Method

Kernel-bounded clustering for spatial transcriptomics enables scalable discovery of complex spatial domains

Hang Zhang,[1,2,3] Yi Zhang,[1,2,3] Kai Ming Ting,[1,2] Jie Zhang,[1,2] and Qiuran Zhao[1,2]

[1]National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210023, China; [2]School of Artificial Intelligence, Nanjing University, Nanjing 210023, China

Spatial transcriptomics are a collection of technologies that have enabled characterization of gene expression profiles and spatial information in tissue samples. Existing methods for clustering spatial transcriptomics data have primarily focused on data transformation techniques to represent the data suitably for subsequent clustering analysis, often using an existing clustering algorithm. These methods have limitations in handling complex data characteristics with varying densities, sizes, and shapes (in the transformed space on which clustering is performed), and they have high computational complexity, resulting in unsatisfactory clustering outcomes and slow execution time even with GPUs. Rather than focusing on data transformation techniques, we propose a new clustering algorithm called kernel-bounded clustering (KBC). It has two unique features: (1) It is the first clustering algorithm that employs a distributional kernel to recruit members of a cluster, enabling clusters of varying densities, sizes, and shapes to be discovered, and (2) it is a linear-time clustering algorithm that significantly enhances the speed of clustering analysis, enabling researchers to effectively handle large-scale spatial transcriptomics data sets. We show that (1) KBC works well with a simple data transformation technique called the Weisfeiler–Lehman scheme, and (2) a combination of KBC and the Weisfeiler–Lehman scheme produces good clustering outcomes, and it is faster and easier-to-use than many methods that employ existing clustering algorithms and data transformation techniques.

[Supplemental material is available for this article.]

Spatial transcriptomics (ST) have become a key tool for scientists to profile gene expression and spatial localization information in tissues simultaneously (Marx 2021). They have greatly accelerated biological and biomedical researches (Asp et al. 2020), allowing for the characterization of transcriptional patterns and regulation in tissues, as well as the identification of tissue neighborhoods and local features that contribute to diseases. They are essential tools, particularly in neuroscience, cancer, immunology, and developmental and reproductive biology (Williams et al. 2022). A crucial step in analyzing ST data is to employ an automated clustering algorithm to group spots with similar gene expression patterns and transcriptomic profiles into one cluster and to discover clusters of dissimilar characteristics. The identified transcriptomically similar clusters assist in further characterizing tissue organization and uncovering potential biomarkers or therapeutic targets.

Existing methods for spatial domain detection have primarily focused on data transformation techniques to combine gene expression and spatial information, and to reduce the dimensionality for subsequent clustering. The clustering algorithm employed is either (1) an off-the-shelf clustering algorithm (such as Louvain [Blondel et al. 2008], *k*-means [MacQueen 1967], Walktrap [Pons and Latapy 2005], and Mclust [Scrucca et al. 2016]) or (2) an end-to-end deep learning method that optimizes graph embedding and clustering in a single objective function, such as SpaGCN (Hu et al. 2021).

These existing clustering methods have two key shortcomings. First, they often have high time and space complexities, leading to slow runtime and high memory usage. One example is the time complexities of commonly used clustering algorithms in the field: Louvain has $O(n\log(n))$ (Blondel et al. 2008), Walktrap (Pons and Latapy 2005) has $O(n^2\log(n))$, and the end-to-end deep learning method SpaGCN has $O(n^2)$ time complexity, where n is the input data size. Second, there is still considerable room for improvement in the accuracy of clustering results, especially in handling complex data characteristics with varying densities, sizes, and shapes (in the transformed space on which clustering is performed).

The primary objective of this paper is to introduce a new clustering algorithm that addresses the two key shortcomings of existing methods. We also compare the performance of a simple data transformation technique with more complex, state-of-the-art methods commonly used in the literature.

Results

We show that the proposed kernel-bounded clustering (KBC) algorithm yields much better clustering outcomes than existing clustering algorithms, using a same data transformation technique. KBC has two unique features: (1) It is the first clustering algorithm that employs a distributional kernel to recruit members of a cluster, in which each cluster is defined by a distribution via a kernel mean embedding method, and (2) it is a clustering algorithm that is applicable to ST data with linear time and space complexities. We also show that KBC works well with a simple and fast data transformation technique called the Weisfeiler–Lehman scheme (WL) (Shervashidze et al. 2011). Our ablation studies reveal that (1) KBC is the best clustering algorithm compared with existing clustering algorithms, and (2) although the WL scheme is the second-best method in terms of the clustering performance compared with SpatialPCA (Shang and Zhou 2022), it has the

[3]**These authors contributed equally to this work.**
Corresponding authors: tingkm@nju.edu.cn, zhangj_ai@nju.edu.cn
Article published online before print. Article, supplemental material, and publication date are at https://www.genome.org/cgi/doi/10.1101/gr.278983.124. Freely available online through the *Genome Research* Open Access option.

runtime advantage, unmatched by any other existing data transformation technique.

Method overview

Figure 1 shows the workflow of performing clustering on an ST data set using the proposed KBC algorithm. The spatial location and gene expression information are preprocessed separately in order to construct the adjacency matrix and feature matrix, respectively. These matrices are then integrated into a graph. Figure 1A illustrates the detailed preprocessing steps. The two key components in subsequent procedures are graph embedding and clustering, as shown in Figure 1B. The former aims to embed a graph of data set D into a low-dimensional space such that an existing clustering algorithm can be applied to discover clusters within D.

As the focus of this paper is clustering, we highlight the distinctive features of KBC in comparison with existing clustering algorithms such as k-means (MacQueen 1967) and Louvain (Blondel et al. 2008). The first distinct feature of KBC has two key aspects. First, KBC treats points in a cluster as independent and identically distributed (iid) samples from an unknown distribution. Second, KBC employs a distributional kernel to represent and grow a cluster. The first aspect forms the foundation of the entire algorithm, whereas the second aspect is encapsulated in the second step of KBC. Although some existing clustering algorithms such as density-based methods and Gaussian mixture models (GMMs) have considered clusters as distributions, they must employ either a density estimator or a mixture model to estimate the distribution. This is the source of the estimation error and high time complexity. KBC employs neither of them but a distributional kernel that does not need to estimate the distributions at all.

The first step of KBC is to identify k initial clusters, in which each cluster is treated as generated from an unknown distribution. They are discovered from a subset of the (embedded) data set D. The second step of KBC assigns each unassigned point in D to the most similar distribution (estimated from the points in an initial cluster), based on a distributional kernel. This gives rise to the first distinctive feature of KBC, namely, *KBC discovers clusters of arbitrary shapes, sizes and densities*, which are congruous to the distributions of the clusters in D. The details of the procedure are given in Algorithm 1 in the Methods.

Existing clustering algorithms have limitations in their ability to detect certain types of clusters. The limitation of k-means clustering is well studied; that is, it can only find clusters of globular shapes because each cluster is represented with an average vector (Aggarwal 2015). Although density-based methods (Ester et al. 1996; Rodriguez and Laio 2014) can detect arbitrary-shaped clusters, they have difficulties discovering clusters of varied densities (Zhu et al. 2016) and clusters of uniform distribution (in which a density estimator often produces multiple peaks in a cluster of uniform distribution, owing to estimation errors) (Zhu et al. 2022). Louvain, a commonly used clustering algorithm in ST, has similar issues because it is also based on density. Louvain is a community detection method that aims to find communities (i.e., clusters) in a network or graph. It maximizes modularity, which is a measure of the relative density of edges within each community compared with the density of edges between communities. The density of edges within a community is defined as the sum of edge weights between nodes that belong to the community. A recent study (Traag et al. 2019) shows that Louvain can produce arbitrarily badly connected clusters (or communities in a graph). Clustering based on GMM (Dempster et al. 1977; Traag et al. 1996) assumes that each cluster is a mixture of Gaussian distributions. This requires modeling the parameters of the Gaussian distributions and the weights of their mixtures, and accurate estimations often require a substantial amount of data. However, many ST data sets may not have sufficient data for GMMs to perform accurate estimations (Lyu et al. 2021). As a result, this often leads to suboptimal clustering outcomes. Mclust (Scrucca et al. 2016) is a typical GMM-based clustering algorithm used in the literature on ST (Zhao et al. 2021; Dong and Zhang 2022). Walktrap (Pons and Latapy 2005) relies on random walks in a network in order to identify communities in the network. The intuition is that random walks on a network tend to get "trapped" into specific communities, characterized by densely interconnected nodes, whereas the connections between different communities tend to

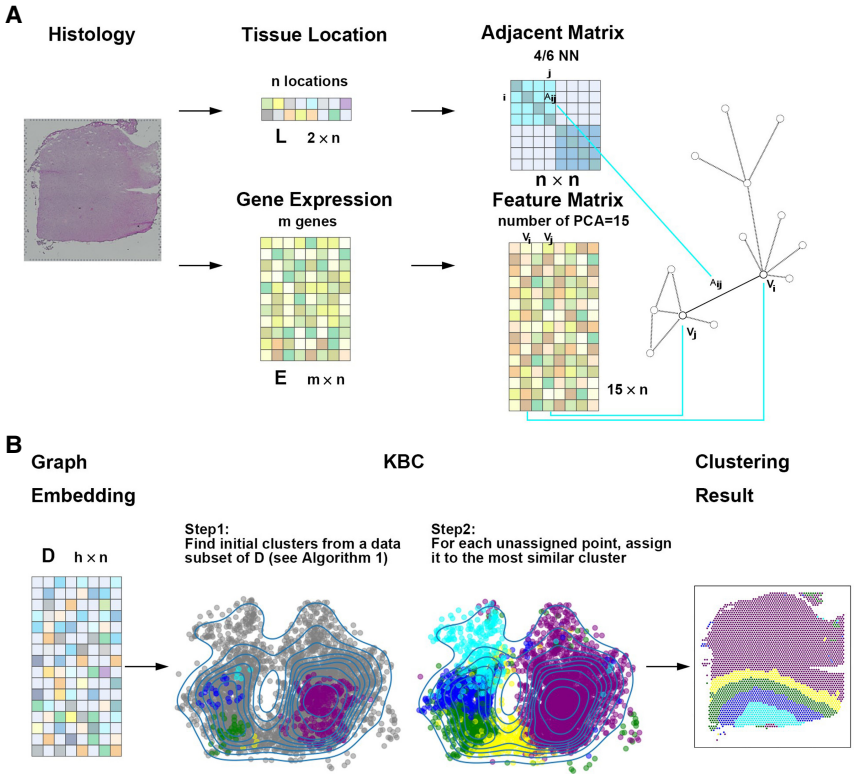

Figure 1. The workflow of clustering analysis using KBC. (*A*) Beginning with a spatial transcriptomics (ST) data set, the spatial information L and the gene expression information E are integrated to produce a graph that contains both cell location and gene expression information. (*B*) A graph-embedding scheme converts a graph into a vector representation, as shown on the *left*, which is ready to be used for clustering. The illustration shows the two steps of the proposed KBC algorithm.

