

[98] Julien Moehlin, Bastien Mollet, Bruno Maria Colombo, and Marco Antonio Mendoza-Parra. Inferring biologically relevant molecular tissue substructures by agglomerative clustering of digitized spatial transcriptomes with multilayer. *Cell Systems*, 12(7):694–705.e3, 2021.

[99] Juexin Wang, Jinpu Li, Skyler T Kramer, Li Su, Yuzhou Chang, Chunhui Xu, Michael T Eadon, Krzysztof Kiryluk, Qin Ma, and Dong Xu. Dimension-agnostic and granularity-based spatially variable gene identification using bsp. *Nature Communications*, 14(1):7367, 2023.

[100] Yaowu Liu, Sixing Chen, Zilin Li, Alanna C Morrison, Eric Boerwinkle, and Xihong Lin. Acat: a fast and powerful p value combination method for rare-variant analysis in sequencing studies. *The American Journal of Human Genetics*, 104(3):410–421, 2019.

[101] Natesh S. Pillai and Xiao-Li Meng. An unexpected encounter with cauchy and lévy, 2015.

[102] Daowen Zhang and Xihong Lin. Hypothesis testing in semiparametric additive mixed models. *Biostatistics*, 4(1):57–74, January 2003.

[103] Faming Liang. A double metropolis–hastings sampler for spatial models with intractable normalizing constants. *Journal of Statistical Computation and Simulation*, 80:1007–1022, 09 2010.

[104] Lulu Shang and Xiang Zhou. Spatially aware dimension reduction for spatial transcriptomics. *Nature Communications*, 13(1):7203, Nov 2022.

[105] Yoav Benjamini and Yosef Hochberg. Controlling the false discovery rate: A practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society: Series B (Methodological)*, 57(1):289–300, 1995.

[106] John D Storey and Robert Tibshirani. Statistical significance for genomewide studies. *Proceedings of the National Academy of Sciences*, 100(16):9440–9445, 2003.

[107] Yoav Benjamini and Daniel Yekutieli. The control of the false discovery rate in multiple testing under dependency. *The Annals of Statistics*, 29(4):1165 – 1188, 2001.

[108] William N Venables and Brian D Ripley. *Modern applied statistics with S-PLUS*. Springer Science & Business Media, 2013.

[109] Jiaqiang Zhu, Lulu Shang, and Xiang Zhou. Srtsim: spatial pattern preserving simulations for spatially resolved transcriptomics. *Genome Biology*, 24(1):39, Mar 2023.

[110] Carissa Chen, Hani Jieun Kim, and Pengyi Yang. Evaluating spatially variable gene detection methods for spatial transcriptomics data. *bioRxiv*, 2022.

[111] Jian Hu, Xiangjie Li, Kyle Coleman, Amelia Schroeder, Nan Ma, David J. Irwin, Edward B. Lee, Russell T. Shinohara, and Mingyao Li. Spagcn: Integrating gene expression, spatial location and histology to identify spatial domains and spatially variable genes by graph convolutional network. *Nature Methods*, 18(11):1342–1351, Nov 2021.

[112] Chihao Zhang, Kangning Dong, Kazuyuki Aihara, Luonan Chen, and Shihua Zhang. Stamarker: Determining spatial domain-specific variable genes with saliency maps in deep learning. *bioRxiv*, 2022.

[113] Marc Elosua-Bayes, Paula Nieto, Elisabetta Mereu, Ivo Gut, and Holger Heyn. SPOTlight: seeded NMF regression to deconvolute spatial transcriptomics spots with single-cell transcriptomes. *Nucleic Acids Research*, 49(9):e50–e50, 02 2021.

[114] Dylan M Cable, Evan Murray, Luli S Zou, Aleksandrina Goeva, Evan Z Macosko, Fei Chen, and Rafael A Irizarry. Robust decomposition of cell type mixtures in spatial transcriptomics. *Nat Biotechnol*, 40(4):517–526, February 2021.

[115] Rui Dong and Guo-Cheng Yuan. Spatialdwls: accurate deconvolution of spatial transcriptomic data. *Genome biology*, 22(1):145, 2021.

[116] Dongyuan Song, Qingyang Wang, Guanao Yan, Tianyang Liu, Tianyi Sun, and Jingyi Jessica Li. scdesign3 generates realistic in silico data for multimodal single-cell and spatial omics. *Nature Biotechnology*, 42(2):247–252, 2024.

[117] Edward Zhao, Matthew R Stone, Xing Ren, Jamie Guenthoer, Kimberly S Smythe, Thomas Pulliam, Stephen R Williams, Cedric R Uytingco, Sarah EB Taylor, Paul Nghiem, et al. Spatial transcriptomics at subspot resolution with bayesspace. *Nature biotechnology*, 39(11):1375–1384, 2021.

[118] Michael E Tipping and Christopher M Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 61(3):611–622, 1999.

[119] Sikta Das Adhikari, Jiaxin Yang, Jianrong Wang, and Yuehua Cui. Recent advances in spatially variable gene detection in spatial transcriptomics. *Comput. Struct. Biotechnol. J.*, February 2024.

[120] Zhijian Li, Zain M Patel, Dongyuan Song, Guanao Yan, Jingyi Jessica Li, and Luca Pinello. Benchmarking computational methods to identify spatially variable genes and peaks. *bioRxivorg*, December 2023.

[121] Charles Swanton. Intratumor heterogeneity: evolution through space and time. *Cancer research*, 72(19):4875–4882, 2012.

[122] Michalina Janiszewska. The microcosmos of intratumor heterogeneity: the space-time of cancer evolution. *Oncogene*, 39(10):2031–2039, 2020.

[123] David T Scadden. Nice neighborhood: emerging concepts of the stem cell niche. *Cell*, 157(1):41–50, 2014.

[124] Miranda V Hunter, Reuben Moncada, Joshua M Weiss, Itai Yanai, and Richard M White. Spatially resolved transcriptomics reveals the architecture of the tumor-microenvironment interface. *Nature communications*, 12(1):6278, 2021.

[125] Qianghu Wang, Baoli Hu, Xin Hu, Hoon Kim, Massimo Squatrito, Lisa Scarpace, Ana C DeCarvalho, Sali Lyu, Pengping Li, Yan Li, et al. Tumor evolution of glioma-intrinsic gene expression subtypes associates with immunological changes in the microenvironment. *Cancer cell*, 32(1):42–56, 2017.

[126] Yi Zhang, Guanjue Xiang, Alva Yijia Jiang, Allen Lynch, Zexian Zeng, Chenfei Wang, Wubing Zhang, Jingyu Fan, Jiajinlong Kang, Shengqing Stan Gu, et al. Metatime integrates single-cell gene expression to characterize the meta-components of the tumor immune microenvironment. *Nature communications*, 14(1):2634, 2023.

[127] Srivatsan Raghavan, Peter S Winter, Andrew W Navia, Hannah L Williams, Alan DenAdel, Kristen E Lowder, Jennyfer Galvez-Reyes, Radha L Kalekar, Nolawit Mulugeta, Kevin S Kapner, et al. Microenvironment drives cell state, plasticity, and drug response in pancreatic cancer. *Cell*, 184(25):6119–6137, 2021.

[128] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

[129] Kristen R Maynard, Leonardo Collado-Torres, Lukas M Weber, Cedric Uytingco, Brianna K Barry, Stephen R Williams, Joseph L Catallini, Matthew N Tran, Zachary Besich, Madhavi Tippani, et al. Transcriptome-scale spatial gene expression in the human dorsolateral prefrontal cortex. *Nature neuroscience*, 24(3):425–436, 2021.

[130] Alma Andersson, Ludvig Larsson, Linnea Stenbeck, Fredrik Salmén, Anna Ehinger, Sunny Z Wu, Ghamdan Al-Eryani, Daniel Roden, Alex Swarbrick, Åke Borg, et al. Spatial deconvolution of her2-positive breast cancer delineates tumor-associated cell type interactions. *Nature communications*, 12(1):6012, 2021.

[131] Patrik L Ståhl, Fredrik Salmén, Sanja Vickovic, Anna Lundmark, José Fernández Navarro, Jens Magnusson, Stefania Giacomello, Michaela Asp, Jakub O Westholm, Mikael Huss, et al. Visualization and analysis of gene expression in tissue sections by spatial transcriptomics. *Science*, 353(6294):78–82, 2016.

[132] Jiaqiang Zhu, Shiquan Sun, and Xiang Zhou. Spark-x: non-parametric modeling enables scalable and robust detection of spatial expression patterns for large spatial transcriptomic studies. *Genome biology*, 22(1):184, 2021.

[133] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 58(1):267–288, 1996.

[134] Jerome Friedman, Trevor Hastie, and Rob Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of statistical software*, 33(1):1, 2010.

[135] Noah Simon, Jerome Friedman, Trevor Hastie, and Rob Tibshirani. Regularization paths for cox's proportional hazards model via coordinate descent. *Journal of statistical software*, 39(5):1, 2011.

[136] Cun-Hui Zhang. Nearly unbiased variable selection under minimax concave penalty. 2010.

[137] Jianqing Fan and Jinchi Lv. Sure independence screening for ultrahigh dimensional feature space. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 70(5):849–911, 2008.

[138] Vincent A Traag, Ludo Waltman, and Nees Jan Van Eck. From louvain to leiden: guaranteeing well-connected communities. *Scientific reports*, 9(1):5233, 2019.

[139] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment*, 2008(10):P10008, 2008.

[140] Pascal Pons and Matthieu Latapy. Computing communities in large networks using random walks. In *Computer and Information Sciences-ISCIS 2005: 20th International Symposium, Istanbul, Turkey, October 26-28, 2005. Proceedings 20*, pages 284–293. Springer, 2005.

[141] Xihong Lin. Variance component testing in generalised linear models with random effects. *Biometrika*, 84(2):309–326, 1997.

[142] Michael C Wu, Seunggeun Lee, Tianxi Cai, Yun Li, Michael Boehnke, and Xihong Lin. Rare-variant association testing for sequencing data with the sequence kernel association test. *The American Journal of Human Genetics*, 89(1):82–93, 2011.

[143] Robert B Davies. The distribution of a linear combination of $\chi 2$ random variables. *Journal of the Royal Statistical Society Series C: Applied Statistics*, 29(3):323–333, 1980.

[144] Yaowu Liu, Sixing Chen, Zilin Li, Alanna C Morrison, Eric Boerwinkle, and Xihong Lin. Acat: a fast and powerful p value combination method for rare-variant analysis in sequencing studies. *The American Journal of Human Genetics*, 104(3):410–421, 2019.

[145] Yaowu Liu and Jun Xie. Cauchy combination test: a powerful test with analytic p-value calculation under arbitrary dependency structures. *J Am Stat Assoc*, 115(529):393–402, April 2019.

[146] Yoav Benjamini and Daniel Yekutieli. The control of the false discovery rate in multiple testing under dependency. *Annals of statistics*, pages 1165–1188, 2001.

[147] Carrie A Davis, Benjamin C Hitz, Cricket A Sloan, Esther T Chan, Jean M Davidson, Idan Gabdank, Jason A Hilton, Kriti Jain, Ulugbek K Baymuradov, Aditi K Narayanan, et al. The encyclopedia of dna elements (encode): data portal update. *Nucleic acids research*, 46(D1):D794–D801, 2018.

