Variation in Spatial Domain Identification Methods

Isaac von Riedemann[301423851]

{imv}@sfu.ca
CMPT419 Spring 2025, Prof. Hamarneh

Abstract. Spatial Transcriptomics allow for novel research on the spatial distribution of genes X. Furthermore, since the analysis is not destructive many types of analyses can be run on the same tissue. However, to explore X often homogeneous regions need to be detected of gene and tissue information. Spatial Domain Identification (SDI) is type of algorithm designed to find those regions and are an upstream task for many types of analyses. Little work has explored the cause of variation in these methods. This paper attempts to fill that gap.

Keywords: Spatial Transcriptomics · Multi Annotator Segmentation · Segmentation Variability · Spatial Domain Identification · Multi-region Segmentation

1 Introduction

1.1 Spatial Transciptomics

Spatial transcriptomics is a relatively new modality that enables the study of where genes are expressed within their cellular and tissue contexts. Its non-destructive nature is especially important, as it allows for simpler analysis of cells anchored in tissue compared to single-cell RNA sequencing (scRNA-seq) [1]. In situ sequencing (ISS) methods provide high read depth, while imaging-based methods can achieve subcellular resolution [2]. Here, I focus on an ISS-based method developed by 10x Genomics called Visium, shown in Figure 1. Visium uses a specialized microscope slide containing a microarray of wells arranged in a hexagonal grid. RNA is dissociated from the tissue, captured in the wells, and then sequenced using next-generation sequencing methods.

1.2 Spatial Domain Identification

Therefore, each well contains a distribution of genes. Because the gene expression of a cell is influenced by surrounding cells and the tissue environment [4], delineating domains of similar gene expression and tissue structure becomes important for downstream analyses. Spatial domain identification (SDI) aims to detect these domains within a tissue sample. Existing methods can be classified into those that incorporate histopathology (H&E) images [5,6,7] and traditional

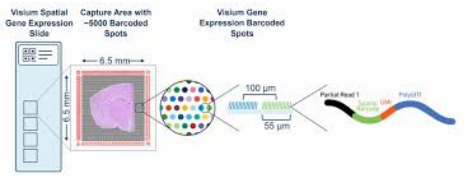


Fig. 1: Infographic of Visium Process [3]

methods that cluster gene expression alone, such as Louvain or k-means. In this work, I consider three methods that integrate H&E information: SpaGCN [7], CLGraph [5], and DeepST [6].

1.3 Downstream effects

SDI serves as an upstream step for many analyses. For example, technologies like Visium, which lack subcellular resolution, may capture multiple cells within a single well. Cell type deconvolution methods attempt to separate the combined gene expression into the specific gene distributions of individual cells. SpaDecon, a method that addresses this task [8], leverages SpaGCN for domain generation. My work investigates how using CLGraph could affect outcomes, or how alternative hyperparameters for SpaGCN may affect outcomes.

1.4 Report Map

The rest of the report is divided as follows:

1. **Materials** will discuss the data I use for analysis.
2. **Methods** will discuss the SDI algorithms I use as well as the analysis algorithms. It will discuss the hyperparameter I will explore and how it behaves in each method
3. **Results** will discuss some initial results from my analyses. I breakdown the analysis into exploring the gene distribution, exploring the segmentation masks and exploring how the gene distribution changes across paired domains between different methods and hyper-parameters
4. **Accomplishments** will summarize the work that I've done and challenges I've faced and overcome.
5. **Conclusion and Discussion** will summarize my findings and discuss limitations in the methods I've used. Future work, inspired by those limitations will be presented
6. **Future Work** The future work presented in the previous section will be explored. More formal definitions of the problems and existing work in the area will be presented.