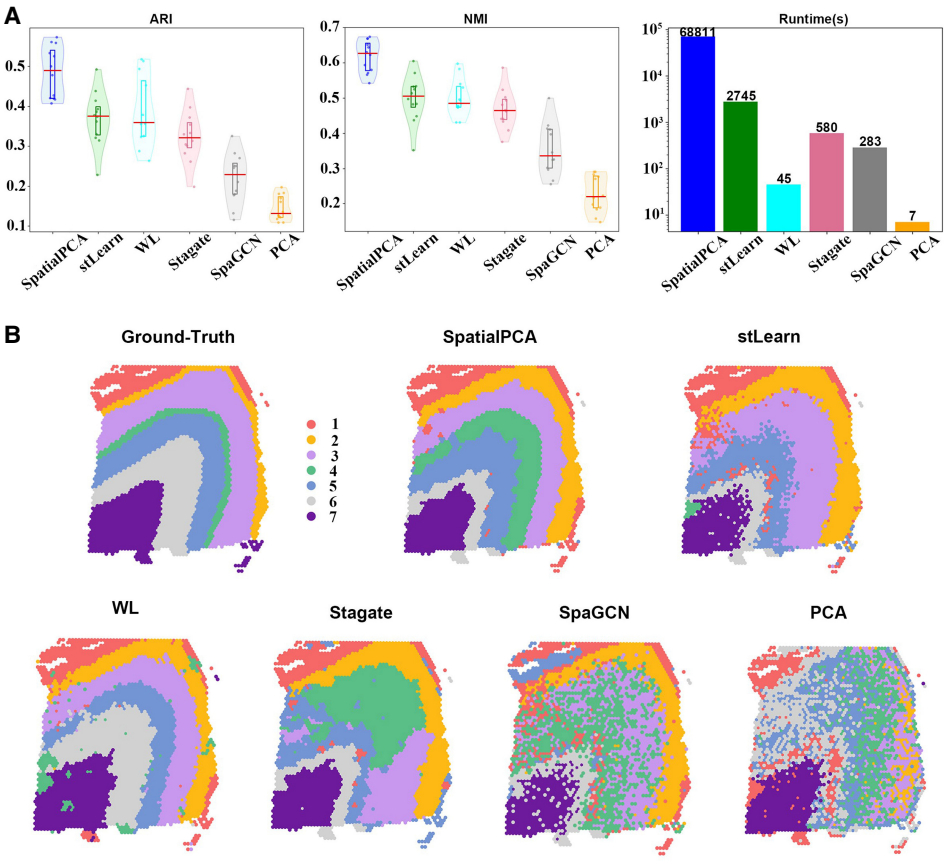

Figure 2. First ablation study. Comparing different data transformation methods using the same *k*-means clustering. (*A*) The violin plots show the ARI and NMI results of all 12 slices of DLPFC. The runtimes are shown in a bar chart. (*B*) The detailed clustering results of different embedding methods together with the ground-truth labels are shown for the slice 151673. Here, SpatialPCA, WL, and PCA ran on CPU only. Other methods ran on GPU.

be sparse. Walktrap defines a distance by using the properties of random walks (Aggarwal 2015) and then uses an agglomerative hierarchical clustering (AHC) algorithm based on Ward's method (Ward 1963) to perform the final clustering. As AHC is known to be sensitive to the merging criterion employed and unable to discover clusters of varied densities (Han et al. 2022), Walktrap has the same issue as other AHC-based clustering algorithms. In essence, none of the existing clustering algorithms are able to discover clusters of arbitrary shapes, sizes, and densities, and these kind of clusters often exists in ST data sets.

The second distinctive feature of KBC is that it is the first linear time-and-space-complexity clustering algorithm. None of the existing clustering algorithms have this feature. Even the fastest *k*-means has superpolynomial time complexity (Arthur and Vassilvitskii 2006), although it often exhibits linear time in small data sets. All the other clustering algorithms mentioned above have at least quadratic time complexity.

In addition, we introduce the WL graph embedding (Weisfeiler and Lehman 1968; Shervashidze et al. 2011; Togninalli et al. 2019) in the treatment of ST data for the first time. It has been theoretically proved that graph neural networks (GNNs) cannot be more powerful than WL in terms of distinguishing nonisomorphic (sub)graphs (Morris et al. 2019). This result holds for a wide range of GNN architectures and all possible parameter choices for them. Moreover, a straightforward and efficient implementation of WL is available (Xu et al. 2021), offering computational advantages and running orders of magnitude faster than existing GNN-based methods. Specifically, we demonstrate that WL produces clustering outcomes nearly as effective as SpatialPCA (Shang and Zhou 2022), the leading data transformation method in ST, while executing much faster.

Ablation studies

We aim to ascertain the importance of the two components in clustering ST data, namely, data transformation and clustering. To achieve this aim, we conduct two experiments to assess:

1. The relative performance of different data transformation methods, while performing the clustering with the same *k*-means clustering algorithm. The six methods under assessment are WL (Togninalli et al. 2019), SpatialPCA (Shang and Zhou 2022), Stagate (Dong and Zhang 2022), SpaGCN (Hu et al. 2021), stLearn (Pham et al. 2023), and PCA (Maćkiewicz and Ratajczak 1993).
2. The relative performance of different clustering algorithms, while using the same best data transformation method determined in the first experiment. The seven clustering algorithms under investigated are the proposed KBC, Walktrap (Pons and Latapy 2005), Mclust (Scrucca et al. 2016), *k*-means (MacQueen 1967), SpaGCN (Hu et al. 2021), BayesSpace (Zhao et al. 2021), and Louvain (Blondel et al. 2008).

The performance is assessed in terms of (1) the clustering outcome measured in adjusted Rand index (ARI) (Yeung and Ruzzo 2001) and normalized mutual information (NMI) (Cover 1999) and (2) the runtime on the same machine with an Intel i7-12700k and a NVIDIA RTX 3090 graphics card. We report the average ARI or NMI over 10 trials on each data set, as each of these methods has some form of randomization in the procedure. The LIBD human dorsolateral prefrontal cortex (DLPFC) (Maynard et al. 2021) data set, which is available at spatialLIBD and consists of 12 slices, is used in both experiments.

Figure 2 shows the results of the first ablation study. SpatialPCA yields the best clustering performance as shown in both violin plots of ARI and NMI, but it is the slowest compared with the other methods. SpatialPCA builds upon the probabilistic PCA, incorporates localization information as additional input, and uses a kernel matrix to explicitly model the spatial correlation structure across tissue locations. SpatialPCA uses SPARK (Sun et al. 2020) or SPARK-X (Zhu et al. 2021), depending on the size of the data set, to select the spatially variable genes (SVGs). In many slices, it learns the best feature representation. PCA, which is a cut-down version of SpatialPCA without the spatial information, produces the worst ARI/NMI result, but it is the fastest. This is not surprising because the spatial information is not used. WL, stLearn, and Stagate are the second-best methods with comparable ARI/NMI results, but WL runs at least one order of magnitude faster than the other two, even though WL uses the CPU and the other two utilize the GPU. In addition, WL runs three orders of magnitude faster than SpatialPCA. Stagate constructs a spatial neighbor network (SNN) based on the relative spatial locations of spots. Then, Stagate learns a low-dimensional embedding with spatial information and gene expressions via a graph attention auto-encoder. It can be observed that Stagate's embedding is not very good on this data set in Figure 2, A and B. The use of auto-encoder provides no advantage because it is computationally expensive and sensitive to parameter settings. stLearn employs a spatial morphological gene expression (SME) normalization method to learn an embedding. Out of all methods under comparison, stLearn is the only method that exploits all three types of information made available in spatially resolved transcriptomics (SRT), namely, spatial location, tissue morphology, and gene expression measurements. Yet, stLearn is not as effective as SpatialPCA and is only as good as WL. SpaGCN is the second-worst performer in terms of ARI/NMI, and it has comparable runtime (in the same order) with stLearn and Stagate. After calculating adjacent matrix, SpaGCN utilizes a graph convolution layer to aggregate gene expressions from neighboring spots. Note that GCN can use two layers only in practice, because using more layers can even hurt its performance (Zhang et al. 2019a). A recent study (Zhang et al. 2019b) shows that clustering can achieve good results without the use of graph convolution. Another interesting thing is that WL has been proven to be the best achievable by a GNN (Morris et al. 2019).

Figure 3, A and B, shows the results of the second ablation study, comparing seven clustering algorithms (using SpatialPCA as the data transformation method). The violin plot reveals that KBC is the best performer in terms of ARI and NMI. Walktrap, Mclust, and *k*-means are the second-best clustering algorithms having comparable result. SpaGCN and BayesSpace fall into the next tier, whereas Louvain is the worst performer.

SpaGCN uses either Louvain or *k*-means to produce a set of initial clusters. As the final clustering outcome of SpaGCN heavily relies on the quality of the initial clusters, when they do not represent the clusters well, they often lead to poor final clustering outcomes. The issues with Louvain and *k*-means have been presented in the section Method Overview.

BayesSpace is a Bayesian statistical method that encourages neighboring spots to belong to the same cluster by introducing a spatial prior into the spatial neighborhood structure. BayesSpace performs the Bayesian clustering after obtaining initial clusters using either *k*-means or Mclust. If ground-truth clusters overlap, which often occurs in ST data, *k*-means or Mclust will produce poor (initial) clusters, leading to the poor final clustering of BayesSpace.

The fastest clustering algorithms are *k*-means and KBC; their runtimes are in the same order of magnitude. Walktrap, Mclust, SpaGCN, and Louvain are the second fastest, but they are one order of magnitude slower than *k*-means and KBC. BayesSpace is the slowest, which is two orders of magnitude slower than *k*-means and KBC.

Summary of the two ablation studies

In a nutshell, the two components (i.e., data transformation and clustering) are equally important. Although significant advancements have been made in data transformation, clustering algorithms remain a less researched area in the ST domain. As articulated in the Introduction, all known works have utilized existing clustering algorithms.

Although SpatialPCA is the best data transformation method in our study, it is the slowest. Its contender is the proposed WL scheme, which has slightly weaker embedding outcome but runs three orders of magnitude faster. Note that the relative clustering outcome between SpatialPCA and WL is on this particular data set only. We will see later that the outcome is reversed on other data sets.

Among all clustering algorithms examined, the proposed clustering KBC is a clear winner in terms of both clustering outcome and runtime.