[148] Alper Yilmaz, Milton Y Nishiyama Jr, Bernardo Garcia Fuentes, Glaucia Mendes Souza, Daniel Janies, John Gray, and Erich Grotewold. Grassius: a platform for comparative regulatory genomics across the grasses. *Plant physiology*, 149(1):171–180, 2009.

[149] Jens Keilwagen, Stefan Posch, and Jan Grau. Accurate prediction of cell type-specific transcription factor binding. *Genome biology*, 20:1–17, 2019.

[150] Hongyang Li, Daniel Quang, and Yuanfang Guan. Anchor: trans-cell type prediction of transcription factor binding sites. *Genome research*, 29(2):281–292, 2019.

[151] Richard I Sherwood, Tatsunori Hashimoto, Charles W O'donnell, Sophia Lewis, Amira A Barkal, John Peter Van Hoff, Vivek Karun, Tommi Jaakkola, and David K Gifford. Discovery of directional and nondirectional pioneer transcription factors by modeling dnase profile magnitude and shape. *Nature biotechnology*, 32(2):171–178, 2014.

[152] Jacob Schreiber, Jeffrey Bilmes, and William Stafford Noble. Completing the encode3 compendium yields accurate imputations across a variety of assays and human biosamples. *Genome biology*, 21:1–13, 2020.

[153] Yanli Wang, Fan Song, Bo Zhang, Lijun Zhang, Jie Xu, Da Kuang, Daofeng Li, Mayank NK Choudhary, Yun Li, Ming Hu, et al. The 3d genome browser: a web-based browser for visualizing 3d genome organization and long-range chromatin interactions. *Genome biology*, 19:1–12, 2018.

[154] Heidi Sutherland and Wendy A Bickmore. Transcription factories: gene expression in unions? *Nature Reviews Genetics*, 10(7):457–466, 2009.

[155] Cameron S Osborne. Molecular pathways: transcription factories and chromosomal translocations. *Clinical Cancer Research*, 20(2):296–300, 2014.

[156] Qianli Dong, Ning Li, Xiaochong Li, Zan Yuan, Dejian Xie, Xiaofei Wang, Jianing Li, Yanan Yu, Jinbin Wang, Baoxu Ding, et al. Genome-wide hi-c analysis reveals extensive hierarchical chromatin interactions in rice. *The Plant Journal*, 94(6):1141–1156, 2018.

[157] Yuri B Schwartz and Giacomo Cavalli. Three-dimensional genome organization and function in drosophila. *Genetics*, 205(1):5–24, 2017.

[158] Wenxiu Ma, Ferhat Ay, Choli Lee, Gunhan Gulsoy, Xinxian Deng, Savannah Cook, Jennifer Hesson, Christopher Cavanaugh, Carol B Ware, Anton Krumm, et al. Fine-scale chromatin interaction maps reveal the cis-regulatory landscape of human lincrna genes. *Nature methods*, 12(1):71–78, 2015.

[159] Peter H Sudmant, Tobias Rausch, Eugene J Gardner, Robert E Handsaker, Alexej Abyzov, John Huddleston, Yan Zhang, Kai Ye, Goo Jun, Markus Hsi-Yang Fritz, et al. An integrated map of structural variation in 2,504 human genomes. *Nature*, 526(7571):75–81, 2015.

[160] Suhas SP Rao, Miriam H Huntley, Neva C Durand, Elena K Stamenova, Ivan D Bochkov, James T Robinson, Adrian L Sanborn, Ido Machol, Arina D Omer, Eric S Lander, et al. A 3d map of the human genome at kilobase resolution reveals principles of chromatin looping. *Cell*, 159(7):1665–1680, 2014.

[161] David M Blei, Alp Kucukelbir, and Jon D McAuliffe. Variational inference: A review for statisticians. *Journal of the American statistical Association*, 112(518):859–877, 2017.

[162] Toby J Mitchell and John J Beauchamp. Bayesian variable selection in linear regression. *Journal of the american statistical association*, 83(404):1023–1032, 1988.

[163] Peter Carbonetto and Matthew Stephens. Scalable variational inference for bayesian variable selection in regression, and its accuracy in genetic association studies. 2012.

[164] Benjamin A Logsdon, Gabriel E Hoffman, and Jason G Mezey. A variational bayes algorithm for fast and accurate multiple locus genome-wide association analysis. *BMC bioinformatics*, 11:1–13, 2010.

[165] Nello Cristianini and John Shawe-Taylor. *An introduction to support vector machines and other kernel-based learning methods*. Cambridge university press, 2000.

[166] Dawei Liu, Xihong Lin, and Debashis Ghosh. Semiparametric regression of multidimensional genetic pathway data: least-squares kernel machines and linear mixed models. *Biometrics*, 63(4):1079–1088, 2007.

[167] Eugene Demidenko. *Mixed models: theory and applications with R*. John Wiley & Sons, 2013.

[168] David A Harville. Bayesian inference for variance components using only error contrasts. *Biometrika*, 61(2):383–385, 1974.

[169] H Desmond Patterson and Robin Thompson. Recovery of inter-block information when block sizes are unequal. *Biometrika*, 58(3):545–554, 1971.

[170] Nan M Laird and James H Ware. Random-effects models for longitudinal data. *Biometrics*, pages 963–974, 1982.

[171] Regev Schweiger, Omer Weissbrod, Elior Rahmani, Martina Müller-Nurasyid, Sonja Kunze, Christian Gieger, Melanie Waldenberger, Saharon Rosset, and Eran Halperin. Rl-skat: an exact and efficient score test for heritability and set tests. *Genetics*, 207(4):1275–1283, 2017.

[172] Geoffrey E Hinton and Sam Roweis. Stochastic neighbor embedding. *Advances in neural information processing systems*, 15, 2002.

[173] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

[174] Jesse H. Krijthe. *Rtsne: T-Distributed Stochastic Neighbor Embedding using Barnes-Hut Implementation*, 2015. R package version 0.17.

[175] L.J.P. van der Maaten and G.E. Hinton. Visualizing high-dimensional data using t-sne. *Journal of Machine Learning Research*, 9:2579–2605, 2008.

[176] L.J.P. van der Maaten. Accelerating t-sne using tree-based algorithms. *Journal of Machine Learning Research*, 15:3221–3245, 2014.

[177] Mark EJ Newman and Michelle Girvan. Finding and evaluating community structure in networks. *Physical review E*, 69(2):026113, 2004.

[178] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment*, 2008(10):P10008, 2008.

[179] Gabor Csardi and Tamas Nepusz. The igraph software package for complex network research. *InterJournal*, Complex Systems:1695, 2006.

[180] Gábor Csárdi, Tamás Nepusz, Vincent Traag, Szabolcs Horvát, Fabio Zanini, Daniel Noom, and Kirill Müller. *igraph: Network Analysis and Visualization in R*, 2024. R package version 2.0.3.

APPENDIX A

KERNEL BASED MODELS: FROM SEMIPARAMETRIC MODEL TO LMM

As mentioned in chapter 2, under the kernel machine regression framework, continuous quantitative traits can be associated to genetic variants or molecular features, along with additional covariates, through a semiparametric model:

$$y_i = X_i\beta + h(Z_i) + \epsilon_i, \qquad i = 1, 2, \cdots, n \tag{A.1}$$

Here, $h(Z_i)$ is an unknown centered smooth function. Model A.1 models covariate effects parametrically and the spatial effect parametrically or nonparametrically. When h(·) = 0, A.1 reduces to the standard linear regression model.

We assume that the nonparametric function $h(Z)$ resides in a function space $\mathcal{H}_k$ defined by a positive definite kernel function $K(.,.)$. According to Mercer's theorem [165], assuming certain regularity conditions hold, a kernel function $K(.,.)$ implicitly defines a unique function space spanned by a specific set of orthogonal basis functions (features) $\phi_j(Z)_{j=1}^J$. This implies that any $h(Z)$ can be expressed as a linear combination of these bases: $h(Z) = \sum_{j=1}^J \omega_j \phi_j(Z) = \phi_j(Z)^T\omega$ (referred to as the primal representation), where ω is a coefficient vector. Alternatively, $h(Z)$ can be represented using the kernel function $K(.,.)$ as $h(Z) = \sum_{l=1}^L \alpha_l K(Z_l^*, Z)$ (known as the dual representation), where L is an integer, α_l are constants, and $\{Z_1^*, Z_2^*, .., Z_L^*\} \in \mathcal{R}^m$.

Now, assuming $h(Z)$ belongs to $\mathcal{H}_k$, the function space generated by a kernel function $K(.,.)$. Estimation of β and $h(.)$ in A.1 proceeds by maximizing the scaled penalized likelihood function:

$$\begin{aligned} J(h, \beta) &= -\frac{1}{2}\sum_{i=1}^{n}[y_i - x_i\beta - h(Z_i)]^2 - \frac{1}{2}\lambda\|h\|^2_{\mathcal{H}_k} \\ &= -\frac{1}{2}\sum_{i=1}^{n}[y_i - x_i\beta - h(Z_i)]^2 - \frac{1}{2}\lambda\alpha^T K\alpha \end{aligned} \tag{A.2}$$

where λ is a tuning parameter which controls the tradeoff between goodness of fit and complexity of the model. When $\lambda = 0$, the model interpolates the data, whereas when $\lambda = \infty$, the model reduces to a simple linear model without $h(.)$. This is exactly similar to the log-likelihood of the linear mixed model[166] up to some constant:

$$y = X\beta + h + \epsilon \tag{A.3}$$

where the random effect $h \sim N(0, \tau K)$, random noise $\epsilon \sim N(0, \sigma^2)$. Here, $y \sim N(X\beta, \Sigma)$, where $\Sigma = \tau K + \sigma^2 I$ and $y|h \sim (X\beta + h, \sigma^2 I)$. The log-likelihood:

$$\begin{gathered} log(l(y)) = log(l(y|h)l(h)) = log(l(y|h)) + log(l(h)) \\ = log(exp^{-\frac{1}{2\sigma^2}\sum_{i=1}^{n}(y_i - X_i\beta - h_i)^T(y_i - X_i\beta - h_i)}) + log(exp^{-\frac{1}{2\tau}h^T K^{-1} h}) + \text{ constant terms} \\ =-\frac{1}{2\sigma^2}\sum_{i=1}^{n}(y_i - X_i\beta - Z_i)^T(y_i - X_i\beta - h_i) - \frac{1}{2\tau}h^T K^{-1} h \\ =-\sum_{i=1}^{n}[y_i - x_i\beta - h_i]^2 - \frac{1}{2\tau}h^T K^{-1} h \end{gathered}$$

Setting $\tau = 1/\lambda$ and $h = K\alpha$, one can easily see that equations A.2 and A.3 are identical.