2 Materials

The dataset I use for this analysis is the SpatialLIBD[9] dataset a collection of tissues slices from the dorsolateral prefrontal cortex in the brain. Visium H&E was applied to these tissues slices and the data is packaged with the gene expression and H&E image. Examples of the H&E images are shown in 2. The

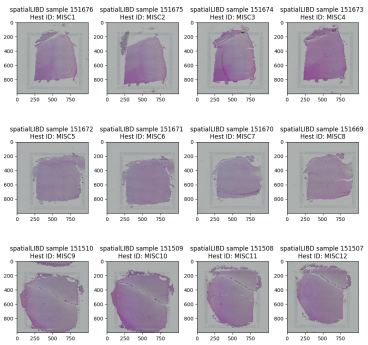


Fig. 2: Tissue Slices from SpatialLIBD

SpatialLIBD dataset has been nicely packaged by the Hest1k[10] project. From the Hest1k library the SpatialLIBD data is packaged in the following objects. The whole slide H&E image, metadata and the sequencing results as an Annotated Data (AnnData)[11] object. The layout of the AnnData object is centered around

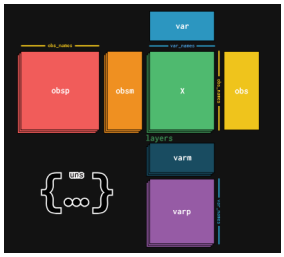


Fig. 3: Layout of AnnData object[11]

the gene distribution X shown in the center of figure 3. Along the y-axis of X is the spot which indexes the *obs* table. Along the x-axis is the gene. SDI methods use this format to read gene expression per-spot and save the spatial domain labels back in the *obs* table. Since image analysis methods are not defined on tabular data. I convert the spot labels to segmentation masks using the *img_col* and *img_row* column of the *obs* table. I then do my analysis of segmentation similarity on the segmentation masks.

3 Methods

The methods used in this study are divided between the SDI methods and the analysis methods. Since each SDI method incorporates the H&E information differently I'll give brief overview of each method. Each method has a hyper-parameter K: then number of clusters that behaves differently. I'll explain how each method uses that hyper-parameter.

3.1 Spatial Domain Identifcation Methods

SpaGCN SpaGCN constructs a weighted graph between spots, where each $w_i \in W$ is a linear combination of the H&E pixel values, the Euclidean distance between spots, and the gene expression at that spot. It applies a Graph CNN to generate spot embeddings, and domains are then identified through iterative clustering of those embeddings [7]. K is treated as the starting point for a search of optimal number of clusters using Louvain. If the optimal number is different than K it will use the optimal number instead

DeepST DeepST uses ImageNet [12] encoders to create embeddings of H&E image patches near each spot. These embeddings are treated as node embeddings of a weighted spot graph, where each $w_i \in W$ is the Euclidean distance between spots. Spot embeddings are then refined through a self-supervised Graph Autoencoder task that predicts the gene distribution at each spot [6]. DeepST treat K as the "expected" number of domains. If the expectation does not match the data it will also ignore it.

CLGraph CLGraph leverages contrastive learning. It first generates image embeddings of patches near each spot via contrastive learning on the H&E image. It then applies Graph Contrastive Learning to create spot embeddings, using a graph formulation similar to DeepST. Finally, these spot embeddings are clustered to define domains [5]. CLGraph passes K into the clustering algorithm and finds exactly K clusters each time.

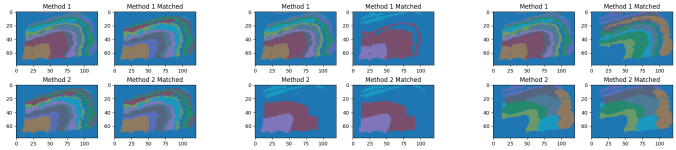
3.2 Analysis Methods

Multi-Region Dice Multi-Region Dice is an extension of the Sorenson Dice coefficient for Multi-Region probabilistic segmentations [13]. It consists of 3 components

1. **The Agreement metric and Cost**. in the paper they leverage the Aitchinson distance. However, since our segmentation masks are crisp rather than fuzzy we use the traditional Sorenson Dice coefficient. Between each pair of masks S_1, S_2 we calculate the cost as $C_{1,2} = 1 - Dice(r_1, r_2)$ for $r_1 \in S_1$ and $r_2 \in S_2$