The ability to find clusters of varied densities and overlapping clusters

Here we examine the ability of a clustering algorithm to find clusters of arbitrary shapes, sizes, and densities using simple two-dimensional data sets, without the need to use a data transformation method. The data sets are those used previously to investigate the fundamental problems of spectral clustering (Nadler and Galun 2006). The first (3Gaussians) data set, shown in Figure 4C, is composed of one sparse Gaussian distribution and two dense Gaussian distributions, having 300 points per cluster. The second (StripC) data set, shown in Figure 4D, comprises a thin strip cluster and a circle with 700 points per cluster. The first examines the ability of a clustering algorithm to separate clusters of varied densities; the second assesses the ability to separate two overlapping clusters. Note that these are not data sets trying to simulate the characteristic of ST data. They are designed to reveal an algorithm's fundamental clustering limitations if it could not successfully cluster these simple data sets.

The clustering outcomes of five clustering algorithms are shown in Figure 4, A and B. On the 3Gaussians data set, KBC is the best performer, having ARI = 0.97, matching almost perfectly with the ground-truth clusters. Louvain, Waltrap, and *k*-means are the second-best performers as they have overextended the middle dense cluster to include neighboring points of the sparse cluster. Mclust has the same issue but with an additional shortcoming; namely, the two dense clusters have been merged into one. It is the only clustering algorithm that made this mistake.

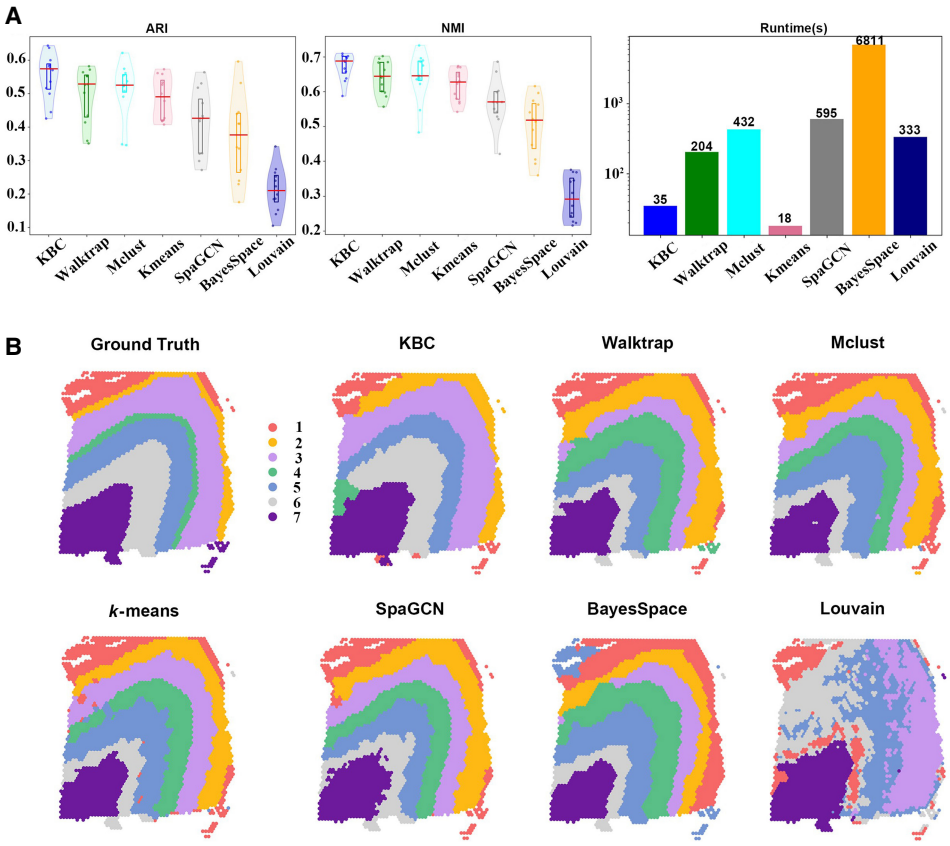

Figure 3. Second ablation study. Comparing different clustering algorithms using the same SpatialPCA embedding. (*A*) The violin plots show the ARI and NMI results of all 12 slices of DLPFC of the seven clustering algorithms. The runtimes are showed in a bar chart. (*B*) The detailed clustering results of different clustering methods are shown for the slice 151673. Only SpaGCN ran on GPU; others ran on CPU.

On the StripC data set, KBC also has the best result with ARI = 0.88, which best matches the ground-truth clusters. Louvain, Walktrap, and *k*-means have the same handicap; that is, each separates the thin strip cluster into two halves, although at different locations on the strip cluster. Mclust has done the reverse; namely, it identifies correctly the strip cluster but includes parts of circle to be members of the strip cluster.

These poor clustering outcomes of the existing clustering algorithms are reminiscent to those of spectral clustering, having fundamental limitations identified previously (Nadler and Galun 2006). In short, these clustering algorithms have the fundamental limitations to discover clusters of arbitrary shapes, sizes, and densities.

KBC successfully discovers clusters of arbitrary shapes, sizes, and densities on an ST data set, but others fail

In addition to the clustering capability of KBC shown thus far, we demonstrate here the ability of KBC to find clusters of arbitrary shapes, sizes, and densities on an ST data set, whereas others fail to do so.

Figure 5 compares the clustering outcomes of four clustering algorithms on the DLPFC data set. The density contour map, shown in the second row (which is the same for all plots), provides interesting information. First, there are two large clusters on either side of a deep density valley in the middle. In the ground-truth column, the right (purple) cluster has one clear density peak, and the left (blue and green) cluster has two clear density peaks, where the former is much denser than the latter. The light blue and yellow regions are the lowest-density regions (outside the white region, which has no data).

There are three interesting observations:

- KBC is the only clustering algorithm that can correctly identify the densest purple cluster as one single cluster. All the other algorithms split this cluster into multiple smaller clusters.
- KBC also correctly splits the second densest two-peak cluster into two. SpatialPCA (with Walktrap) and stLearn (*k*-means) identify it as one cluster only.
- KBC correctly identifies the region in-between the two densest clusters as a single low-density (yellow) cluster and as another low-density (light-blue) cluster at the edge of the dense (blue) cluster. Although Walktrap (with SpatialPCA) and *k*-means (with stLearn) can also correctly identify the light-blue cluster, they did poorly on the second-densest cluster and the low-density region in-between the two densest clusters. Mclust (with Stagate) is the worst performer in these regions, in which each of these low-density regions is merged with a neighboring dense cluster.

KBC is a linear time-and-space-complexity clustering algorithm

The time complexities of KBC and *k*-means (Hartigan and Wong 1979) are $O(n)$, where n is the data set size. Other clustering algorithms such as Mclust (Scrucca et al. 2016), SpaGCN (Hu et al.

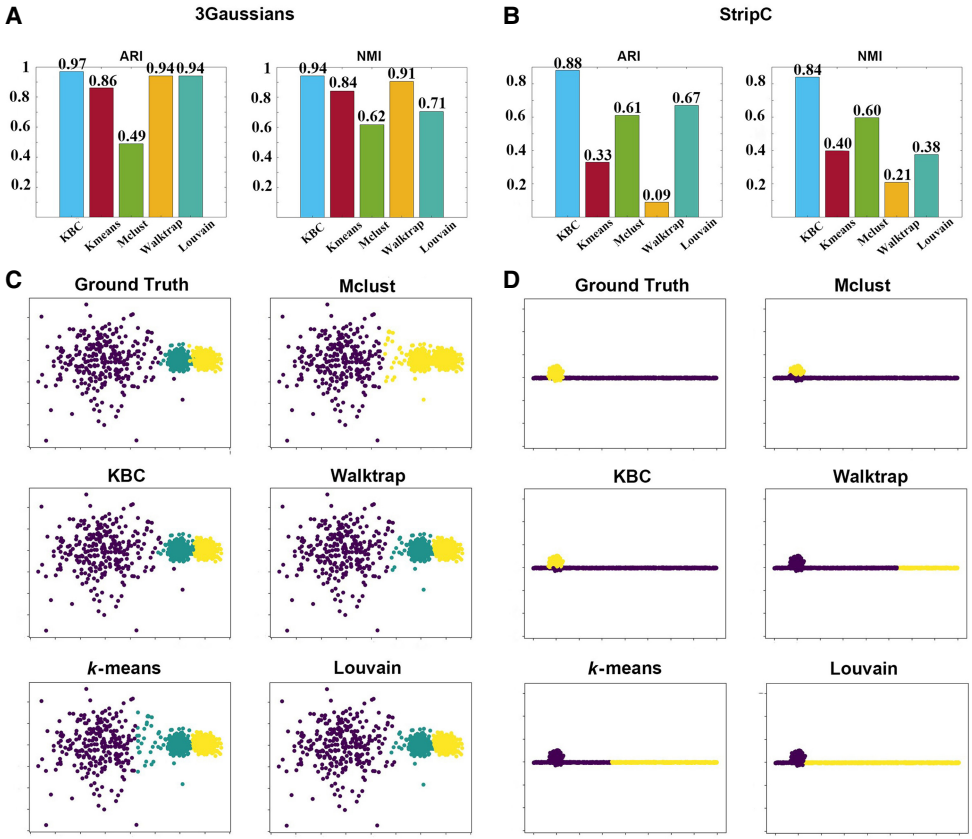

Figure 4. Examining the ability to find clusters of varied densities and overlapping clusters on two synthetic data sets: 3Gaussians and StripC. (*A*) The bar charts (in terms of ARI and NMI) of the five clustering methods on 3Gaussians. (*B*) The bar charts of the five clustering methods on StripC. (*C*) The ground truth and the clustering results of the five clustering methods on 3Gaussians. (*D*) The ground truth and the clustering results of the five clustering methods on StripC.

2021), and BayesSpace (Zhao et al. 2021) have $O(n^2)$ time complexity; Louvain has $O(n\log(n))$ (Blondel et al. 2008); and Walktrap (Pons and Latapy 2005) has $O(n^2\log(n))$. The algorithms with at least quadratic time complexity are unable to deal with large-scale data sets.