APPENDIX B

SCORE STATISTIC AND THE NULL DISTRIBUTION

B.0.1 Score test for SVG detection

The Gaussian process (GP) regression model which models the normalized gene expression y for a given gene using the following multivariate normal model:

$$p(y|\mu, \sigma_s^2, \delta, K) \sim N(y|X\beta, \sigma_s^2 K + \delta I), \tag{B.1}$$

where the covariance term is decomposed into a spatial and a non-spatial part, where δI represents the non-spatial part and $\sigma_s^2 K$ is the spatial covariance matrix, whose $(i, j)^{th}$ element in the kernel matrix K denotes the spatial similarity between the i^{th} and j^{th} spot calculated based on the corresponding coordinates s_i and s_j. The choice of the kernel function plays a very important role in detecting the spatial correlation present in the gene expressions. $X^{N\times k}$ represents the covariate matrix, while $\beta^{N\times 1}$ denotes the array of corresponding coefficients. This model can incorporate up to $k-1$ covariates, such as cell type information or domain structure information.

For the i^{th} location, the model can be written like this:

$$y_i(s_i) = X_i(s_i)\beta + h_i(s_i) + \epsilon_i \tag{B.2}$$

where ϵ_i is the iid residual error which follows $N(0, \delta)$, $\alpha_i(s_i)$ is such that: $h(s) = (h_1(s_1), h_2(s_2), \cdots, h_n(s_i)) \sim MVN(0, \sigma_s^2 K(s))$ is a spatial random effect which captures the spatial variation using the spatial covariate matrix $\sigma_s^2 K$; Overall the covariance for the normalized gene expression $y(s)$ is $\Sigma = \sigma_s^2 K + \delta I$.

As we mentioned in the main text, testing if a gene is a SVG is equivalent to testing $H_0 : \sigma_s^2 = 0$. The null hypothesis $H_0 : \sigma_s^2$ =0 can be tested using the variance-component score test which is the locally most powerful test[141]. The variance-component score statistic is:

$$Q = (y - X\hat{\beta})^T K (y - X\hat{\beta})$$

where $\hat{\beta}$ is the MLEs under the null model. Under the null hypothesis, the score statistic Q follows a mixture of chi-square distributions [142], which can be closely approximated with the computationally efficient Davies' method[143]. More details about the test is provided in the next subsection.

B.1 More details on its null distribution

The Gaussian process model presented in B.1 has many unknown parameters, for example δ, the kernel parameter ρ, σ_s^2. The unknown parameters are estimated simultaneously by treating them as variance components in the linear mixed model and estimating them using the Restricted Maximum Likelihood(RML) model. The RML model was used to reduce bias in the variance components model [167][168][169].

The restricted log-likelihood[170][166] for the lmm in 4.1 is as follows:

$$L_R(\sigma_s^2, \delta, \rho) = -\frac{1}{2}log|\Sigma| - \frac{1}{2}log|X^T\Sigma^{-1}X| - \frac{1}{2}(y - X\beta)^T\Sigma^{-1}(y - X\beta) \tag{B.3}$$

where $\Sigma = \sigma_s^2 K + \delta I$.

The score statistic can be written in the form of $\tilde{Q}(\hat{\beta}, \hat{\delta}) - tr(HK)$[166], where $H = I - X(X^TX)^{-1}X^T$, $\hat{\beta}$ and $\hat{\delta}$ are the MLEs under the null model and $\tilde{Q}(\beta, \delta) = \frac{1}{2\delta}(y - X\beta)^T K(y - X\beta)$

Under H_0, the final score statistic for the test Q reduces to $y^T H^T KHy$ and

$$Q \sim \sum_{i=1}^{N} \lambda_i \chi_{i,1}^2 \tag{B.4}$$

where $\chi_{i,1}^2$ are independent χ_1^2 random variables and $\lambda_1, \lambda_2, \cdots, \lambda_N$ are the eigenvalues of $\hat{\delta}H^{1/2}KH^{1/2}$.

The form of the null distribution of the score statistic distribution stated follows from this argument [171]. Under H_0, $y \sim N(X\beta, \delta I)$, the covariates can be removed using the transformation $\tilde{y} = Hy$. Now $\tilde{y} \sim N(0, \delta HH^T)$ and $K^{1/2}\tilde{y} \sim N(0, \delta K^{1/2}HH^TK^{1/2})$. Let U be the matrix whose columns are the eigenvectors of $\hat{\delta}K^{1/2}HH^TK^{1/2}$ and the corresponding eigenvalues are $\tilde{\lambda}_1, \cdots, \tilde{\lambda}_n$. Therefore, $U^TK^{1/2}Hy \sim N(0, diag(\tilde{\lambda}_i))$ and $Q = (U^TK^{1/2}Hy)^T(U^TK^{1/2}Hy) \sim \sum_{i=1}^{n} \tilde{\lambda}_i \chi_{i,1}^2$. Using the fact that the eigenvalues of any matrix A, AA^T, A^TA are the same and $H^2 = H$, it can be shown that $\tilde{\lambda}_i = \lambda_i$, the eigen values of $\hat{\delta}H^{1/2}KH^{1/2}$.

APPENDIX C

T-SNE PLOT

t-Distributed Stochastic Neighbor Embedding (t-SNE) [172][173] is a powerful nonlinear dimensionality reduction technique designed to facilitate the visualization of high-dimensional data by mapping it into two or three dimensions. This reduction process enables the identification of inherent patterns, clusters, and relationships that are otherwise difficult to discern.

t-SNE begins by calculating pairwise similarities between data points x_i and x_j in the high-dimensional space. The similarity of datapoint x_j to datapoint x_i is the conditional probability, $p_{j|i}$, that x_i would pick x_j as its neighbor if neighbors were picked in proportion to their probability density under a Gaussian centered at x_i . For nearby data points, $p_{j|i}$ is relatively high, while for widely separated data points, $p_{j|i}$ becomes almost infinitesimal.$p_{j|i}$ is given by:

$$p_{j|i} = \frac{\exp\left(-\|x_i - x_j\|^2/2\sigma_i^2\right)}{\sum_{k\neq i}\exp\left(-\|x_i - x_k\|^2/2\sigma_i^2\right)}$$

For the low-dimensional counterparts y_i and y_j of the high-dimensional data points x_i and x_j , we model the similarity of map point y_j to map point y_i by:

$$q_{j|i} = \frac{\exp\left(-\|y_i - y_j\|^2\right)}{\sum_{k\neq i}\exp\left(-\|y_i - y_k\|^2\right)}$$

If the map points y_i and y_j correctly model the similarity between the high-dimensional data points x_i and x_j, the conditional probabilities $p_{j|i}$ and $q_{j|i}$ will be equal. To achieve this, t-SNE aims to minimize the mismatch between p_{ij} and q_{ij}. This is done by minimizing the sum of Kullback-Leibler divergences over all data points using gradient descent. The cost function C is given by:

$$C = \sum_i KL(P_i\|Q_i) = \sum_i \sum_j p_{j|i} \log \frac{p_{j|i}}{q_{j|i}}$$

where P_i represents the conditional probability distribution over all other datapoints given datapoint x_i , and Q_i represents the conditional probability distribution over all other map points given map point y_i.

In summary, t-SNE reduces high-dimensional data to two or three dimensions by preserving the pairwise similarities between points as much as possible, thereby facilitating the visualization of complex data structures. Because the Kullback-Leibler divergence is not symmetric, different types of errors in the pairwise distances in the low-dimensional map are weighted unequally. Specifically, there is a high cost for using widely separated map points to represent nearby data points (i.e., using a small $q_{j|i}$ to model a large $p_{j|i}$). Conversely, there is a relatively small cost for using nearby map points to represent widely separated data points. This means that the SNE cost function prioritizes preserving the local structure of the data in the map.

In Chapter 4, t-SNE plots are extensively used to illustrate the similarity between the gene clusters identified by cSVG. The R package "Rtsne" [174][175][176] has been used for that purpose. In complex biological data, traditional methods such as PCA often fall short, necessitating the use of advanced techniques like t-SNE to achieve better performance and more meaningful visualizations.

APPENDIX D

COMMUNITY DETECTION ALGORITHMS

In many complex networks, nodes cluster and form relatively dense groups—often called communities, where the nodes within each community are more densely connected to each other than to the rest of the network. Such a modular structure is usually not known beforehand, making the detection of communities an important problem. One of the best-known methods for community detection is called modularity[177]. This method tries to maximize the difference between the actual number of edges in a community and the expected number of such edges. Community detection algorithms use various methodologies to partition networks into meaningful clusters, revealing insights into the relationships and interactions within the network. Different community detection algorithms like Louvain and Leiden are widely used in single-cell RNA sequencing (scRNA-seq) analysis for clustering single cells, aiding in the identification of distinct cell types and states.

The modularity is defined by:

$$\mathcal{H} = \frac{1}{2m} \sum_{c} (e_c - \gamma \frac{K_c^2}{2m})$$

where m is the total number of edges in the network, e_c is the actual number of edges in community c, the expected number of edges can be expressed as $\frac{K_c^2}{2m}$, where K_c is the sum of the degrees of the nodes in the community c. $\gamma > 0$ is a resolution parameter. Higher resolutions lead to more communities, while lower resolutions lead to fewer communities. The Louvain method[178] is one of the most popular community detection algorithms due to its simplicity and effectiveness, typically operating based on modularity optimization. However, in recent years, some drawbacks of the Louvain algorithm have been identified[138], leading to the increased popularity of the Leiden algorithm. The Leiden method is more robust and accurate, making it better suited for analyzing complex networks where high-quality community detection is crucial.

In step 3 of the cSVG algorithm, we use the Leiden algorithm, implemented through the "igraph" R package[179],[180] with the default resolution parameter value $\gamma = 1$.

APPENDIX E

CHAPTER 2 SUPPLEMENTARY TABLES AND FIGURES

Table E.1 List of significant genes selected by BayesKAT and their posterior probability of association.

Gene Name	p(H1|Data)
CARD10	0.9835514
MGAT5	0.9474112
TMEM71	0.9237384
FAM174B	0.9042780
ACTA2	0.8677653
C18orf45	0.8569289
TMEM163	0.8357041
LRP1B	0.8257515
SLC35B4	0.8248786
EDNRA	0.7878737
ABCB7	0.7699132
C9orf135	0.7660853
SLC25A18	0.7583290
DKK2	0.7501011
PCID2	0.7387756
TMEM38A	0.7042073
NXNL2	0.7037605

Table E.2 List of the significant KEGG pathways selected by BayesKAT and their corresponding posterior probabilities of association.