2. **Bi-partite Graph Matching Algorithm**. The pair of segmentation masks S_1, S_2 are treated as a bi-partite graph $G = (V, E)$ where for each distinct region $r_i \in (S_1 \cup S_2)$ $r_i \in V$. Furthermore, edges only exists from some $s_i \in S_1$ to $s_j \in S_2$ or the reverse. Each edge is weighted by $w_i = C_i$ for all $e_i \in E$. The Hungarian Algorithm finds the sub graph $O \subseteq G$ for which the cost $C = \sum_{i=0}^{|E|} w_i$ is minimized.
3. **Region Merging and Splitting** in the original Multi-Region Dice the score is defined on an automatic segmentation and a ground truth segmentation. Since the Hungarian Algorithm finds a 1 to 1 mapping between regions it ignores possible over and under segmentation's. Originally the paper considers an Unmatched set U. Then for each $u \in U$ the super-region $sr = u \cup r_i$ is created and compared with the corresponding $r_j \in S_2$. The region r_i for which the sr has the highest score is merged with u. However, if u is from the ground truth segmentation the region r_j is split into two regions such that $r_{j1} = r_j - r_j \cap u$ and $r_{j2} = r_j \cap u$.

Extension to pairs of automatic segmentation. Since we want to compare differences between methods rather than each method to a ground truth. We need to define how we will merge and split without this knowledge. If we always merge to the larger object the metric will remain symmetric which is our goal. In, short for my implementation I add any unmatched region $u \in U$ to the region for which the score between sr and r_j is highest. Some initial results of the mapping are shown in figure 4.



(a) DeepST K=2 vs DeepST K=3 (b) DeepST K=2 vs CLGraph K=3 (c) DeepST K=2 vs CLGraph K=8

Fig. 4: Comparison of Mapping Output of Multi-Region Dice

Normalized Mutual Information Normalized Mutual Information calculates the entropy of the pairwise histogram of all image intensities. If images have large counts of paired intensities, this means there are large homogeneous regions of intensity in common across the images. Consider, for example, a pair of homogeneous regions where in region 1 the intensity is $i = 5$ and in region 2 the intensity is $i = 7$. Following the entropy formula

$$H = -\sum_{x \in X} p(x) \log(x), \tag{1}$$

in our example, since 100% of our intensity pairs are $(5, 7)$, their probability is 1. Therefore, the entropy of $(5, 7)$ is

$$H((5, 7)) = -1 \cdot \log(1)$$
$$H((5, 7)) = 0$$

We can see that a homogeneous region has low entropy. Furthermore, if we plot information, defined as $I = -p(x)\log(p(x))$, we observe that both low and high probabilities result in less information (see Figure 5). Since entropy is the sum of information, it is intuitive that homogeneous regions have low information and, consequently, low entropy. Therefore, if our methods are similar they should

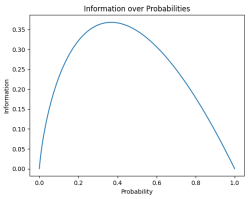


Fig. 5: Information at different probabilities

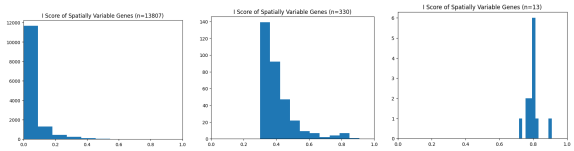
have low entropy and Mutual Information

4 Results

4.1 Gene Distribution across Domains

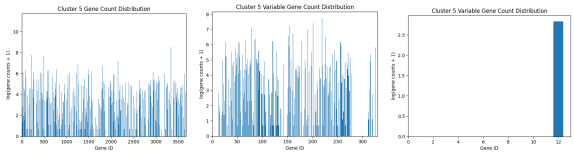
Because SDI has many downstream applications, I chose to analyze the gene distribution per cluster directly. This introduced unique challenges due to the technology: Visium and other ISS methods have relatively high read depth, often detecting over 3,500 genes per domain. While this resolution is valuable, not all genes are equally relevant for disease progression or SDI tasks. For example, BRCA1 and BRCA2 are key indicators of breast and ovarian cancer, with harmful mutations increasing cancer risk nearly fourfold [14]. Many other genes, however, are much less important for disease progression. Since my goal was to examine how gene distributions vary across domains, I focused on spatially varying genes. To identify them, I used Moran's I [15], a spatial autocorrelation statistic ranging from –1 (evenly dispersed) to 1 (clustered).