We conduct a scaleup test to examine the scalability of seven clustering algorithms. The results of the scaleup test on the seven clustering algorithms are shown in Figure 6. In terms of runtime, BayesSpace is the slowest which, is four orders of magnitude slower than the fastest KBC and *k*-means at data size $10^{4.5}$. The other four

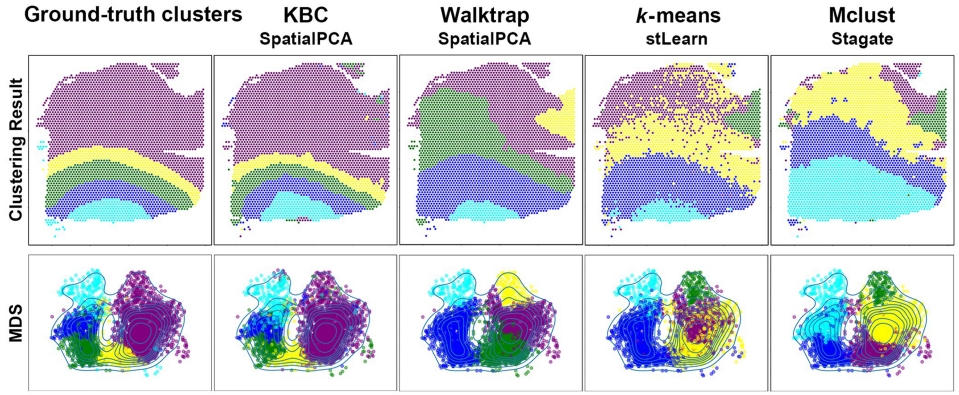

Figure 5. Clustering outcomes on the density contour map created using multidimensional scaling (MDS) (Torgerson 1952). MDS reduces the number of dimensions of the features derived from SpatialPCA (identified to be the best data transformation method previously). The density is estimated using kernel density estimation (Scott 2015) on the space of the MDS reduced dimensions. The data transformation methods used are SpatialPCA, stLearn, and Stagate, and the clustering methods are as employed in their respective papers (Dong and Zhang 2022; Shang and Zhou 2022; Pham et al. 2023), except the proposed KBC.

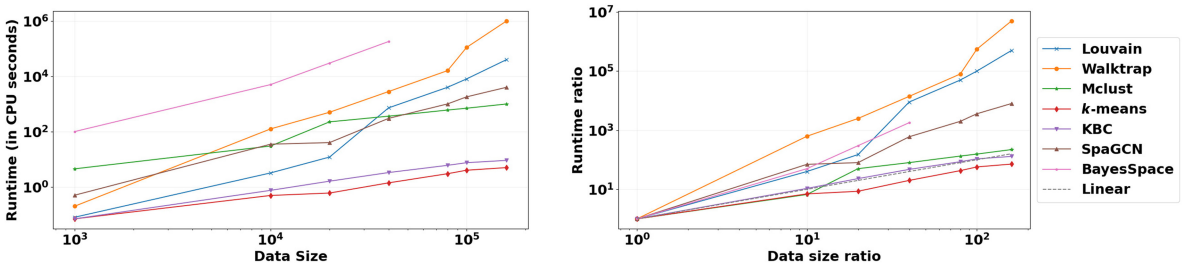

Figure 6. Scaleup test result for different clustering algorithms on the Slide-seq V2 mouse hippocampus data set (Stickels et al. 2021). This data set facilitates the creation of increasing larger subsets for the test. The data sizes range from 1000 to 160,000. We reduce the dimensionality of the data set using principal component analysis (PCA) and retain the top 20 principal components, which capture the majority of the variance in the data set. This is used for all algorithms. The data set size has 1000 points at data size ratio = 1. BayesSpace has no results on larger data sets because it took >48 h. Note that SpatialPCA and stLearn employ Walktrap and *k*-means/Louvain, respectively, as their clustering algorithms. Only SpaGCN's runtime is in GPU seconds. Note that the linear time has a gradient of one in the runtime ratio plot (shown by the line labeled as linear). Those runtimes that are worse than linear have a higher gradient.

clustering is at least two orders of magnitude slower. The gap is getting worse as the data size increases (except Mclust). In terms of runtime ratio, KBC and *k*-means have linear time, and all other methods (except Mclust) have at least quadratic time. Note that SpaGCN ran on GPU, and other methods ran on CPU. Yet, SpaGCN still ran significantly slower than KBC and *k*-means.

These results are consistent with the time complexities of the algorithms shown earlier. There are two exceptions. First, Mclust runs in linear time, instead of quadratic time stated in the original paper (Scrucca et al. 2016). An examination of the code indicates that the speedup could be caused by the use of a kind of search (default hard coded setting), which limits to a fixed number of models examined in order to reduce the runtime. Second, Louvain runs significantly slower than $O(n\log(n))$, and we suspect that the time complexity stated in their paper is incorrect.

The space complexities of KBC and *k*-means are $O(n)$, and the space complexity of Walktrap is $O(n^2)$. The space complexities of Mclust, Louvain, BayesSpace, and SpaGCN are not specified in their papers.

We have used the names of the existing methods to refer to the data transformation methods only thus far. In the rest of this paper, the name of each existing method refers to both the data transformation method and the clustering algorithm employed in their individual paper. KBC uses the WL embedding in the following sections.

Application of KBC to the HER2 tumor data

The overexpression of *ERBB2* (also known as human epidermal growth factor receptor 2 [HER2]) on tumor cells defines the major subtypes of breast cancer. HER2-positive tumors are typically caused by the amplification of a domain on Chromosome 17 (cytogenetic band Chr17q12) containing the *ERBB2* gene.

The source of HER2-positive breast tumor data collected from the ST platform (Andersson et al. 2021) has a data set containing 36 samples, labeled from patient A to patient H, each of which has different numbers of sections. Following the method of Shang and Zhou (2022), we have selected eight patient sections (A1–H1) as our experimental data for clustering analysis. Figure 7, A–C, shows the overall results obtained from different methods.

Following the method of Shang and Zhou (2022), we show the detailed results of section H1 in Figure 7, D and E. This particular section includes approximately 10,000 genes and 600 cells collected from ST. The HER2 tumor data set has been examined and annotated by a pathologist, resulting in seven spatial domains in section H1, as shown in Figure 7D. The focus is on two domains: The orange domain corresponds to cancer in situ, and the red domain corresponds to invasive cancer. The connective tissue has the most cells and overlaps with five other domains. The domain annotation has provided us with a valuable reference that we can use to evaluate the effectiveness of each clustering outcome.

The HER2 tumor data set is more challenging than the DLPFC data set (used before this section) in a clustering task for two reasons. First, the size of the HER2 tumor data set is small. As a result, the data quality may not be high enough. Second, the spatial structure of the HER2 tumor data set is more complex than that in the DLPFC data set. As seen from Figure 7D, each domain has more than one cluster; for example, invasive cancer has two clusters, and cancer in situ has three clusters. In contrast, the DLPFC data set has only one contiguous cluster for each type (see the ground truth in Fig. 2B). The clustering outcomes on this data set, presented thus far in the current literature, are not satisfactory. This is consistent with our repeated experiment using their methods (see our result below).

Our experimental result shown in Figure 7E reveals that KBC successfully identified the two cancer domains, namely, cancer in situ and invasive cancer. In this task, it is crucial to detect and distinguish these two cancer domains.

Yet, all other methods can only partially detect some subdomains, and they are often fragmented. For example, although SpatialPCA has identified the two cancers, there are many incorrect identifications; for example, a large noncancerous domain was identified as cancerous, and cancer in situ was mistaken as invasive cancer, and vice versa. In addition, the two cancers identified are fragmented, and the regions of the correct identifications are small.

This poor clustering outcome of SpatialPCA has two main reasons. First, it is the use of the Walktrap clustering algorithm, in which its weakness has been stipulated in the ablation study section. Second, the spatial information learning ability of SpatialPCA is not suitable for the spatial structure of this data set. We have applied KBC on the SpatialPCA transformed data but still could not get good results. This is one of the example data sets for which SpatialPCA is not the best data transformation method, unlike what we have shown earlier using the DLPFC data set.

The weak clustering outcomes of BayesSpace and SpaGCN may be owing to the clustering algorithms employed to perform the initial clustering. They use *k*-means, Louvain, or Mclust to find the initial clusters. As can be seen in the last row in Figure 7E, the clustering results of *k*-means, Louvain, and Mclust

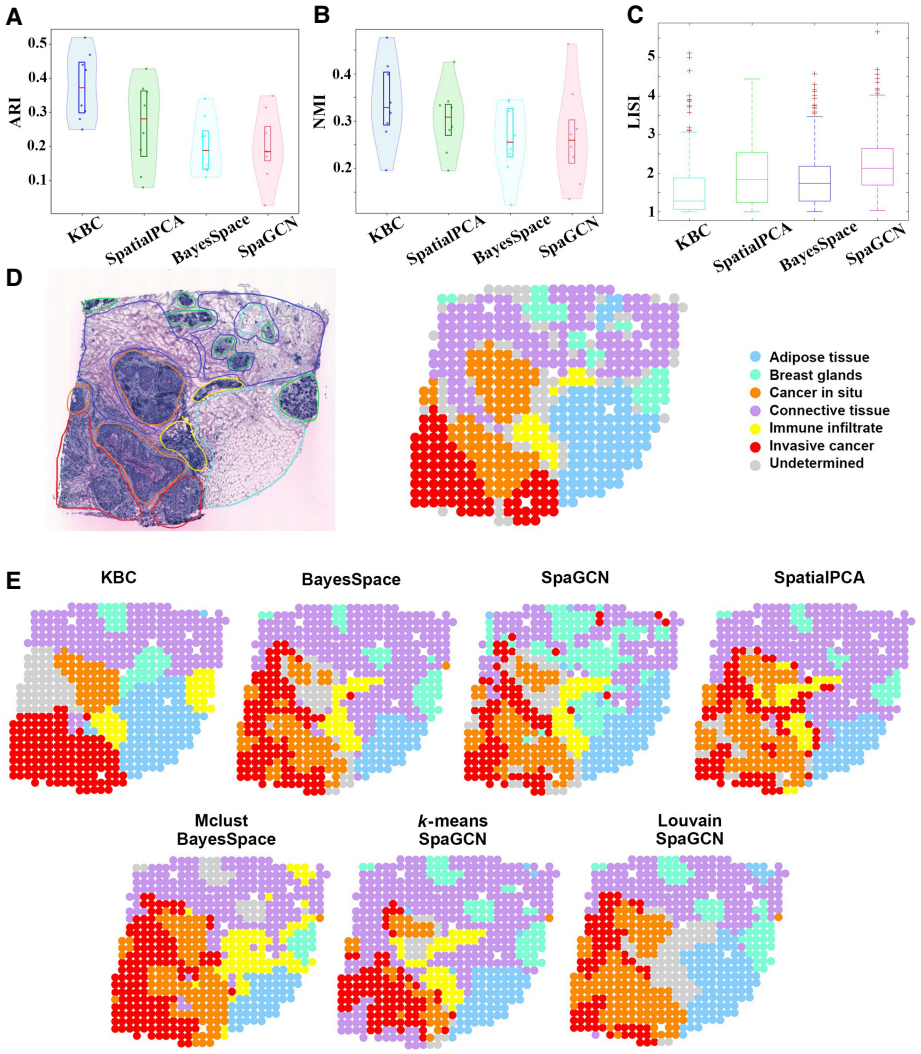

Figure 7. Application of KBC to the HER2 tumor data. (*A*,*B*) The violin plots of results obtained from different methods (for sections A1 to H1). (*C*) The boxplot in terms of local inverse Simpsons index (LISI) (Korsunsky et al. 2019) for different sections (from A1 to H1). A lower LISI value indicates a more uniform cluster of adjacent spatial domains. Thus, the smaller LISI the better. The red cross points are outliers of the LISI. (*D*) The histology image and manual annotation plot of section H1. (*E*) Clustering outcomes of four methods: KBC, BayesSpace (*k*-means), SpaGCN (Louvain), and SpatialPCA for section H1. The *bottom* row indicates three example cluster outcomes of BayesSpace and SpaGCN, but they employ the initial clusters produced by Mclust, *k*-means, and Louvain, respectively. Two results of SpaGCN use Louvain to produce the initial clusters but with different parameter settings.

are not good enough. Using them as the initial clusters is not a good choice. In addition, excluding the impact of initial clusters employed, the clustering performances of SpaGCN and BayesSpace are not good either. For example, SpaGCN does not produce a clustering result that is better than its initial clusters, produced by Louvain.