Pathway Name	p(H1|Data)
Pathways of neurodegeneration - multiple diseases	0.9998840
Alzheimer disease	0.9982576
Salmonella infection	0.9897521
Antifolate resistance	0.9666899
Huntington disease	0.9630876
Bile secretion	0.9530779
Alcoholic liver disease	0.9285561
Dilated cardiomyopathy	0.9240475
Metabolic pathways	0.9211165
Amyotrophic lateral sclerosis	0.9070785
Hypertrophic cardiomyopathy	0.8975824
Cardiac muscle contraction	0.8966726
mTOR signaling pathway	0.8869947
cAMP signaling pathway	0.8688107
Pathogenic Escherichia coli infection	0.8645842
Calcium signaling pathway	0.7962372
ABC transporters	0.7851350
Citrate cycle (TCA cycle)	0.7766141
Non-alcoholic fatty liver disease	0.7676582
AMPK signaling pathway	0.7640913
Parkinson disease	0.7242965

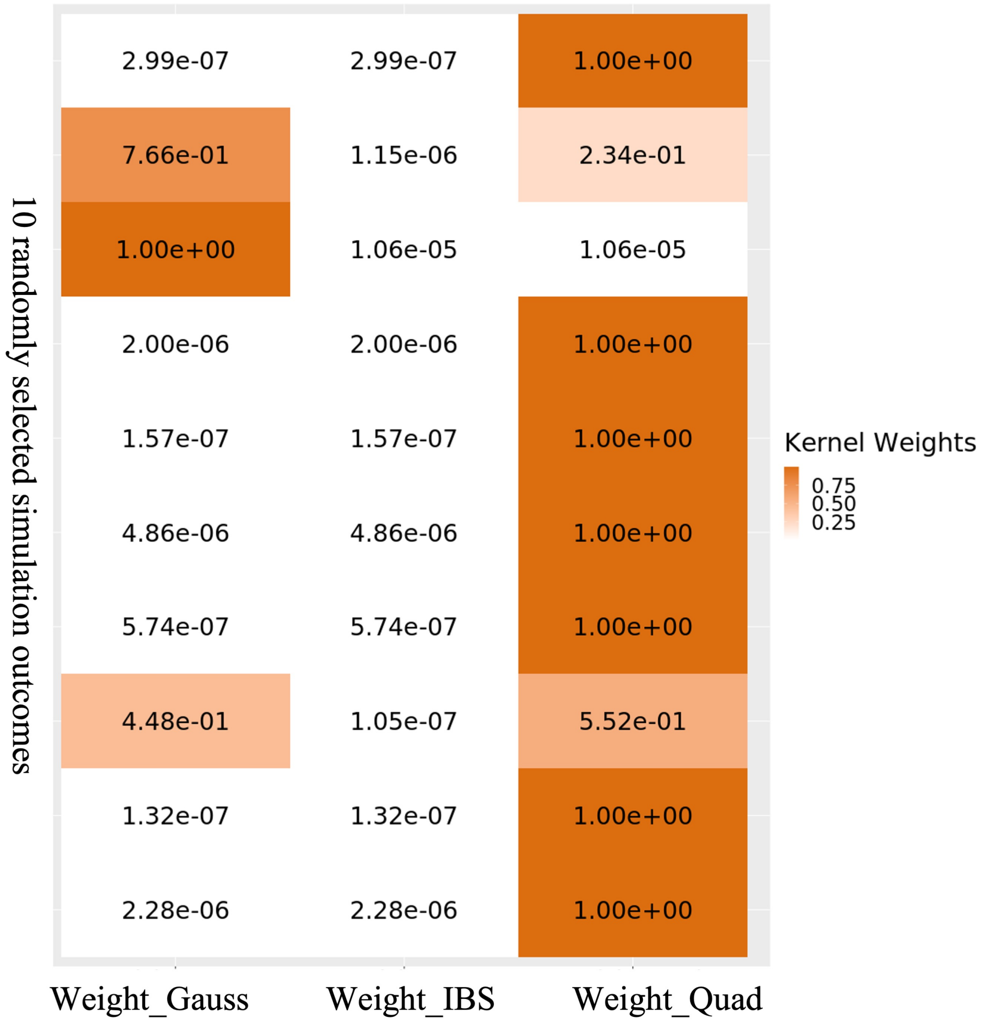


Figure E.1 Inferred weights of each candidate kernel within the composite kernels based on the simulation tests as presented in Figure 1(B). The inferred composite kernel demonstrates strong agreements with the underlying true kernel function (the Quadratic kernel) used for data generation. High kernel weights are consistently inferred for the Quadratic kernel.

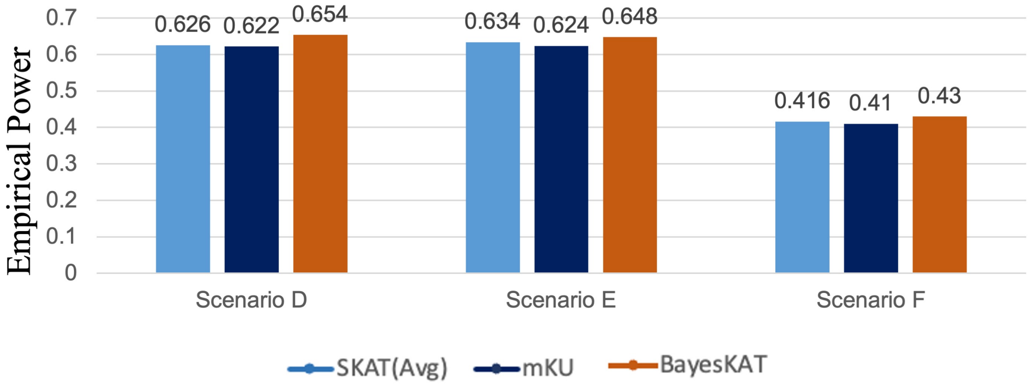


Figure E.2 Performance comparison based on simulations involving unrelated discrete genetic features. With the same fixed level of empirical type 1 error, the empirical power of all three methods are overall low, because this basic simulation only groups unrelated genetic features together, which is not recommended for real-world genetic association studies. Even through, BayesKAT still achieves better empirical power.

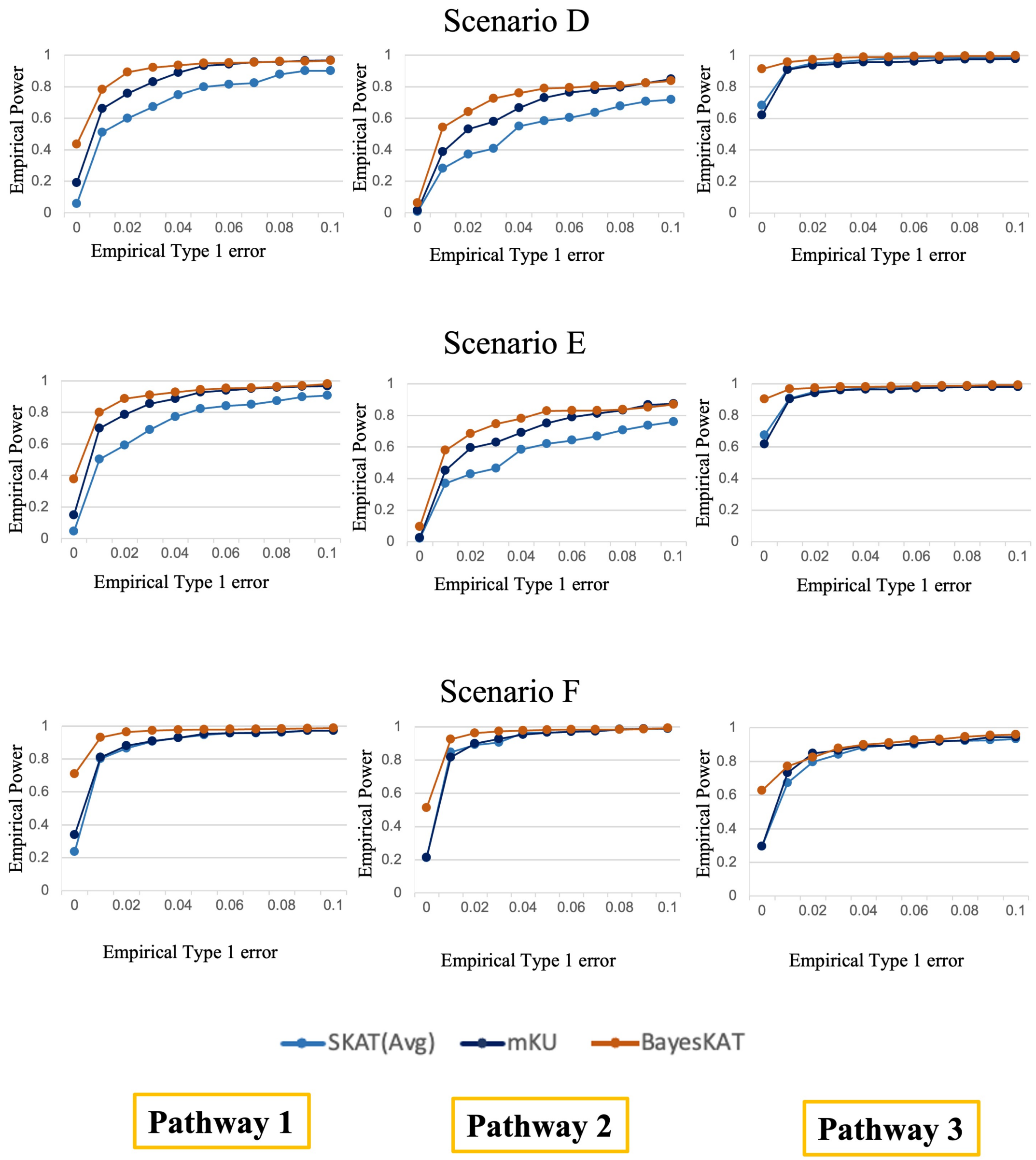
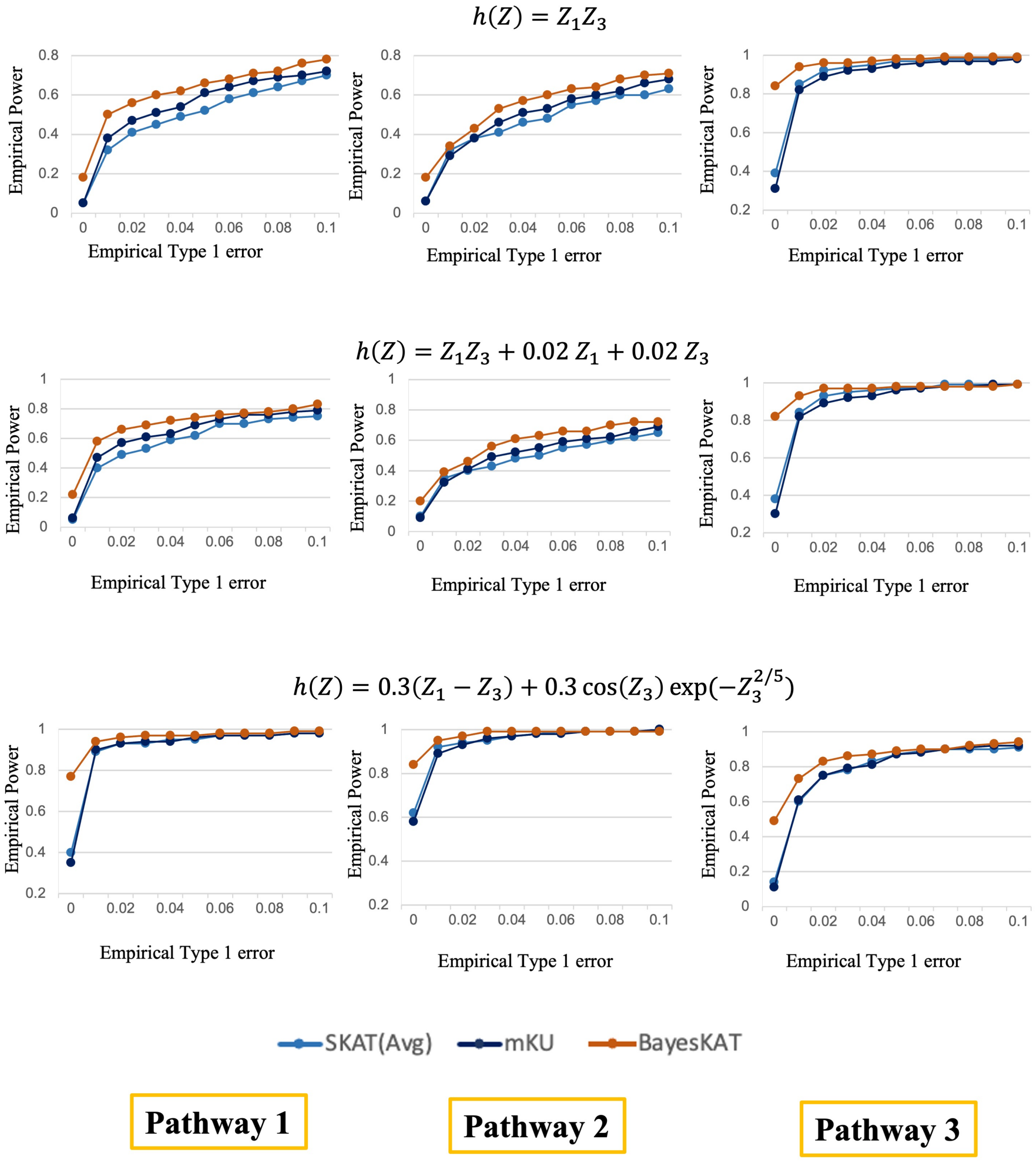


Figure E.3 The empirical power vs empirical type 1 error plot from the pathway-based simulations with sample size n=755. For each scenario, BayesKAT consistently achieves the best performance. The aggregated result presented in main figure 3(B).