I thought it was interesting that the distribution of spatial variability was very right skewed across the first 2 thresholds of figure 6 but the 3rd looks more normal. Setting the threshold at 0.7 seemed tempting especially since variation in gene distribution across clusters was visible with that setting. However, statistical power was crippled because only 13 genes had a spatial autocorrelation



(a) Threshold = 0 (b) Threshold = 0.3 (c) Threshold = 0.7

Fig. 6: Distribution shape of I statistic



(a) Threshold = 0 (b) Threshold = 0.3 (c) Threshold = 0.7

Fig. 7: Gene Distribution with Different Levels of Variability

higher than 0.7 and only 47 were higher than 0.5. I chose to continue my analysis with the panel of genes created by a threshold of 0.3 giving me 330 genes to consider.

4.2 Intra-Method Variation

For analysis of the segmentations we begin by comparing segmentations which belong to the same method. By varying our hyper parameter K we can create different segmentation masks. For these experiments I varied K from 1 to 8 for each method. Below in figures 8 910 I show the resulting segmentation masks for each method. While each method creates different segmentation masks depending on K. DeepST seems to not treat K like a hard parameter. When it agrees with the methods internal logic the K is respected however when I set K=2 or K=3 the resulting segmentation mask had 12 distinct regions! CLGraph seems to treat K as a hard hyper parameter. Qualitatively the shape of the regions in CLGraph seem to vary much more randomly. DeepST seems to just add the layered regions. How that difference behaves in the gene distribution will be discussed later. Quantitatively, we can discuss the differences in the segmentation masks using both Multi-Region Dice and Normalized Mutual Information. For visualization I have created a matrix to visualize the pairwise average Dice values on the bottom triangle and the pairwise standard deviation in Dice in the upper triangle. For the Normalized Mutual Information I show only the bottom triangle as it's a global measure and does not have variation across regions. Speaking globally, some reasonable patterns appear. When K is similar, for example: $K = 8$ and $K = 9$ in Figure 11a. The resulting Dice is

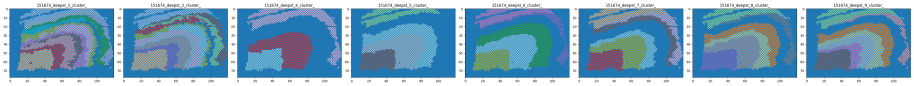


Fig. 8: DeepST segmentations Resulting from varying K

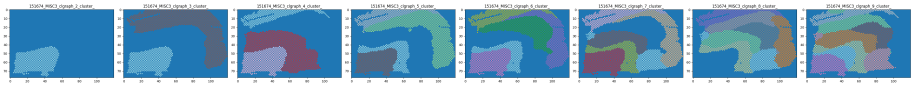


Fig. 9: CLGraph segmentations Resulting from varying K

high (0.9), and the standard deviation is low (0.3). DeepST, however, also is an outlier. For instance, in the $K = 2$ and $K = 3$ segmentation masks there exist more than 10 regions, so the similarity scores between $K = 3$ and $K = 4$, and between $K = 2$ and $K = 4$, are very low (0.2 and 0.3, respectively). In these cases, the standard deviation is 0.4, which is higher than the average score. That makes sense since in reality there's a large difference in reulting domains

Globally, CLGraph also yields lower scores, which seems to quantitatively support the idea that the shapes of its domains change more than those of other methods. For SpaGCN, we observe that average scores are consistently lower. However, between $K = 6$ and $K = 7$, we obtain a perfect match, which is particularly interesting. Since SpaGCN does not treat K as a hard parameter but rather as a seed for searching optimal clusters, it effectively recreates the domains from the more optimal K. I would attribute the generally lower scores to noise in the domains. The result from the Normalized Mutual Information was particularly interesting. Even though visually the SpaGCN domains were less continuous and were noisier. Across the hyper parameter sweep it has the lowest Normalized Mutual Information and therefore the most consistent regions at different numbers of K. For this discover, we can infer that SpaGCN is more likely to split existing regions than to create a new set of completely different regions. Even though the regions in DeepST seem to have the most similar shape we can see that they actually change the most using the NMI. Furthermore, for K=2 and K=3 when the number of regions was completely different we can see it's very dissimilar from all other methods. A behaviour in common between SpaGCN and CLGraph is that K=8 and K=9 are very different from other