The clustering result of KBC was the best because almost every cluster discovered has a solid mass of neighboring points, and many clusters correspond closely to the annotated domains. This is also reflected in terms of ARI and NMI results, which are the highest among all contenders (Fig. 7A,B). Also, KBC's median LISI = 1.28 is also the best (the median LISIs of SpatialPCA, BayesSpace, and SpaGCN are 1.84, 1.74, and 2.13, respectively) (Fig. 7C).

Clustering plays an important role for discovering new biological insights from spatial genomics data. Here we provide an example. There are two key discrepancies between the ground-truth labels and the clusters identified by KBC. First, KBC suggests that the cells in bottom-left region are from the same group, but the ground-truth labels indicate that it has two separate regions of invasive cancer and cancer in situ. Second, the middle-right region is identified to be immune infiltrate by KBC, but the ground truth label is breast glands. Our examination via differential expression analyses reveal that some ground-truth labels may be re-examined (for details, see Supplemental Fig. S1 in Supplemental Text S2, Further Analysis of the KBC Clustering Outcomes of the HER2 Tumor Data). It is important to note that ground-truth labels are often assigned by humans, who may introduce potential errors in the labeling process.

Application of KBC to the mouse hippocampus data

We downloaded the Slide-seq V2 (Stickels et al. 2021) data set from the Single Cell Portal SCP815 website and focused on the mouse

hippocampus data (in the file named "Puck_200115_08"). The sample contains 53,208 cells, each having 23,264 genes. To have a fair comparison of all methods, any data transformation method employed deals with the same gene expression and spatial information. This data set does not have tissue domain annotation that could be used as ground truth. So, we relied on the Allen Brain Atlas (Fig. 8A) to help us understand the original tissue structure. Note that although the Allen Brain Atlas provides the regional divisions, there are no good mappings between the altas and the data set, so it cannot be relied upon completely.

The spatial domains identified by KBC, SpatialPCA, and Stagate were consistent with the coronal mouse olfactory bulb from the Allen Brain Atlas in the central area, labeled as CA1sp and CA3sp in Figure 8A. Looking at the CA1sp, CA3sp, and DG-sg domains only in Figure 8C, all methods could identify these domains. The exception is SpaGCN, which groups all three domains into one single cluster. KBC's clustering outcome, shown in Figure 8C, matches the most to the Allen Brain Atlas, in which each of the three domains is accurately identified. In terms of identifying the DG-sg domain, KBC and SpatialPCA produce similarly good results, but Stagate's result is not satisfactory because the cluster merges with other domains. In addition, KBC clearly distinguishes the CA1sp domain from the other domains. But, SpatialPCA produces a cluster that encompasses more than the CA1sp domain. For the CA3sp domain, the results of the KBC, Stagate, and SpatialPCA are similar.

In terms of LISI, KBC has the smoothest clustering result, achieving the lowest mean LISI = 1.03 among all the four methods shown in Figure 8B. Note that no ARI/NMI results can be produced because there is no ground truth in this data set.

This data set is the largest of the data sets used in this paper, almost one order of magnitude larger than the DLPFC data. Therefore, the clustering of this data set took significantly more computing resources than other data sets. Yet, the proposed KBC and graph-embedding WL could complete running this data set in a short time. The WL graph-embedding method took 2541 sec, which is the fastest. In comparison, the other methods took >10,000 sec. For the clustering component, KBC took 5 CPU seconds, whereas the fastest of the other methods, SpaGCN, took 231 GPU seconds.

Application of KBC to the DLPFC data

DLPFC (Pardo et al. 2022) is a 10x Genomics Visium data set generated from healthy human brain samples from the DLPFC domain. We downloaded the data set from the spatialLIBD website. There are 12 samples from three individuals in the full data set. Each sample has been manually annotated with the white matter (WM) and layers of the cortex based on the morphological features and gene markers (Maynard et al. 2021). These annotations are treated as ground-truth labels.

Here we take DLPFC tissue slice 151669 as an example to demonstrate the advantages of KBC. The tissue slice has 3661 spots and 33,538 genes. The spots are divided into five groups with four layers and the WM. The manual annotation plot in Figure 9D shows the distribution of these five groups, which have distinctive layers from top to bottom. Layer1 is the largest, which has 2141 cells, whereas others have only 400 cells on average. It can be clearly seen from Figure 9D that KBC has achieved the best result in terms of recognizing the largest (red) domain of Layer1. Other methods tend to divide Layer1 into several clusters. For all 12 slices, KBC achieves the highest accuracy with a mean ARI 0.58 (Fig. 9A) and a mean NMI 0.68 (Fig. 9B). In terms of LISI, KBC produces the lowest mean LISI = 1.06 (Fig. 9C).

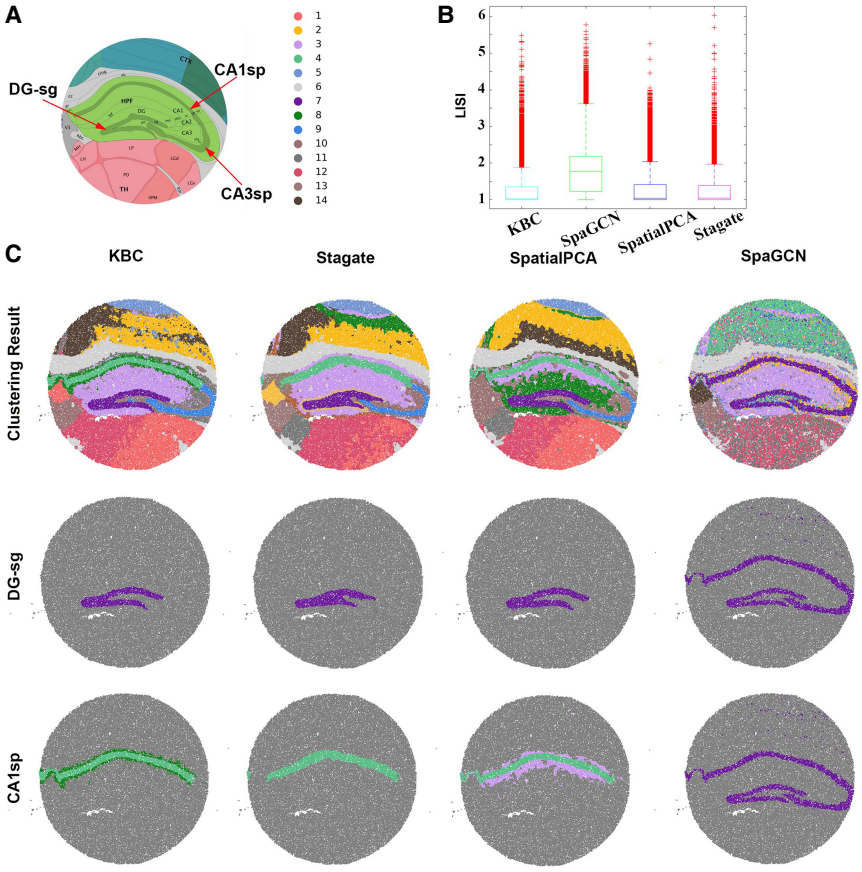

Figure 8. Application of KBC to the mouse hippocampus data. (*A*) Allen Brain Atlas P56 coronal. The diagram shows the structure of the mouse hippocampus. (*B*) The LISI index of the clustering results of KBC, SpaGCN, SpatialPCA, and Stagate. (*C*) A comparison of four clustering results on the CA1sp domain and the DG-sg domain.

Studies on simulated ST data

SRTsim, which is developed by Zhu et al. (2023), is a spatial pattern preserving simulator for scalable, reproducible, and realistic SRT simulations. Here we apply SRTsim to generate simulated data sets. The LIBD human DLPFC (Maynard et al. 2021) data set (12 sections from 151507 to 151676) are used as references

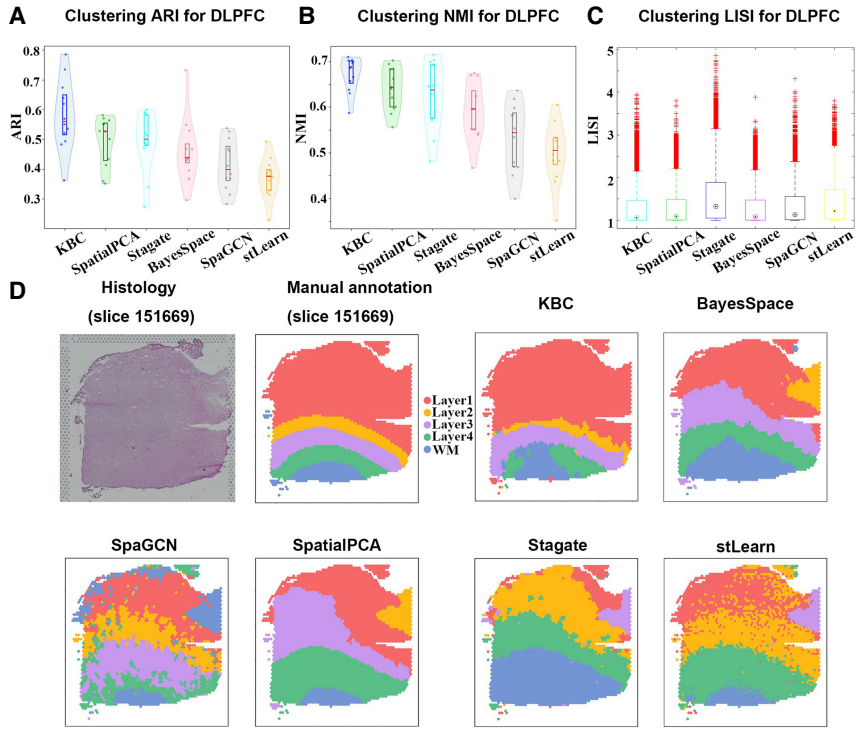

Figure 9. Application of KBC to the DLPFC data. (*A*,*B*) The violin plots of the results obtained from six different methods on the DLPFC data set. (*C*) The boxplot of clustering LISI of the six different methods on the DLPFC data set. (*D*) Histology image, manual annotation (Maynard et al. 2021), and the clustering results of KBC, BayesSpace, SpaGCN, SpatialPCA, Stagate, and stLearn plotted on DLPFC slice 151669.