$h(Z) = Z_1 Z_3$

$h(Z) = Z_1 Z_3 + 0.02\,Z_1 + 0.02\,Z_3$

$h(Z) = 0.3(Z_1 - Z_3) + 0.3\cos(Z_3)\exp(-Z_3^{2/5})$

Figure E.4 The empirical power vs empirical type 1 error plot from the pathway-based simulations with sample size n=1000. The strength of the relationship is intentionally weakened by adjusting effect sizes for performance comparison purposes. Note that for larger sample sizes under the scenarios in S3, all methods exhibit a power of 1, making them incomparable. For each scenario, BayesKAT consistently achieves the best performance.

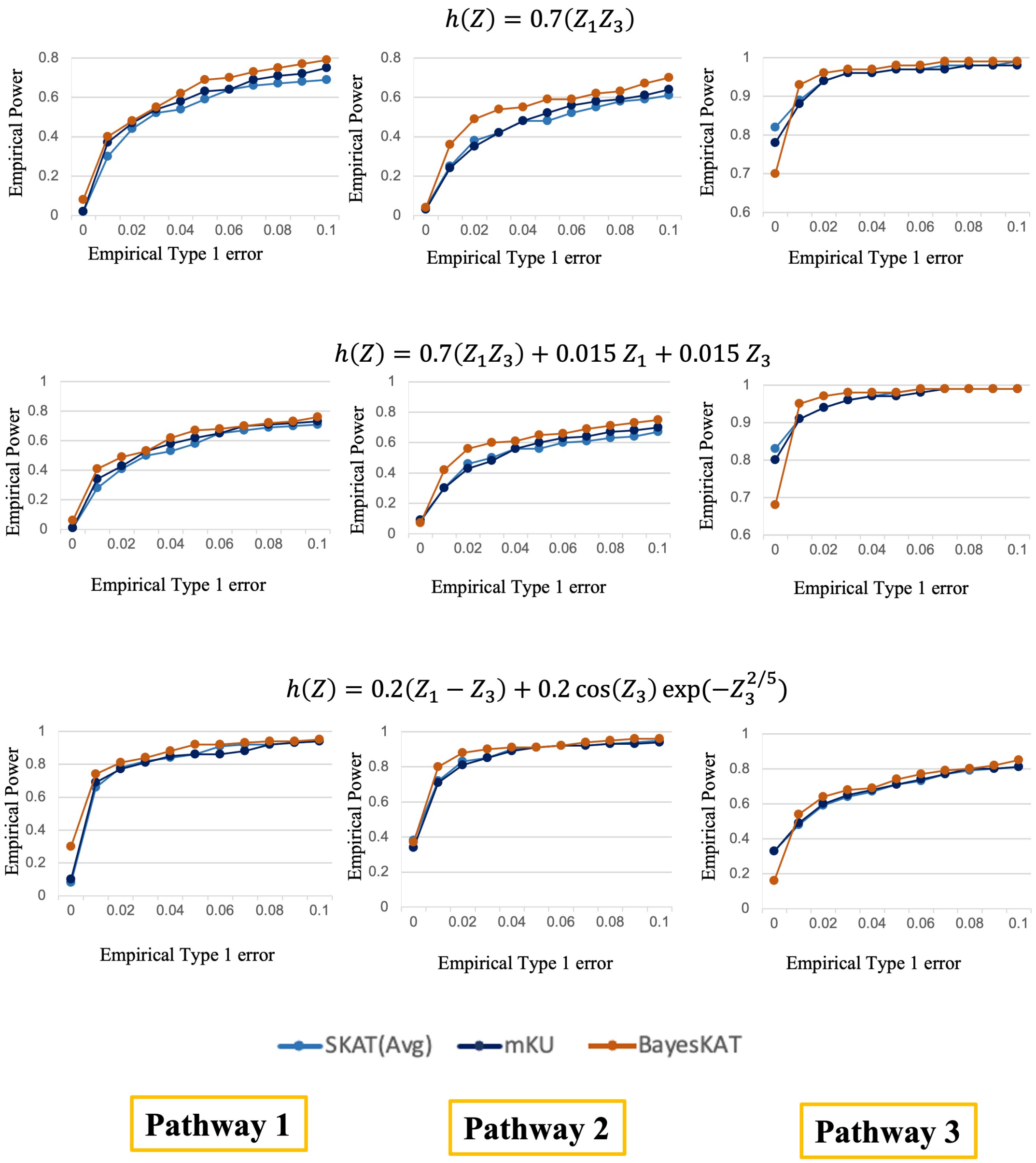


Figure E.5 The empirical power vs empirical type 1 error plot from the pathway-based simulations with sample size n=1500. The strength of the relationship is intentionally weakened by adjusting effect sizes for performance comparison purposes. Note that for larger sample sizes under the scenarios in S3, all methods exhibit a power of 1, making them incomparable. BayesKAT consistently performs reasonably well for each scenario.

APPENDIX F

CHAPTER 4 SUPPLEMENTARY FIGURES

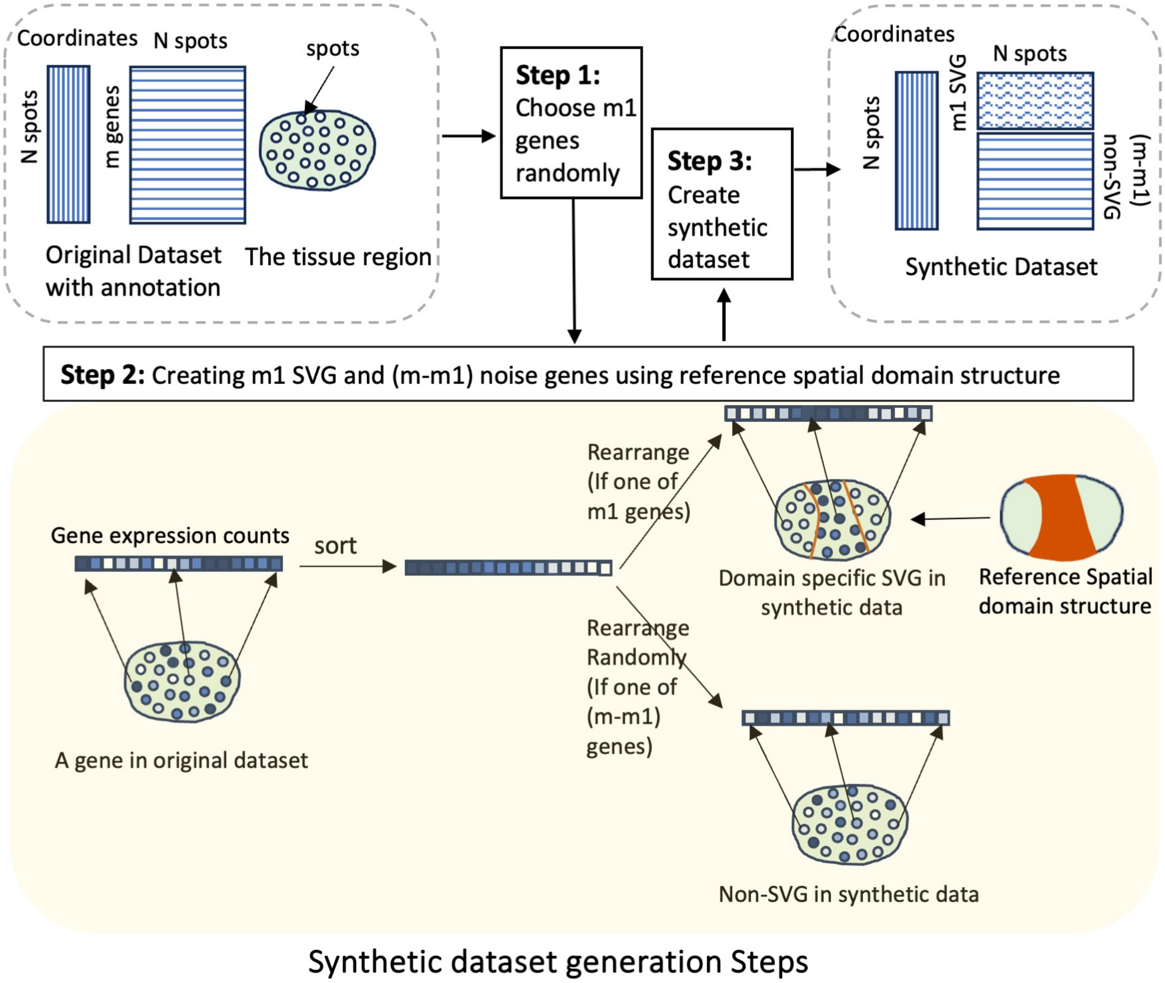

Coordinates
N spots
spots
N spots
m genes
Original Dataset with annotation
The tissue region
Step 1: Choose m1 genes randomly
Step 3: Create synthetic dataset
Coordinates
N spots
m1 SVG
N spots
(m-m1) non-SVG
Synthetic Dataset
Step 2: Creating m1 SVG and (m-m1) noise genes using reference spatial domain structure
Gene expression counts
sort
A gene in original dataset
Rearrange (If one of m1 genes)
Rearrange Randomly (If one of (m-m1) genes)
Domain specific SVG in synthetic data
Reference Spatial domain structure
Non-SVG in synthetic data
Synthetic dataset generation Steps

Figure F.1 Steps to create a synthetic dataset from an annotated original Spatial dataset. We begin with a filtered gene expression count matrix and a location matrix where spots are annotated. The tissue region contains N spots where gene expression measurements are measured for each of the m genes. **Step 1:** Randomly select m_1 genes from the original dataset to be converted into new SVGs in the synthetic dataset. The remaining $m - m_1$ genes will serve as noise genes with no spatial pattern. **Step 2:** For each of the m_1 genes in the original dataset, sort and arrange the gene expression count values according to the reference domain structure. For the remaining genes, sort and arrange the count values randomly on the tissue region.