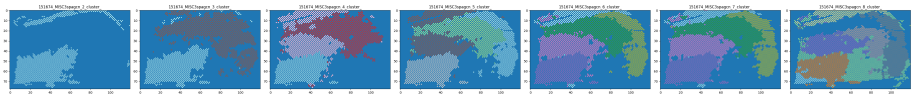
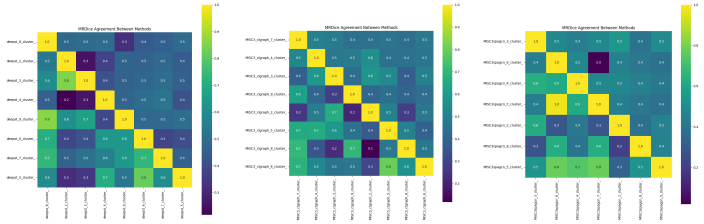
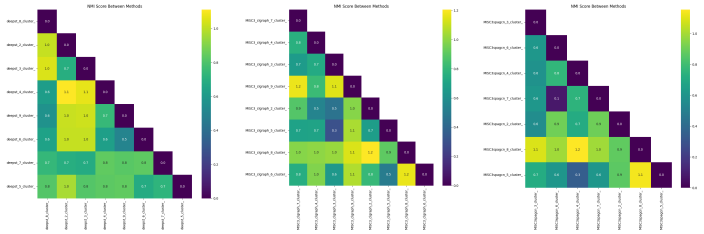


Fig. 10: SpaGCN segmentations Resulting from varying K



(a) DeepST Multi-Region Dice Similarity Matrix (b) CLGraph Multi-Region Dice Similarity Matrix (c) SpaGCN Multi-Region Dice Similarity Matrix

Fig. 11: Intra Method Multi-Region Dice Similarity Matrix



(a) DeepST NMI Similarity Matrix (b) CLGraph NMI Similarity Matrix (c) SpaGCN NMI Similarity Matrix

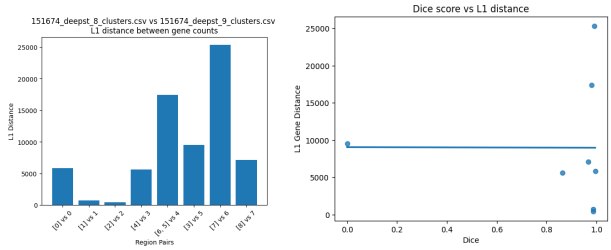
Fig. 12: Intra Method NMI Similarity Matrix

hyper-parameters. From looking at the masks. It seems that for both methods the masks is strongly changing the shape of domains to fit the higher number of domains. Indicating that K may be too high for that sample. For analyzing Inter-Methods Variation I aim to choose a reasonable value of K for each method. For, Deepst K = 8 has both a low NMI score and a High Multi-Region Dice across other values of K. For CLGraph I choose K=6 and for SpaGCN K=5 with the exact same rationale.

4.3 Intra-Method Gene Exploration

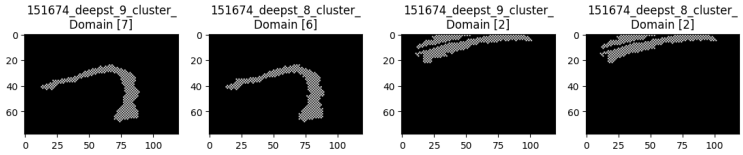
I found this behaviour when exploring the different gene distributions per domain for DeepST. I consider the mask pair with the highest similarity K=8 and K=9. You can see the two masks are very similar in figure 8. For each domain that was matched by the Multi-Region Dice algorithm, I indexed all spots belonging to that domain and summed the total gene expression for my panel of spatially varying genes. I was interested in how the similarity between the shape of the domains and the gene distribution related. For each optimal match I calculated the L1 distance between the 2 distributions. defined as $L1 = \sum_{x,y \in (X,Y)} abs(x - y)$. The L1 distance between the optimal matches is shown in figure 13a. There is

a large variation in this distance anywhere from less than 5000 to over 25000. So I was interested in which domains had high differences and which ones had low differences. In figure 14a we see that a highly similar domain has a very low L1