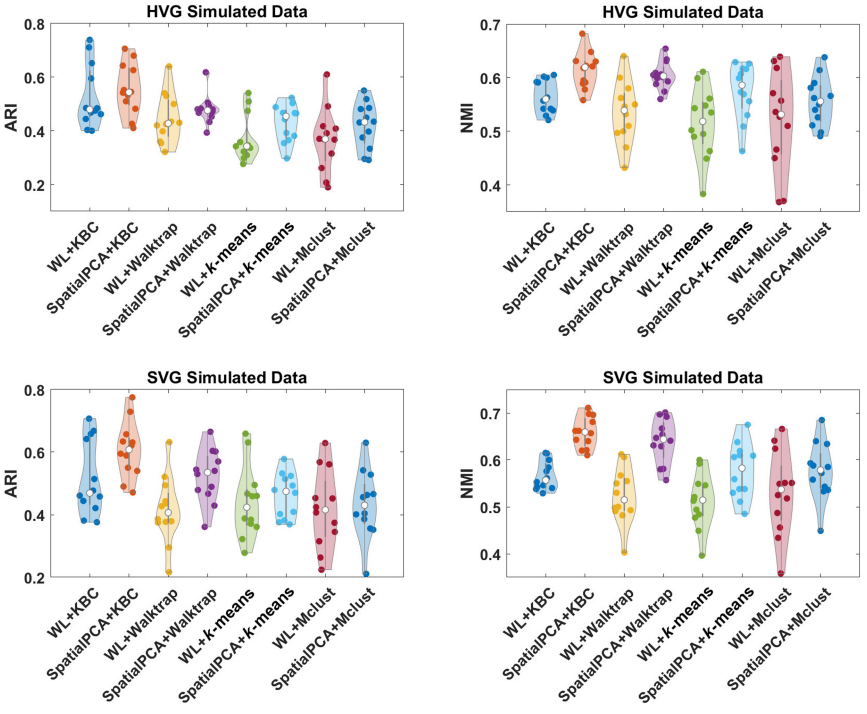

Figure 10. Results of the further ablation studies on four clustering methods and two data transformation methods using the HVG and SVG simulated data sets.

for SRTsim to conduct reference-based tissue-wise model fitting and data simulation procedures. Specifically, we randomly generated a simulated data set for each section of the real data. We further selected up to 3000 significant SVGs with SPARK (Sun et al. 2020) and the top 2000 highly variable genes (HVGs) with Seurat (Satija et al. 2015) for each simulated section to generate the SVG and HVG data sets for each section. To demonstrate the robustness of the proposed method, we conducted the experiments 10 times with different random seeds and evaluated the performance of each method on each of the simulated SVG and HVG data sets.

Based on the simulated data sets, we conducted ablation studies to investigate the clustering performance of different clustering algorithms, including KBC, Walktrap, *k*-means, and Mclust, as well as different embedding methods, such as WL and SpatialPCA. Additionally, we performed sensitivity analyses focusing on the number of principal components used in the data preprocessing steps (Supplemental Fig. S2), the parameters ψ and τ chosen for KBC (Supplemental Fig. S3), and the parameter *h* in WL (Supplemental Fig. S4). The detailed results are presented in the following subsections. Parameter search ranges used in the experiments are shown in Supplemental Text S1 (Supplemental Table S1).

Ablation studies on simulated data sets

This section reports the results of ablation studies on simulated data sets to evaluate the performance of different clustering algorithms (KBC, Walktrap, *k*-means, and Mclust) and embedding methods (WL and SpatialPCA). They are the best clustering algorithms and data transformation methods found in the two ablation studies we have reported in the section on ablation studies. The experiments are conducted on both simulated HVG and SVG data sets.

When integrated with the same embedding method (either WL or SpatialPCA), KBC clearly outperforms other algorithms in either setting, achieving the highest median NMI and ARI scores (represented by hollow circles in Fig. 10). Furthermore, SpatialPCA + KBC outperforms WL + KBC in both HVG and SVG data sets. Note that these observations are consistent with the results presented in the section on ablation studies.

A caveat is in order with regard to SpatialPCA + KBC, which has the best performance on the DLPFC data set (as reported in the section on ablation studies) and on the simulated data sets based on the same DLPFC data set. One shall take care in generalizing this outcome to other data sets. As we have pointed out in the two sections on the HER2 tumor and mouse hippocampus data sets, the WL method works much better than SpatialPCA on these data sets, which contain complex patterns.

Discussion

WL and SpatialPCA are efficient methods for data transformation and dimension reduction. When integrated with WL or SpatialPCA, KBC can accurately identify spatial domains in a single tissue section. However, WL and SpatialPCA could not directly handle data from multiple batches or tissue sections. When analyzing multiple batches or tissue sections, our pipeline could either (1) analyze each tissue independently with WL or SpatialPCA or (2) use an alternative data transformation method that could correct batch effects and simultaneously integrate multiple sections. Recently, a dependency-aware deep generative model spaVAE has been proposed (Tian et al. 2024) that could jointly model multiple batches and sections by conditional variational auto-encoders and perform efficient dimension reduction for a SRT data. Equipped with the transformed data generated by spaVAE, KBC (or any stand-alone clustering method) is expected to detect cell types across related tissue sections concurrently.

We have only used data sets with non-single-cell resolution in our applications thus far. But, neither the WL/SpatialPCA method nor KBC is limited to this resolution only. These methods are natively applicable to data sets generated by ST technologies with cellular or subcellular resolutions. Supplemental Text S4 (Supplemental Fig. S8) shows an example result of KBC integrated with WL for analyzing the mouse olfactory bulb Stereo-seq data (Chen et al. 2022). Stereo-seq is a newly emerging ST technology that could achieve the cellular or subcellular spatial resolution by DNA nanoball patterned array chips (Chen et al. 2022). As shown in Supplemental Text S4, the results from KBC accurately reflected the laminar organization and corresponded well with the annotated layers.

Note that the above-mentioned two capabilities of KBC are the same for all clustering methods, independent of the data transformation employed and the SRT technology used.

Methods

Data preprocessing

For all data sets, we normalized the raw molecular count matrix using the variance stabilizing transformation method called SCTransform (Choudhary and Satija 2022), which is a negative binomial regression model implemented in the Seurat package (Satija et al. 2015). Following the method of Shang and Zhou (2022), we filter genes to reserve a set of SVGs only. Specifically, we apply SPARK (Sun et al. 2020) for SVG analysis in small data sets owing to its higher statistical power and use SPARK-X (Zhu et al. 2021) for SVG analysis in large data sets (e.g., the mouse hippocampus data set) in order to save time and memory. We select up to 3000 significant SVGs for each single data set with a false-discovery rate (FDR) of 0.05 as input. The normalization and gene selection steps are conducted with the default parameters recommended by those packages. The details of the original data source of each data set used and the sources of competing methods can be found in the Supplemental Text S1 (Supplemental Table S2).

The processed expression matrix is centered (to have zero mean) but not scaled for each feature before applying the principal component analysis (PCA). Here, we applied PCA to extract the first 15 principal components to produce the feature matrix to represent the gene expression. We show in Supplemental Figure S2 (in Supplemental Text S2) that the proposed pipeline is robust against the number of principal components selected. At the end of the preprocessing, a gene expression feature matrix of m (genes) × n (tissue locations) was obtained.

A k-nearest neighbor search was performed on the spatial coordinates. Depending on the SRT technology employed, there are four ($k=4$) and six ($k=6$) neighboring spots for ST platform (Andersson et al. 2021) and Visium (Pardo et al. 2022), respectively. An adjacency matrix was generated based on the locations of neighboring cells. Note that this is the data characteristic, and this property is used for all methods of data transformation, including the WL method we introduced.

The combined information from the gene expression feature matrix and the adjacency matrix produces a graph. Then, a graph-embedding method (Supplemental Table S3), such as the WL scheme, converts the graph to an embedded matrix of $h \times n$, where h is the parameter of the WL scheme. Note that many data transformation methods, such as Stagate and SpaGCN, also involve a graph and thus require graph embedding, albeit the actual embedding methods employed differ. A clustering algorithm such as the proposed KBC takes the embedded matrix as input to find the clusters in it.

Kernel bounded clustering

Let $\kappa(\mathbf{x}, \mathbf{y})$ be a kernel that measures the similarity between two points $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$, and let $\mathsf{K}(\mathbf{P}_X, \mathbf{P}_Y)$ be a distributional kernel that measures the similarity between two distributions $\mathbf{P}_X$ and $\mathbf{P}_Y$, where $\mathbf{P}_W$ is an unknown distribution that generates iid points in set $W \subset \mathbb{R}^d$. Further let $\delta(\mathbf{x})$ be a Dirac measure that converts a point $\mathbf{x}$ into a distribution.

Given a data set $D \subset \mathbb{R}^d$, the proposed clustering algorithm, named kernel-bounded clustering (KBC), discovers the clusters in D. KBC has two key steps, as shown in Algorithm 1. The first step identifies k initial clusters G_j having points with similarities higher than a threshold τ. As points in each of the initial clusters have high similarity to each other, they are good representatives of a distribution that generates these points. As a result, the initial clusters can be obtained using a subset of the given data set, namely, $D_s \subset D$, in order to reduce its time complexity.

Algorithm 1 Kernel bounded clustering

input: D- given data set, k- number of clusters, s- sample size, τ- similarity threshold
output: $\mathbb{C} = \{C_1, \ldots, C_k\}$

1. From $D_s \subset D$, find the largest k initial clusters G_j via kernel κ as follows: $G_j = \{\mathbf{x}, \mathbf{y} \in D \mid$ there exists a chain: $\mathbf{z}_1, \mathbf{z}_2, \cdots, \mathbf{z}_l; \mathbf{z}_1 = \mathbf{x}, \mathbf{z}_l = \mathbf{y}, \forall i\ \kappa(\mathbf{z}_i, \mathbf{z}_{i+1}) > \tau\}$, $j = 1, \ldots, k$.
2. From D, recruit members of C_i via distributional kernel K and initial clusters G_j, $j = 1, \ldots, k$:
   $C_i = \{\mathbf{x} \in D | \text{argmax}_{j \in [1,k]} \mathsf{K}(\delta(\mathbf{x}), \mathbf{P}_{G_j}) = i\}$, $i = 1, \ldots, k$.