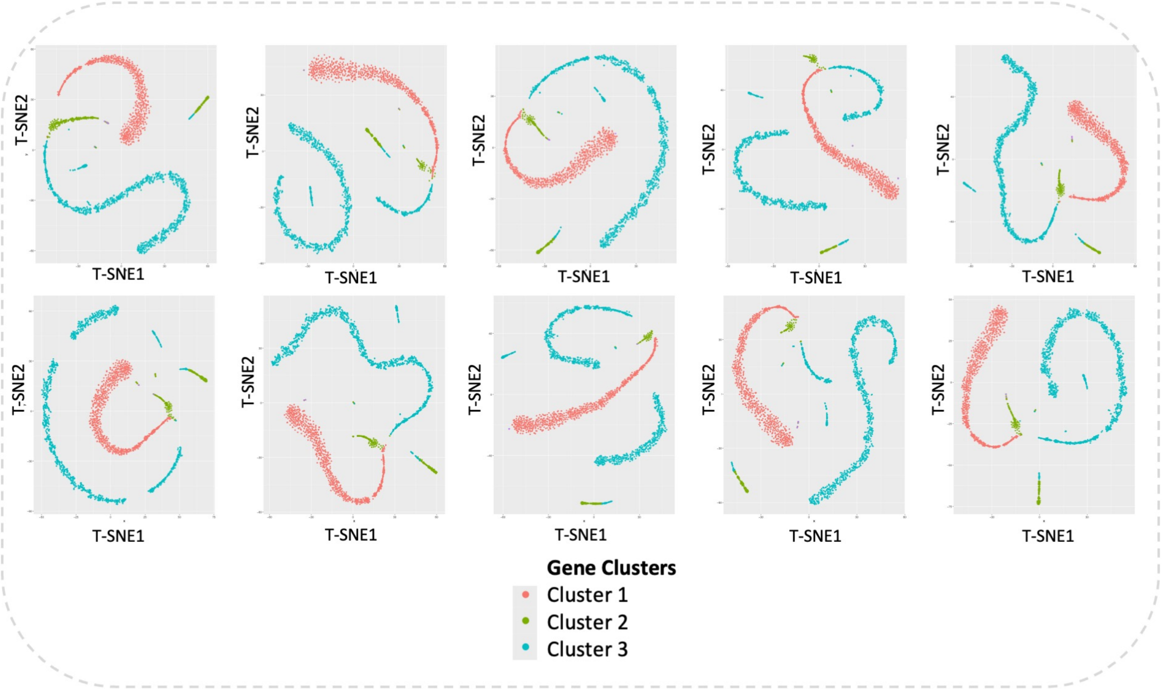


Figure F.2 The t-SNE plots[128] illustrate the separation of domain-specific SVGs by cSVG across 10 synthetic datasets.

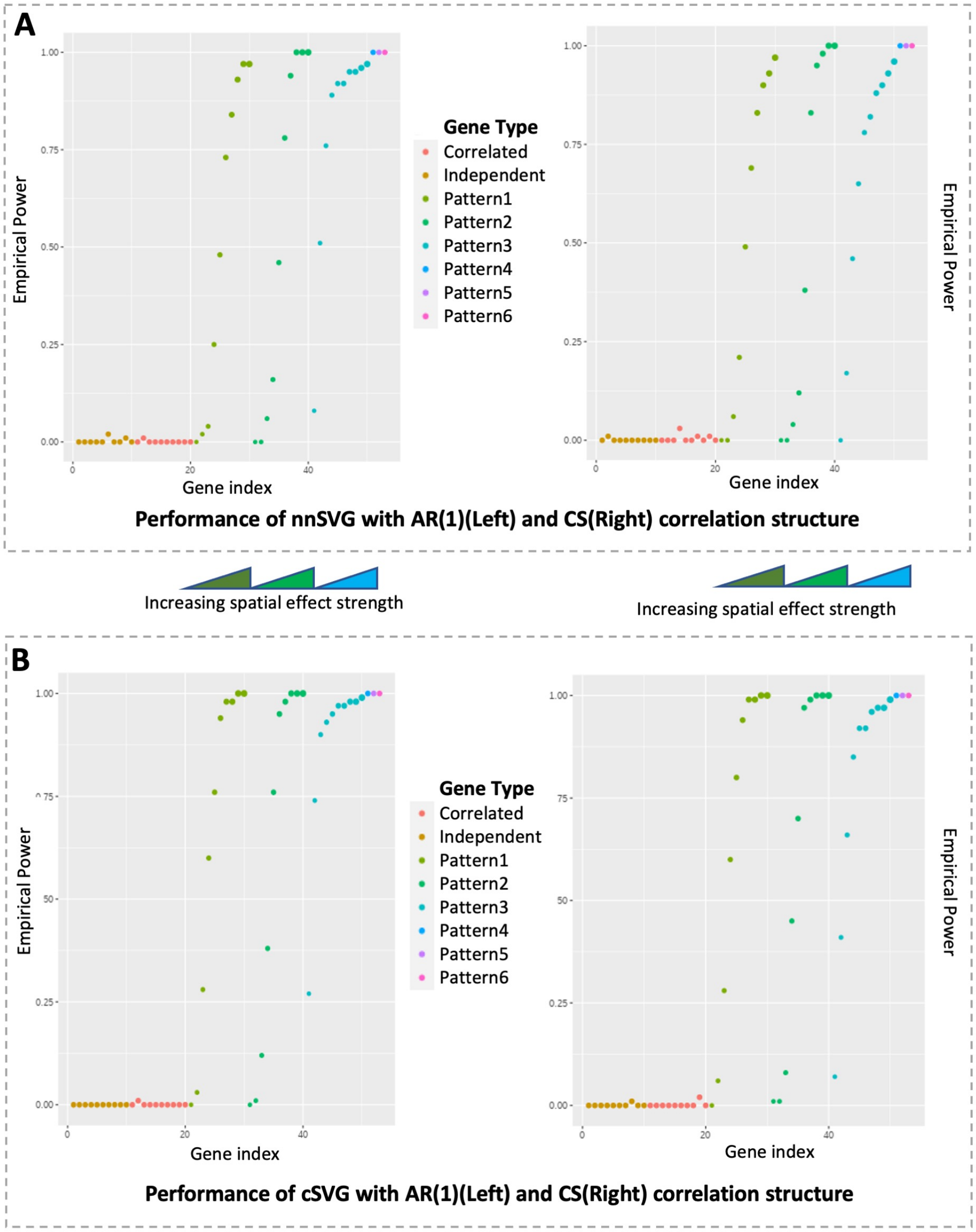


Figure F.3 The performance comparison between A)nnSVG[78] and B)cSVG is conducted for the simulation setup outlined in section 3.1 of the main manuscript. The gene groups are: Independent: Uncorrelated Gene group for genes without any spatial pattern(g1-g10), Correlated: Correlated Gene group for genes without any spatial pattern(g11-g20). Pattern 1-3: Correlated Gene group for genes with spatial pattern 1-3 (g21-g30,g31-g40,g41-g50). Pattern 4-6: Single gene with spatial pattern 4-6(g51-g53) The spatial pattern strength intensifies within each spatial gene group(pattern1-pattern3) as indicated by the triangles between the plots. The empirical power of the SVG detection step is evaluated for simulated datasets with AR(1) (Left) and compound symmetry (CS) (Right) correlation structures within the gene groups.

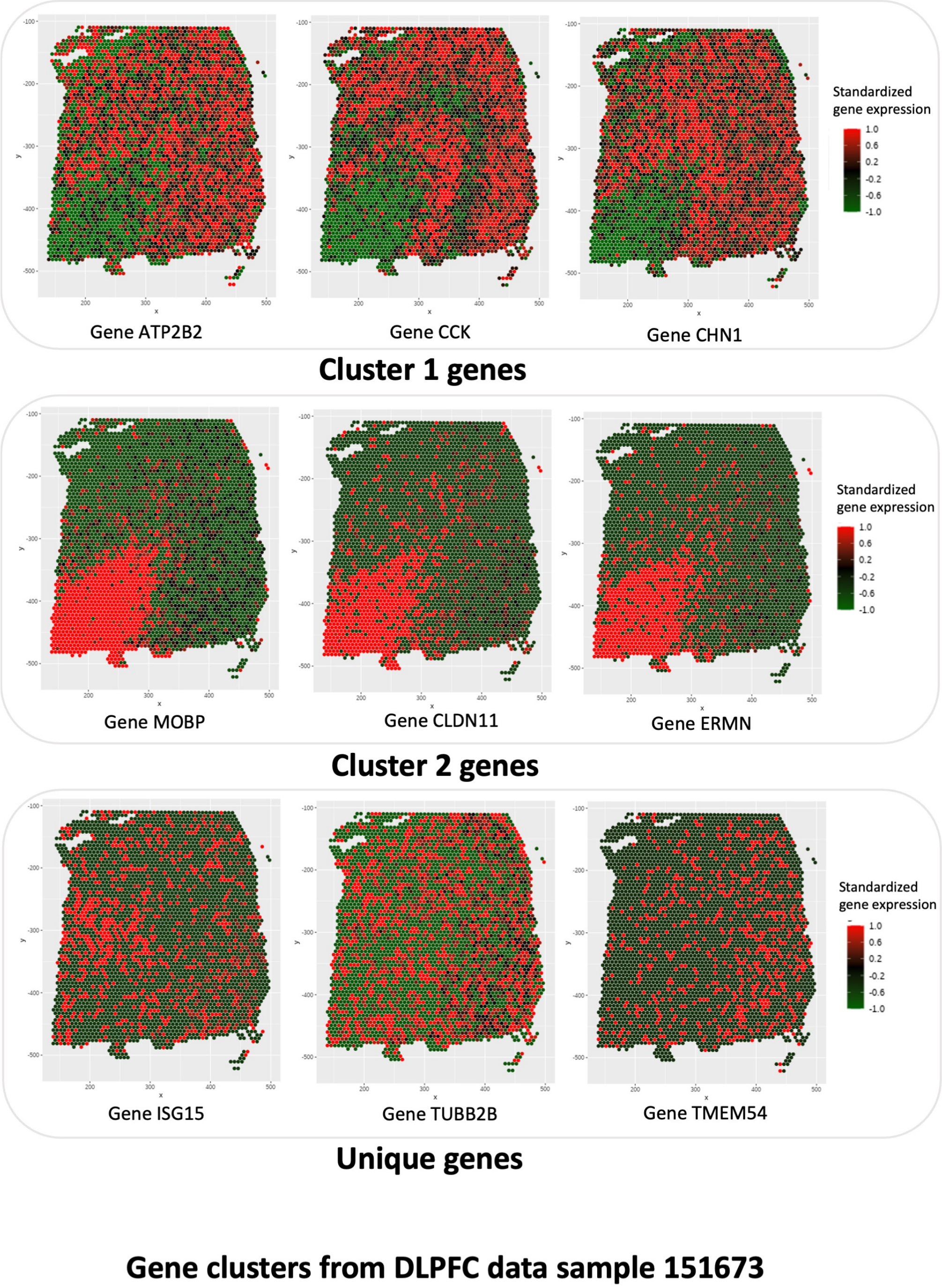


Figure F.4 The SVG-clusters detected from the DLPFC data using cSVG distinctly highlight the disparity between genes in the two main clusters. The representative genes in the second cluster demonstrate overexpression in the white matter region, while those in the first cluster exhibit overexpression in the other six cortex layers. Additionally, the three unique genes in the final row display a slightly different pattern compared to the genes in the main clusters.