(a) L1 Gene distance between matched domains

(b) Relationship Between Dice and L1 Gene Distance



(a) Domain 7 and 6 with Dice Similarity of 0.97

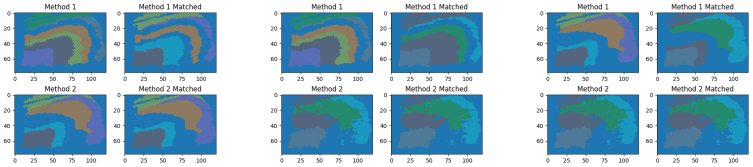
(b) Domain 2 with Dice Similarity of 0.98

distance but figure 14b that a different pair of domains also with high similarity have a high L1 distance. In 13b we see that there is little relation between dice and gene distribution distance for that pair of methods. That finding is not generalizable. However, it is interesting as it goes against my intuition on the spatial domains.

4.4 Inter-Method Variation

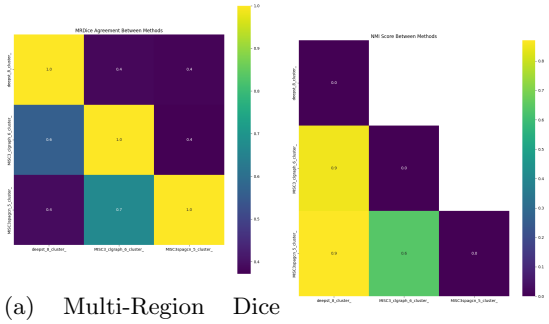
I repeated the previous steps in Intra-Method variation across the 3 optimal parameter values. To see which methods are most similar, I use the Multi-Region Dice and Normalized Mutual Information again in figure 16. I also show the Multi-Region Dice Mapping Between the 3 pairs of methods in figure 15. We can see the CLGraph is more similar to both other methods. Two things are immediately interesting about that.

1. K for CLGraph is between the K values for SpaGCN and DeepST. Therefore, similar values of K create consistent domains across methods



(a) DeepST(1) vs CLGraph(2) (b) DeepST(1) vs SpaGCN(2) (c) CLGraph(1) vs SpaGCN(2)

Fig. 15: Inter Method Multi-Region Dice Mapping



(a) Multi-Region Dice Matrix (b) NMI Matrix

Fig. 16: Inter Method Similarity Matrices

2. CLGraph creates domains that are both similar to DeepST and SpaGCN. IE the masks look like some combination between the other methods. Therefore, it has learned something common between them

I would argue for the second option. While it's supported in the Intra-Method comparison that a similar K results in a similar mask. The K is similar across all methods and the largest difference between them is 2 in a parameter sweep from 2-8. Furthermore, I think it's interesting based on the 3 architectures. While SpaGCN considers raw pixel patches as inputs to it's encoding model, DeepST generates spot embeddings using ImageNet[12] encoders trained on everyday images of dogs and cats and other classes and embeddings that in a graph auto encoder to learn representations. CLGraph is similar that it also creates spot embeddings. However, it uses contrastive learning creating custom embeddings for the H&E data rather than unaligned embeddings like DeepST. Furthermore, it generates node embeddings via contrastive learning similar to the autoencoder DeepST uses. However, it uses the gene distribution as input rather than supervision. I find it interesting that a method that combines aspects of both other methods outputs segmentation masks that seem to be a combination of both methods.

5 Accomplishments

In this project, I learned a great deal. Setting up multiple methods with different hyperparameters was challenging—partly due to poor or missing documentation, and partly because reasoning across methods and hyperparameters together quickly became overwhelming. To address this, I separated my analysis into intra- and inter-method variation. For example, it's difficult to compare DeepST with K=2 to SpaGCN with K=8 because multiple variables influence the outcome. Developing a structured way to compare across these variables took time, but it ultimately allowed me to identify and interpret quantitative similarities between methods.