The second step recruits members of each cluster from D via a distributional kernel K by assigning each point $\mathbf{x} \in D$ to a distribution estimated by initial cluster G in which $\mathbf{x}$ has the highest similarity, as measured by K.

Using kernel mean embedding (Smola et al. 2007; Muandet et al. 2017), the empirical estimation of the distributional kernel K on two distributions $\mathbf{P}_X$ and $\mathbf{P}_Y$, which is based on a point kernel

κ on points $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$, is given as

$$\mathrm{K}(\mathbf{P}_X, \mathbf{P}_Y) = \frac{1}{|X||Y|}\sum_{\mathbf{x}\in X}\sum_{\mathbf{y}\in Y}\kappa(\mathbf{x}, \mathbf{y}) = \langle \phi(\mathbf{P}_X), \phi(\mathbf{P}_Y)\rangle, \quad (1)$$

where $\phi(\mathbf{P}_W) = \frac{1}{|W|}\sum_{\mathbf{x}\in W}\varphi(\mathbf{x})$ is the empirical estimation of the feature map of $\mathrm{K}(\mathbf{P}_W, \cdot)$, and $\varphi(\mathbf{x})$ is the feature map of the point kernel $\kappa(\mathbf{x}, \cdot)$.

The distributional approach for clustering has two main advantages over a nondistributional approach. First, the discovered clusters can have arbitrary shapes, varied sizes and densities. This is because each cluster is represented as a distribution, and using a nonparametric method such as kernel mean embedding, the distribution does not need to be modeled and it can be left unknown. Second, the kernel computation is very efficient using a finite-dimensional feature map of the kernel (for details, see Ting et al. 2020; Xu et al. 2021). KBC has linear time complexity $O(s^2 + kn)$, as all parameters, except n, are constant.

KBC employs isolation kernel

The proposed KBC shall use isolation kernel, a data-dependent kernel introduced recently (Ting et al. 2018; Qin et al. 2019). It has been shown to improve the accuracy of SVM classifier and density-based clustering algorithm DBSCAN. The pertinent details of isolation kernel are provided in this section.

Let $D \subset \mathbb{R}^d$ be a data set sampled from an unknown $\mathbf{P}_D$, and let $\mathbb{H}_\psi(D)$ denote the set of all partitionings H that are admissible from $\mathcal{D} \subset D$, where each point $\mathbf{z} \in \mathcal{D}$ has the equal probability of being selected from D, and $|\mathcal{D}| = \psi$. Each isolating partition $\theta[\mathbf{z}] \in H$ isolates a point $\mathbf{z} \in \mathcal{D}$ from the rest of the points in $\mathcal{D}$. Let $\mathbb{1}(\cdot)$ be an indicator function.

Definition 1. For any two points $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$, the similarity of $\mathbf{x}$ and $\mathbf{y}$, as measured by isolation kernel (Ting et al. 2018; Qin et al. 2019), is defined to be the expectation taken over the probability distribution on all partitionings $H \in \mathbb{H}_\psi(D)$ that both $\mathbf{x}$ and $\mathbf{y}$ fall into the same isolating partition $\theta[\mathbf{z}] \in H$:

$$\begin{aligned}\kappa_\psi(\mathbf{x}, \mathbf{y}|D) &= \mathbb{E}_{\mathbb{H}_\psi(D)}[\mathbb{1}(\mathbf{x}, \mathbf{y} \in \theta[\mathbf{z}]|\theta[\mathbf{z}] \in H)] \\ &= P(\mathbf{x}, \mathbf{y} \in \theta[\mathbf{z}]|\mathbf{z} \in \mathcal{D} \subset D).\end{aligned} \quad (2)$$

In practice, $\kappa_\psi(\cdot|D)$ is constructed using a finite number of partitionings H_i, $i = 1, \ldots, t$, where each H_i is created using a randomly sampled subset $\mathcal{D}_i \subset D$, and θ is a shorthand for $\theta[\mathbf{z}]$:

$$\kappa_\psi(\mathbf{x}, \mathbf{y}|D) = \frac{1}{t}\sum_{i=1}^{t}\mathbb{1}(\mathbf{x}, \mathbf{y} \in \theta|\theta \in H_i) = \frac{1}{t}\sum_{i=1}^{t}\sum_{\theta\in H_i}\mathbb{1}(\mathbf{x} \in \theta)\mathbb{1}(\mathbf{y} \in \theta). \quad (3)$$

Isolation kernel defines a reproducing kernel Hilbert space because Equation 3 is a quadratic form.

The isolation partitioning mechanisms that have been used previously to implement isolation kernel are iForest (Liu et al. 2008; Ting et al. 2018), Voronoi diagram (Qin et al. 2019), and iNNE (Bandaragoda et al. 2018). We use iNNE in this work.

Each point $\mathbf{z} \in \mathcal{D}$ is isolated from the rest of the points in $\mathcal{D}$ by building a hypersphere that covers $\mathbf{z}$ only. The radius of the hypersphere is determined by the distance between $\mathbf{z}$ and its nearest neighbor in $\mathcal{D}$. In other words, a partitioning H consists of ψ hyperspheres $\theta[\mathbf{z}]$ and the $(\psi+1)$-th partition. The latter is the domain in $\mathbb{R}^d$, which is not covered by all ψ hyperspheres, where $2 \le \psi < |D|$.

Definition 2. The feature map of isolation kernel. For point $\mathbf{x} \in \mathbb{R}^d$, the feature mapping $\varphi: \mathbf{x} \to \{0, 1\}^{t\psi}$ of κ_ψ is a vector that represents the partitions in all the partitioning $H_i \in \mathbb{H}_\psi(D)$, $i = 1, \ldots, t$, where $\mathbf{x}$ falls into either only one of the ψ hyperspheres or none in each partitioning H_i.

Let $\mathbf{1}$ denote the value of $\varphi_i(\mathbf{x})$ such that $\varphi_{ij}(\mathbf{x}) = 1$ and $\varphi_{ik}(\mathbf{x}) = 0$, $\forall k \neq j$ for any $j, k \in [1, \psi]$.

φ has the following geometrical interpretation:

1. $\varphi(\mathbf{x}) = [\mathbf{1}, \ldots, \mathbf{1}]$: $\| \varphi(\mathbf{x}) \| = \sqrt{t}$ and $\kappa_\psi(\mathbf{x}, \mathbf{x}|D) = 1$ iff $\varphi_i(\mathbf{x}) \neq \mathbf{0}$ for all $i \in [1, t]$.
2. For point $\mathbf{x}$ such that $\exists i \in [1, t]$, $\varphi_i(\mathbf{x}) = \mathbf{0}$, then $\| \varphi(\mathbf{x}) \| < \sqrt{t}$.
3. If point $\mathbf{x} \in \mathbb{R}^d$ falls outside of all hyperspheres in H_i for all $i \in [1, t]$, then it is mapped to the origin of the feature space $\varphi(\mathbf{x}) = [\mathbf{0}, \ldots, \mathbf{0}]$.

Note that the isolation kernel is a data-dependent kernel, which derives its feature map from a data set directly, and it has no closed form expression. In contrast, the Gaussian kernel, like most other commonly used kernel, is a data-independent kernel, which has a closed form expression. A comparison between the isolation kernel and Gaussian kernel and adaptive Gaussian kernel (Zelnik-Manor and Perona 2005) used in KBC can be found in Supplemental Text S3 (Supplemental Figs. S5–S7).

Software availability

The preprocessed spatial genomics data sets we used and the KBC software code are available at GitHub (https://github.com/IsolationKernel/Kernel-Bounded-Clustering-for-Spatial-Transcriptomics) and as Supplemental Code. The source code is released under a noncommercial use license.

Competing interest statement

The authors declare no competing interests.

Acknowledgments

K.M.T. is supported by the National Natural Science Foundation of China (NNSFC, grants no. 92470116 and no. 62076120). J.Z. is supported by the Natural Science Foundation of Jiangsu Province (grant no. BK20230781) and the NNSFC (grant no. 62306134).

Author contributions: KBC model development and implementation were by H.Z. Conceptualization and experimental design were by Y.Z., K.M.T., and J.Z. Experiments and data analysis were by Y.Z. and Q.Z. Supervising was by K.M.T. and J.Z. Writing was by Y.Z., K.M.T., and J.Z. All authors critically reviewed the manuscript.

References

Aggarwal CC. 2015. *Data mining: the textbook*, Vol. 1. Springer, Cham, Switzerland.

Andersson A, Larsson L, Stenbeck L, Salmén F, Ehinger A, Wu SZ, Al-Eryani G, Roden D, Swarbrick A, Borg Å, et al. 2021. Spatial deconvolution of HER2-positive breast cancer delineates tumor-associated cell type interactions. *Nat Commun* **12:** 6012. doi:10.1038/s41467-021-26271-2

Arthur D, Vassilvitskii S. 2006. How slow is the *k*-means method? In *SCG '06: Proceedings of the twenty-second annual symposium on Computational Geometry*, Sedona, AZ, pp. 144–153. doi:10.1145/1137856.1137880

Asp M, Bergenstråhle J, Lundeberg J. 2020. Spatially resolved transcriptomes: next generation tools for tissue exploration. *Bioessays* **42:** 1900221. doi:10.1002/bies.201900221

Bandaragoda TR, Ting KM, Albrecht D, Liu FT, Zhu Y, Wells JR. 2018. Isolation-based anomaly detection using nearest-neighbor ensembles. *Comput Intell* **34:** 968–998. doi:10.1111/coin.12156

Blondel VD, Guillaume JL, Lambiotte R, Lefebvre E. 2008. Fast unfolding of communities in large networks. *J Stat Mech-Theory Exp* **2008:** P10008. doi:10.1088/1742-5468/2008/10/P10008

Chen A, Liao S, Cheng M, Ma K, Wu L, Lai Y, Qiu X, Yang J, Xu J, Hao S, et al. 2022. Spatiotemporal transcriptomic atlas of mouse organogenesis using DNA nanoball-patterned arrays. *Cell* **185:** 1777–1792.e21. doi:10.1016/j.cell.2022.04.003

Choudhary S, Satija R. 2022. Comparison and evaluation of statistical error models for scRNA-seq. *Genome Biol* **23:** 27. doi:10.1186/s13059-021-02584-9

Cover TM. 1999. *Elements of information theory*. Wiley, Hoboken, NJ.

Dempster AP, Laird NM, Rubin DB. 1977. Maximum likelihood from incomplete data via the *EM* algorithm. *J R Stat Soc Ser B-Stat Methodol* **39:** 1–22. doi:10.1111/j.2517-6161.1977.tb01600.x

Dong K, Zhang S. 2022. Deciphering spatial domains from spatially resolved transcriptomics with an adaptive graph attention auto-encoder. *Nat Commun* **13:** 1739. doi:10.1038/s41467-022-29439-6

Ester M, Kriegel HP, Sander J, Xu X. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Portland, OR, Vol. 96, no. 34, pp. 226–231.