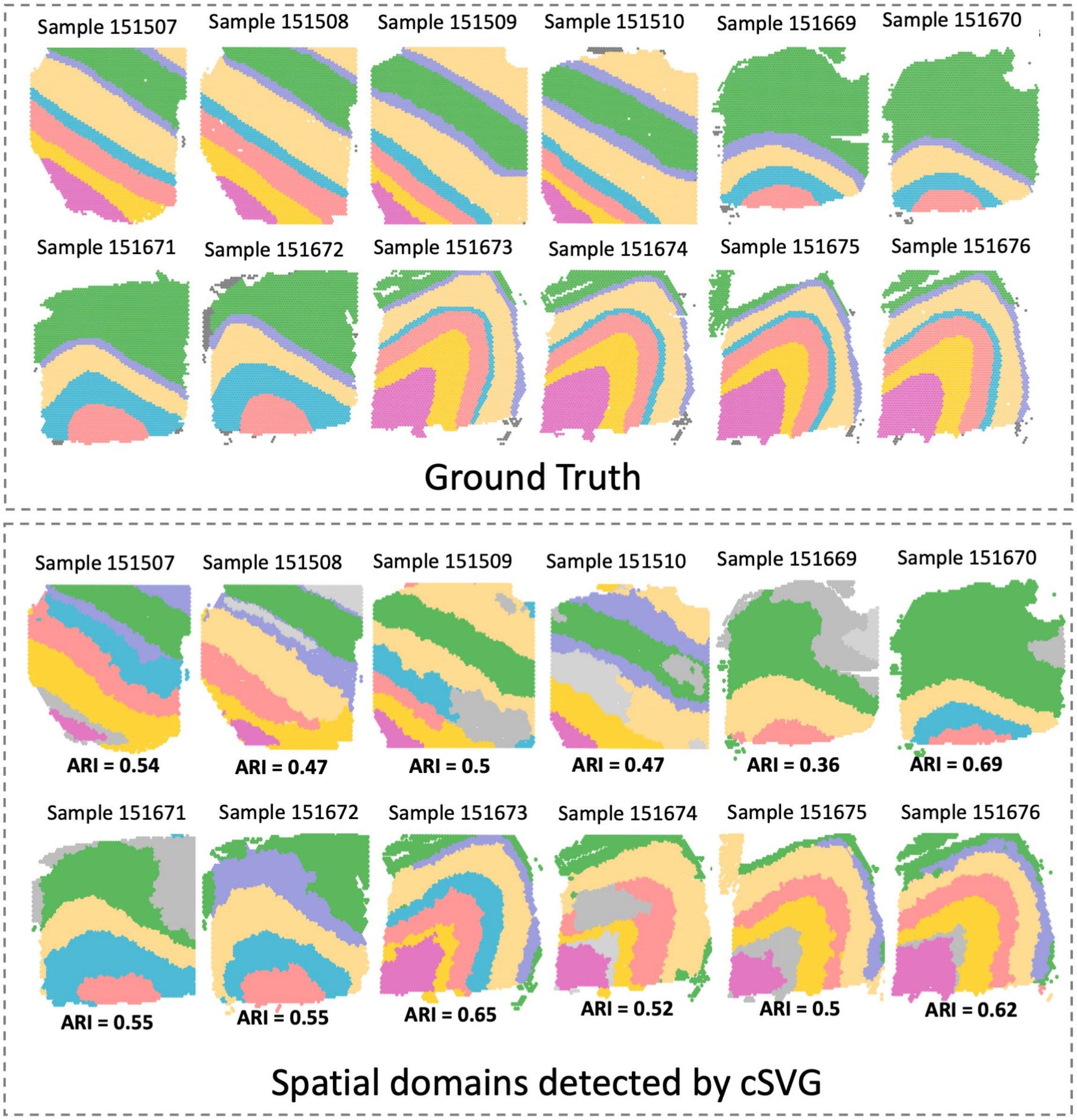


Figure F.5 Annotated and predicted spatial domains for all 12 DLPFC samples using cSVG, alongside ARI scores for each sample indicating prediction accuracy.

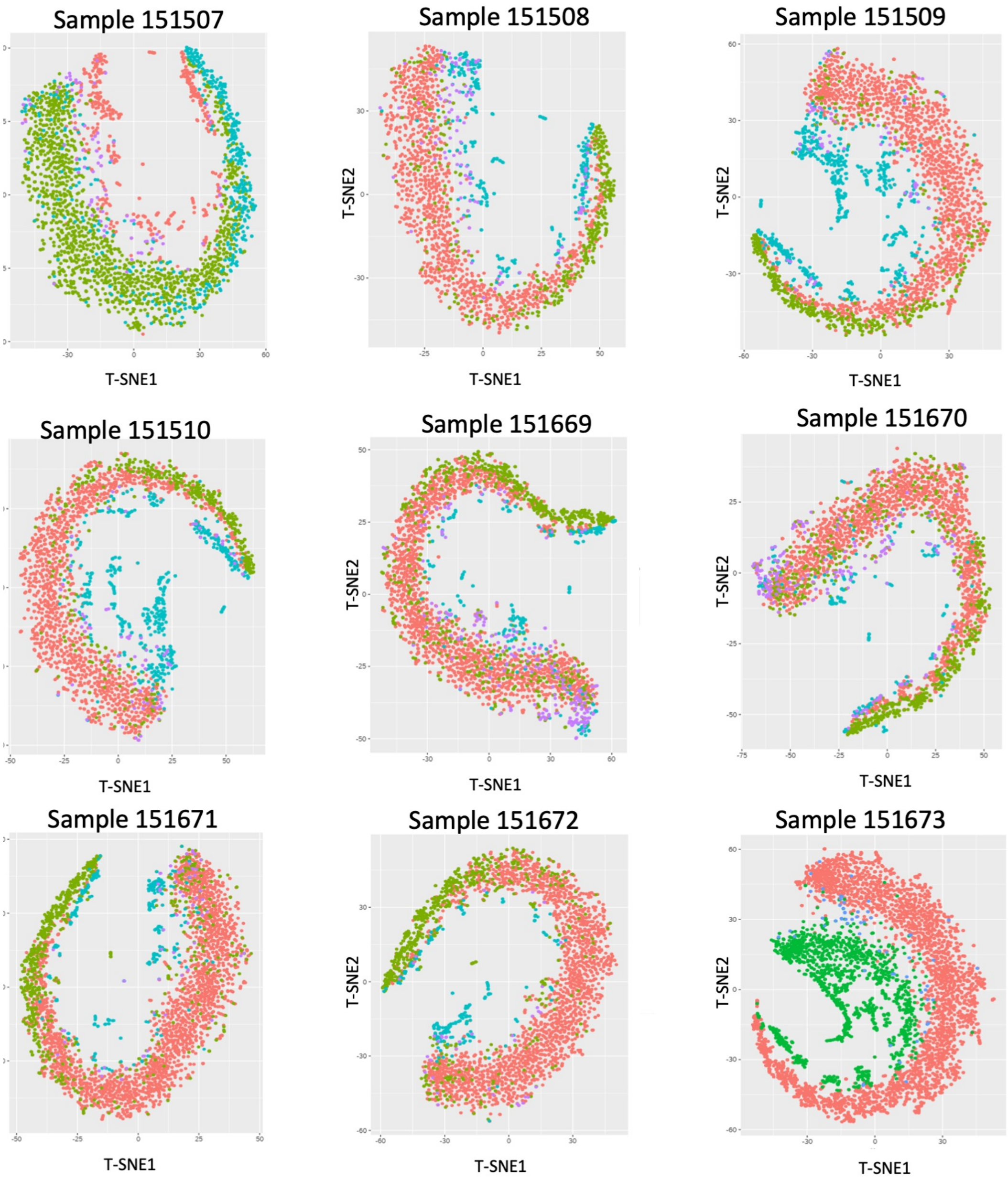
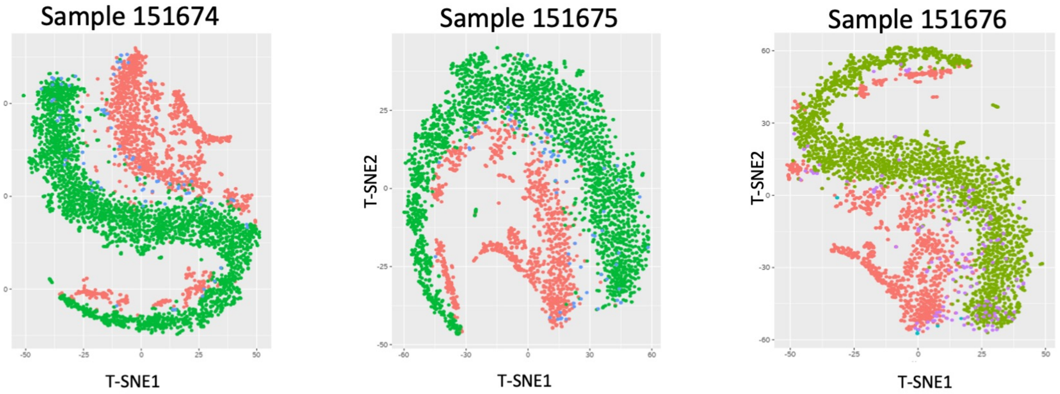
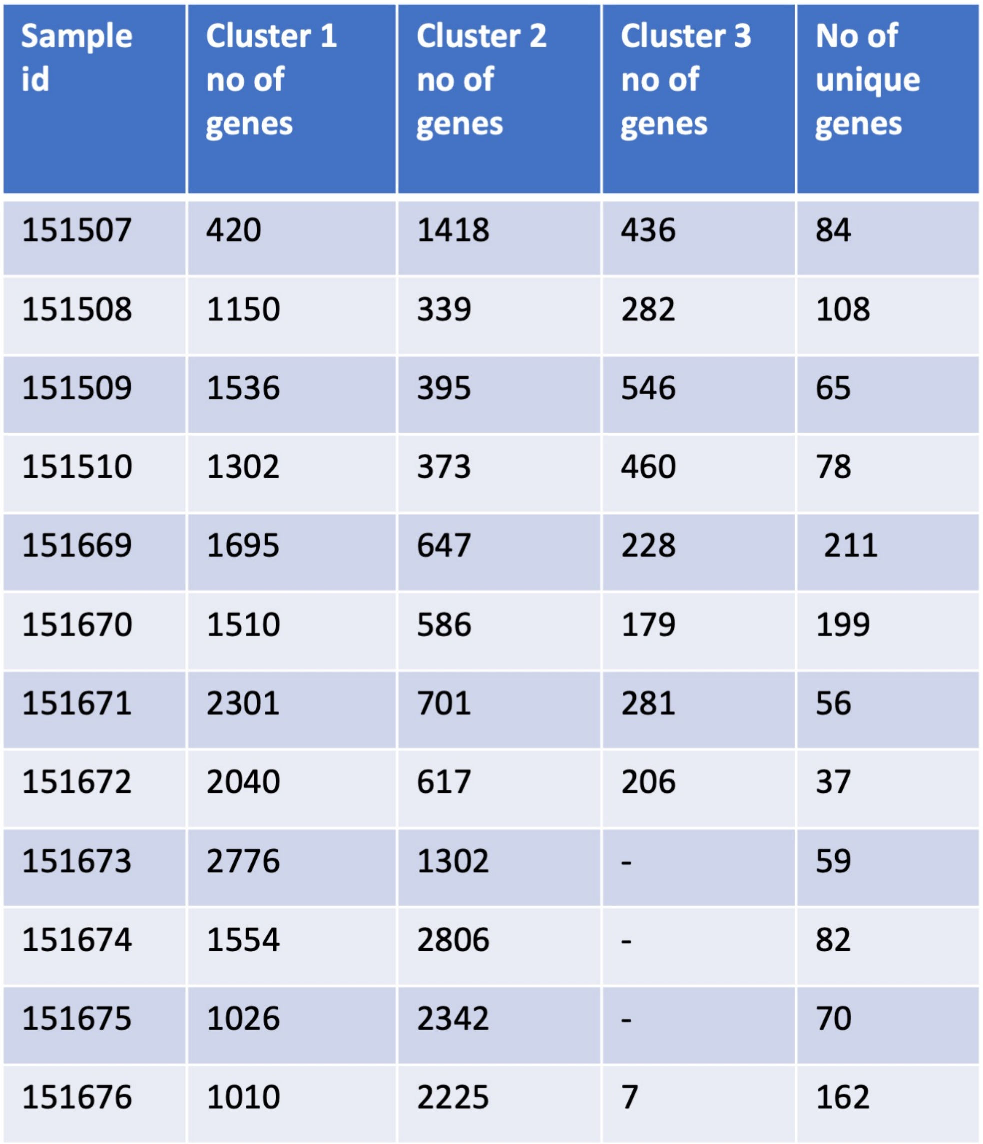