6 Conclusion and Discussions

In short, I have built a pipeline to compare the outputs of different SDI methods quantitatively on the spatialLIBD dataset. I've discovered finding that challenge the way I understood spatial domains intuitively and I've been able to find methods that are more different and more similar to each other and reason about why they are. A major drawback of this type of analysis is that it's pairwise. I can only match region between sets of 2 methods. However, an interesting analysis would be is there a domain common across all methods? Which domains are segmented first as K is increased? Is a domain that was segmented at a lower K have a more distinct gene distribution than one segmented later?

7 Future Work

We can model the correspondence between regions from different SDI methods as a weighted k-partite graph G. Each partition represents a method, each vertex a region, and edge weights are the Dice similarity between regions. A cycle in G defines a global mapping, since each region in the cycle is maximally similar to the next, implying mutual similarity across the set. Even without a cycle, a path can be used to identify a mapping that maximizes similarity across methods. This relates to work in computational geometry on spanning trees in multipartite graphs [16], which could be applied.

Acknowledgements

Thank you to SpatialLIBD for the data, Hest1K for packaging it and the base implementation for Multi-region Dice from Professor Hamarneh.

Appendix

code is available for the project on the Project Github

References

1. C. G. Williams et al. An introduction to spatial transcriptomics for biomedical research. *Genome Medicine*, 14(1):68, 2022.
2. Y. Wang et al. Spatial transcriptomics: Technologies, applications and experimental considerations. *Genomics*, 115(5):110671, 2023.
3. National Genomics Infrastructure (NGI), Sweden. Visium for fresh frozen samples (poly-a based). NGI Sweden – SciLifeLab Methods webpage, 2025. Last updated: 10 February 2025.
4. J. Huang et al. A graph self-supervised residual learning framework for domain identification and data integration of spatial transcriptomics. *Communications Biology*, 7(1):1123, 2024. Received Jan 20 2024; Accepted Aug 30 2024; Published Sept 12 2024.
5. S. Du et al. Clgraph: A contrastive learning-based multimodal fusion approach for spatial domain identification. In *2024 IEEE International Conference on Medical Artificial Intelligence (MedAI)*, pp. 201–211, 2024.
6. C. Xu et al. Deepst: identifying spatial domains in spatial transcriptomics by deep learning. *Nucleic Acids Research*, 50(22):e131, 2022.
7. J. Hu et al. 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, 2021. Received 1 December 2020; Accepted 29 July 2021; Published online 28 October 2021.
8. K. Coleman et al. Spadecon: cell-type deconvolution in spatial transcriptomics with semi-supervised learning. *Communications Biology*, 6, 2023.
9. K. R. Maynard et al. Transcriptome-scale spatial gene expression in the human dorsolateral prefrontal cortex. *Nature Neuroscience*, 24(3):425–436, 2021. Received: 12 June 2020; Accepted: 18 December 2020; Published: 8 February 2021.
10. G. Jaume et al. Hest-1k: A dataset for spatial transcriptomics and histology image analysis, 2024.
11. AnnData Developers. Anndata: Annotated data structures for single-cell genomics, 2025. Accessed: 2025-08-16.
12. O. Russakovsky et al. Imagenet large scale visual recognition challenge, 2015.
13. S. Andrews and G. Hamarneh. Multi-region probabilistic dice similarity coefficient using the aitchison distance and bipartite graph matching, 2015.
14. National Cancer Institute. *BRCA Gene Changes: Cancer Risk and Genetic Testing Fact Sheet*. National Cancer Institute, 2025. Accessed: 2025-08-16.
15. Esri. *How Spatial Autocorrelation (Global Moran's I) works*. Esri, 2025. Accessed: 2025-08-16.
16. A. Biniaz et al. Spanning trees in multipartite geometric graphs. In *Spanning Trees in Multipartite Geometric Graphs*. Carleton University, 2017. Presented April 28, 2017.