Han X, Zhu Y, Ting KM, Zhan DC, Li G. 2022. Streaming hierarchical clustering based on point-set kernel. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, Washington, DC.

Hartigan JA, Wong MA. 1979. Algorithm as 136: a *k*-means clustering algorithm. *J R Stat Soc Ser C-Appl Stat* **28:** 100–108. doi:10.2307/2346830

Hu J, Li X, Coleman K, Schroeder A, Ma N, Irwin DJ, Lee EB, Shinohara RT, Li M. 2021. SpaGCN: integrating gene expression, spatial location and histology to identify spatial domains and spatially variable genes by graph convolutional network. *Nat Methods* **18:** 1342–1351. doi:10.1038/s41592-021-01255-8

Korsunsky I, Millard N, Fan J, Slowikowski K, Zhang F, Wei K, Baglaenko Y, Brenner M, Loh P, Raychaudhuri S. 2019. Fast, sensitive and accurate integration of single-cell data with harmony. *Nat Methods* **16:** 1289–1296. doi:10.1038/s41592-019-0619-0

Liu FT, Ting KM, Zhou ZH. 2008. Isolation forest. In *IEEE International Conference on Data Mining*, Pisa, Italy.

Lyu B, Wu W, Hu Z. 2021. A novel bidirectional clustering algorithm based on local density. *Sci Rep* **11:** 14214. doi:10.1038/s41598-021-93244-2

Maćkiewicz A, Ratajczak W. 1993. Principal components analysis (PCA). *Comput Geosci* **19:** 303–342. doi:10.1016/0098-3004(93)90090-R

MacQueen J. 1967. Some methods for classification and analysis of multivariate observations. In *Berkeley Symposium on Mathematical Statistics and Probability*, University of California Press, Berkeley, CA.

Marx V. 2021. Method of the year: spatially resolved transcriptomics. *Nat Methods* **18:** 9–14. doi:10.1038/s41592-020-01033-y

Maynard KR, Collado-Torres L, Weber LM, Uytingco C, Barry BK, Williams SR, Catallini JL, Tran MN, Besich Z, Tippani M, et al. 2021. Transcriptome-scale spatial gene expression in the human dorsolateral prefrontal cortex. *Nat Neurosci* **24:** 425–436. doi:10.1038/s41593-020-00787-0

Morris C, Ritzert M, Fey M, Hamilton WL, Lenssen JE, Rattan G, Grohe M. 2019. Weisfeiler and Leman go neural: higher-order graph neural networks. In *AAAI Conference on Artificial Intelligence*, Vancouver, Canada.

Muandet K, Fukumizu K, Sriperumbudur B, Schölkopf B. 2017. Kernel mean embedding of distributions: a review and beyond. *Found Trends Mach Learn* **10:** 1–141. doi:10.1561/2200000060

Nadler B, Galun M. 2006. Fundamental limitations of spectral clustering. In *International Conference on Neural Information Processing Systems*, Vancouver, British Columbia, Canada, pp. 1017–1024.

Pardo B, Spangler A, Weber LM, Page SC, Hicks SC, Jaffe AE, Martinowich K, Maynard KR, Collado-Torres L. 2022. spatialLIBD: an R/Bioconductor package to visualize spatially-resolved transcriptomics data. *BMC Genomics* **23:** 434. doi:10.1186/s12864-022-08601-w

Pham D, Tan X, Balderson B, Xu J, Grice LF, Yoon S, Willis EF, Tran M, Lam PY, Raghubar A, et al. 2023. Robust mapping of spatiotemporal trajectories and cell–cell interactions in healthy and diseased tissues. *Nat Commun* **14:** 7739. doi:10.1038/s41467-023-43120-6

Pons P, Latapy M. 2005. Computing communities in large networks using random walks. In *International Symposium on Computer and Information Sciences*, Istanbul, Turkey.

Qin X, Ting KM, Zhu Y, Lee VCS. 2019. Nearest-neighbour-induced isolation similarity and its impact on density-based clustering. In *AAAI Conference on Artificial Intelligence*, Honolulu, HI.

Rodriguez A, Laio A. 2014. Clustering by fast search and find of density peaks. *Science* **344:** 1492–1496. doi:10.1126/science.1242072

Satija R, Farrell JA, Gennert D, Schier AF, Regev A. 2015. Spatial reconstruction of single-cell gene expression data. *Nat Biotechnol* **33:** 495–502. doi:10.1038/nbt.3192

Scott DW. 2015. *Multivariate density estimation: theory, practice, and visualization*, 2nd ed. Wiley, Hoboken, NJ.

Scrucca L, Fop M, Murphy TB, Raftery AE. 2016. MClust 5: clustering, classification and density estimation using Gaussian finite mixture models. *R J* **8:** 289. doi:10.32614/RJ-2016-021

Shang L, Zhou X. 2022. Spatially aware dimension reduction for spatial transcriptomics. *Nat Commun* **13:** 7203. doi:10.1038/s41467-022-34879-1

Shervashidze N, Schweitzer P, Van Leeuwen EJ, Mehlhorn K, Borgwardt KM. 2011. Weisfeiler-Lehman graph kernels. *J Mach Learn Res* **12:** 2539–2561.

Smola A, Gretton A, Song L, Schölkopf B. 2007. A Hilbert space embedding for distributions. In *International Conference on Algorithmic Learning Theory* (ed. Hutter M, et al.), pp. 13–31. Springer, Berlin, Heidelberg.

Stickels RR, Murray E, Kumar P, Li J, Marshall JL, Di Bella DJ, Arlotta P, Macosko EZ, Chen F. 2021. Highly sensitive spatial transcriptomics at near-cellular resolution with slide-seqV2. *Nat Biotechnol* **39:** 313–319. doi:10.1038/s41587-020-0739-1

Sun S, Zhu J, Zhou X. 2020. Statistical analysis of spatial expression patterns for spatially resolved transcriptomic studies. *Nat Methods* **17:** 193–200. doi:10.1038/s41592-019-0701-7

Tian T, Zhang J, Lin X, Wei Z, Hakonarson H. 2024. Dependency-aware deep generative models for multitasking analysis of spatial omics data. *Nat Methods* **21:** 1501–1513. doi:10.1038/s41592-024-02257-y

Ting KM, Zhu Y, Zhou ZH. 2018. Isolation kernel and its effect on SVM. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, London, pp. 2329–2337.

Ting KM, Xu BC, Washio T, Zhou ZH. 2020. Isolation distributional kernel: a new tool for kernel based anomaly detection. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Virtual Event, CA, pp. 198–206.

Togninalli M, Ghisu E, Llinares-López F, Rieck B, Borgwardt K. 2019. Wasserstein Weisfeiler–Lehman graph kernels. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems (NeurIPS 2019)*, Vancouver, Canada. Curran Associates Inc., Red Hook, NY.

Torgerson WS. 1952. Multidimensional scaling: I. Theory and method. *Psychometrika* **17:** 401–419. doi:10.1007/BF02288916

Traag VA, Waltman L, van Eck NJ. 1996. Probability density estimation using a Gaussian clustering algorithm. *Neural Comput Appl* **4:** 149–160. doi:10.1007/BF01414875

Traag VA, Waltman L, van Eck NJ. 2019. From Louvain to Leiden: guaranteeing well-connected communities. *Sci Rep* **9:** 5233. doi:10.1038/s41598-019-41695-z

Ward JHJ. 1963. Hierarchical grouping to optimize an objective function. *J Am Stat Assoc* **58:** 236–244. doi:10.1080/01621459.1963.10500845

Weisfeiler B, Lehman A. 1968. A reduction of a graph to a canonical form and an algebra arising during this reduction. *Nauchno-Technicheskaya Informatsia* **2:** 12–16.

Williams CG, Lee HJ, Asatsuma T, Vento-Tormo R, Haque A. 2022. An introduction to spatial transcriptomics for biomedical research. *Genome Med* **14:** 68. doi:10.1186/s13073-022-01075-1

Xu BC, Ting KM, Jiang Y. 2021. Isolation graph kernel. In *Proceedings of the AAAI Conference on Artificial Intelligence*, New York.

Yeung KY, Ruzzo WL. 2001. Principal component analysis for clustering gene expression data. *Bioinformatics* **17:** 763–774. doi:10.1093/bioinformatics/17.9.763

Zelnik-Manor L, Perona P. 2005. Self-tuning spectral clustering. In *Proceedings of the 18th International Conference on Neural Information Processing Systems*, Vancouver, British Columbia, Canada, pp. 1601–1608.

Zhang S, Tong H, Xu J, Maciejewski R. 2019a. Graph convolutional networks: a comprehensive review. *Comput Soc Netw* **6:** 11. doi:10.1186/s40649-019-0069-y

Zhang X, Liu H, Li Q, Wu XM. 2019b. Attributed graph clustering via adaptive graph convolution. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, Macao, China, pp. 4327–4333.

Zhao E, Stone MR, Ren X, Guenthoer J, Smythe KS, Pulliam T, Williams SR, Uytingco CR, Taylor SE, Nghiem P, et al. 2021. Spatial transcriptomics at subspot resolution with BayesSpace. *Nat Biotechnol* **39:** 1375–1384. doi:10.1038/s41587-021-00935-2

Zhu Y, Ting KM, Carman MJ. 2016. Density-ratio based clustering for discovering clusters with varying densities. *Pattern Recognit* **60:** 983–997. doi:10.1016/j.patcog.2016.07.007

Zhu J, Sun S, Zhou X. 2021. SPARK-X: non-parametric modeling enables scalable and robust detection of spatial expression patterns for large spatial transcriptomic studies. *Genome Biol* **22:** 184. doi:10.1186/s13059-021-02404-0

Zhu Y, Ting KM, Jin Y, Angelova M. 2022. Hierarchical clustering that takes advantage of both density-peak and density-connectivity. *Inf Syst* **103:** 101871. doi:10.1016/j.is.2021.101871

Zhu J, Shang L, Zhou X. 2023. SRTsim: spatial pattern preserving simulations for spatially resolved transcriptomics. *Genome Biol* **24:** 39. doi:10.1186/s13059-023-02879-z

Received January 17, 2024; accepted in revised form December 19, 2024.

Kernel-bounded clustering for spatial transcriptomics enables scalable discovery of complex spatial domains

Hang Zhang, Yi Zhang, Kai Ming Ting, et al.

Genome Res. 2025 35: 355-367 originally published online February 5, 2025
Access the most recent version at doi:10.1101/gr.278983.124