Figure F.6 t-SNE plots illustrating genes from all 12 DLPFC samples, with colors representing SVG cluster labels detected by cSVG. Across the majority of samples, distinct clustering is observed, indicating accurate separation of genes within different clusters.

Figure F.6 (cont'd)





Sample id	Cluster 1 no of genes	Cluster 2 no of genes	Cluster 3 no of genes	No of unique genes
151507	420	1418	436	84
151508	1150	339	282	108
151509	1536	395	546	65
151510	1302	373	460	78
151669	1695	647	228	211
151670	1510	586	179	199
151671	2301	701	281	56
151672	2040	617	206	37
151673	2776	1302	-	59
151674	1554	2806	-	82
151675	1026	2342	-	70
151676	1010	2225	7	162

Figure F.7 Table displays gene-cluster sizes by cSVG for all 12 DLPFC samples.

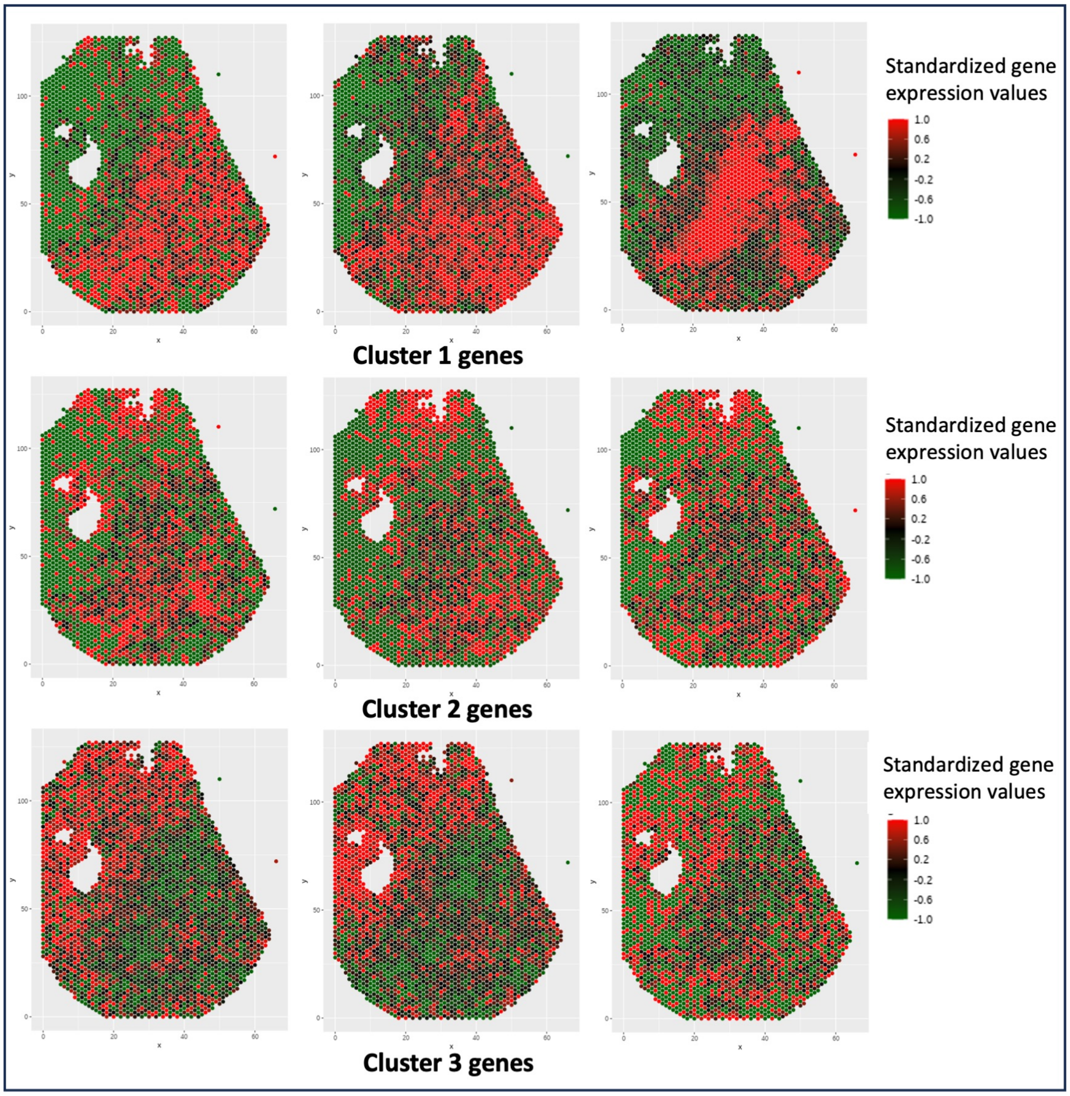


Figure F.8 Analysis of Pancreatic Cancer Data using cSVG unveils three primary SVG clusters. Representative genes from each cluster (Cluster 1, Cluster 2, and Cluster 3) are showcased. The genes from these distinct clusters exhibit overexpression in three different regions.

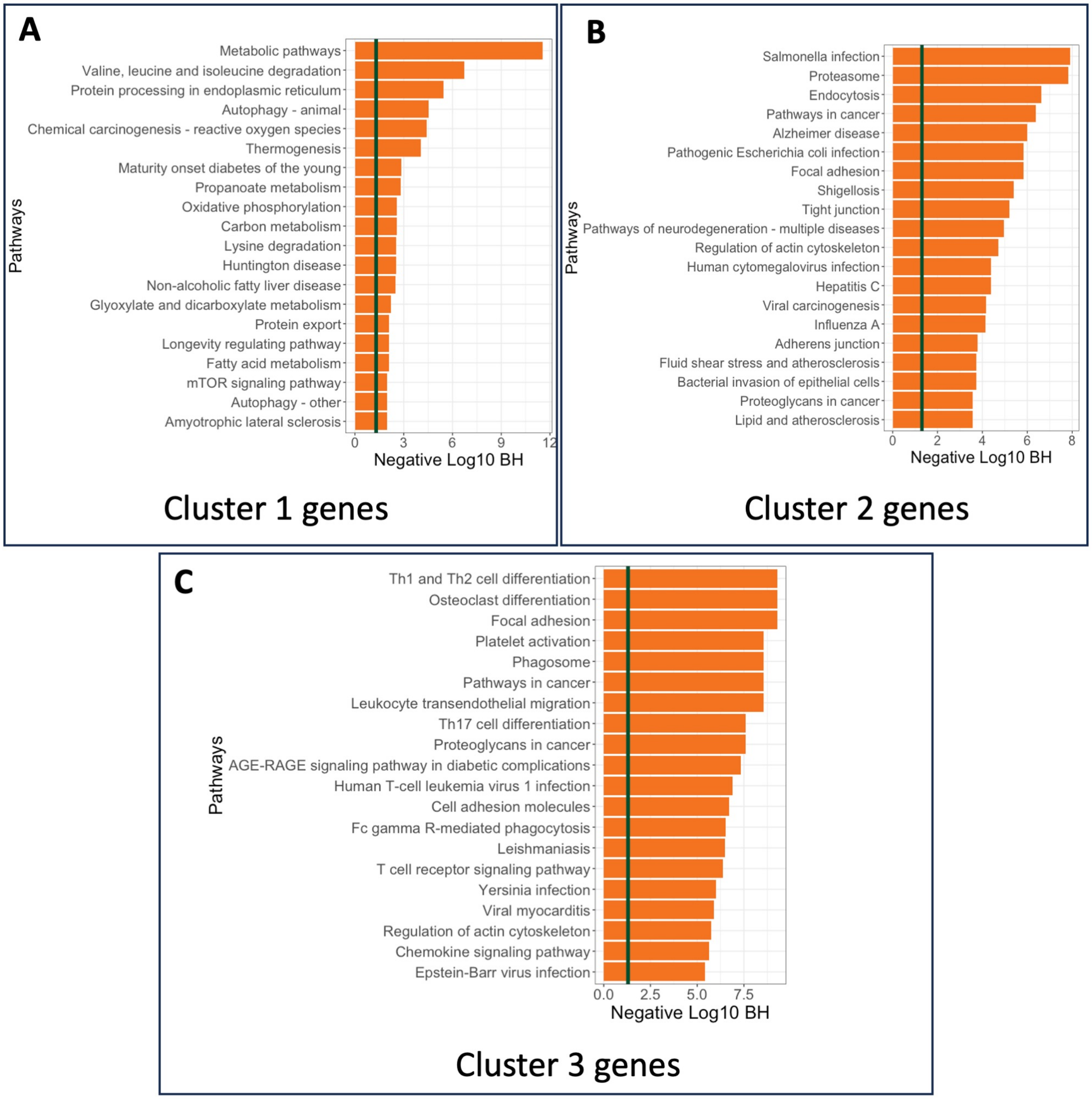


Figure F.9 Analysis of Pancreatic cancer data: Pathway enrichment analysis of genes from A) Cluster 1, B) Cluster 2, and C)Cluster are showcased. As expected, cluster 2 and cluster 3 genes show enrichment in cancer-related pathways.