Curvature informed neighboring graph trimming facilitate single cell data analysis

Qirui Guo[1†], Kun Qian[1†], Zhigui Wu[1†]

[1*]Center for quantitive biology, Peking University, Street, Beijing, China.

Contributing authors: qirui.guo@stu.pku.edu.cn;
2301110076@stu.pku.edu.cn; 2201111931@stu.pku.edu.cn;
[†]These authors contributed equally to this work.

Abstract

Many of the scRNA data analysis tools rely on the neighboring graph from expression data and size of neighboring graph dramatically affects the efficiency and performance of these tools. In this article, we propose a curvature-based neighboring graph trimming method to balance computational efficiency and performance in clustering and trajectory inference. By pruning less informative edges using Olliver-Ricci curvature, we maintain or improve results with sparser graphs. Tested on four simulated and one real-world T-cell dataset, our method preserved or enhanced clustering and trajectory inference outcomes. Minimal curvature trimming improved clustering homogeneity, while maximal trimming enhanced trajectory inference. Despite challenges in curvature computation and its impact on discrete graphs, our approach offers a novel, geometry-based method for adaptive graph construction in large-scale datasets.

Keywords: scRNA-seq, Neighboring Graphs, Graph Trimming

1 Introduction

In the past decade, single-cell RNA sequencing techniques have emerged as a pivotal method for expression profiling and cell state identification in biomedical research. Besides the development of these sequencing methods, researchers have also proposed various analytical methods to decipher biologically significant information within these data, including visualization, clustering, and trajectory inference. For instance, t-SNE[1] and UMAP[2] enable the representation and visualization of high-dimensional

data, while Louvain and Leiden are introduced as clustering methods to identify sample categories. As for trajectory inference, several tools like scVelo[3], Dynamo[4], DPT[5], and TIGON[6] rank samples along a pseudo-developmental hierarchy and have successfully revealed various critical issues such as immunological cell differentiation and fate decisions in developmental biology. Despite their wide applications, most of these methods rely on a k-nearest neighbors (KNN) graph within the expression space to provide a basis for sample relevance or interaction evaluation. For example, to maintain the consistency of distance between samples and their neighbors in both the expression space and the latent space, UMAP constructs such a KNN graph to define the neighborhood of a data point. Important as it is, neighboring graph construction has always been regarded as a preprocessing step and neglected for simplicity, with only the number of nearest neighbors k left as a hyper-parameter. Facing the trade-off between tool performance and computational efficiency, the selection of such a hyper-parameter highly depends on some rule of thumb given data properties and downstream tasks.

Recently, many computational tools has been developed to refine the construction of neighboring graph. These methods could be broadly categorized into two classes: expression augmentation methods and graph denoising methods. Expression augmentation methods like BANKSY[7] extend the original expression data to some additional information and thus alternate the data distribution, which further change the structure of neighboring graph. **TBC:graph denoising**

In this article, we proposed a novel curvature based neighboring graph trimming tools. By leverage the concept of discrete Olliver-Ricci curvature [8, 9] on graph, we equip each edges e_{ij} in the original graph $G = (V, E)$ a scalar curvature to evaluate the deviation of neighborhood of sample v_i from euclidean space in the direction to v_j, which could then reflects on the "information" of the edge. We then construct a **TBC:model implementation**. It can be shown that introduction of curvature impressively improve the downstream analysis including clustering and trajectory inference on both simulated datasets and real datasets even with naive sorting and pruning of edges. Furthermore, by combining such geometry properties and **TBC: model results**

2 Results

2.1 Method Overview

In this work, we propose a neighboring graph trimming method based on the flatness of the original neighboring graph $G = (V, E)$, which can be represented by the discrete Ollivier-Ricci curvature. The input for this tool is a pre-computed neighboring graph with pairwise Euclidean distances as the weights of the graph. Throughout the trimming pipeline, we first construct a transition probability matrix P for a random walk on the graph. We then compute the Wasserstein distance between the transition measures $m_i$and m_j of nodes v_i and v_j to represent the change of neighborhood on the graph. The Ollivier-Ricci curvature is then computed using the shortest path on the graph d_{ij} as follows:

$$ric_{ij} = 1 - \frac{W(m_i, m_j)}{d_{ij}}$$

Generally speaking, the curvature on edge e_{ij} represents how the neighboring sets of a point change when transporting the node from v_i to v_j. As showcased in Figure 1(B), edges with higher $|ric_{ij}|$ indicate that the neighborhood of v_j is significantly different from that of v_i, while lower $|ric_{ij}|$ implies that the neighborhoods of nodes v_i and v_j are almost "equivalent." Given the informative nature of this curvature, we can rank all the edges of a sample and select only the most important ones. For instance, in clustering tasks that seek the homogeneity of samples within a cluster, we can neglect edges with higher curvature to focus on neighbors with similar neighborhoods. Conversely, for trajectory inference tasks, we preserve these high-curvature edges to capture transitions between clusters. Additionally, we introduce these geometric properties into XXX.

2.2 Curvature based trimming refines clustering

We first applied our method to two simulated spiral datasets. In the closed spiral dataset, the sample points formed a closed, worm-like chain in the original space. An optimal clustering algorithm should be able to distinguish between points on different "circles." Initially, we performed Louvain clustering on the naive neighboring graph with varying neighborhood sizes but the same resolution. As shown in the PCA visualization in Figure 2(B), Louvain clustering on the larger graph ($k = 30$) successfully identified 6 clusters from the data. However, due to the narrow neighborhood size and lack of global information, clustering on the smaller neighboring graph ($k = 10$) failed to merge points from the same circle, resulting in 12 clusters. We also visualized the neighboring graph and marked each edge with computed scalar curvature. Alongside the circle, we observed a band-like structure of edges with higher (yellow) and lower (blue) curvature. Next, we performed curvature-based trimming by selecting the highest (or lowest) 10 edges from the larger graph and clustered the data based on the trimmed graph. As depicted in Figure 2(B), minimal curvature trimming dramatically eliminated edges between adjacent circles while preserving all low curvature edges within the circles. Consequently, with only 10 edges, clustering on such a minimally curvature-trimmed graph achieved almost the same performance as clustering on the larger graph. In contrast, maximal curvature trimming, which aimed to preserve edges between circles, did not align with the clustering objective, resulting in relatively poor performance compared to minimal trimming. We also tested our methods on a galaxy-like dataset. In this dataset, samples are distributed across four branches, and the optimal clustering method should identify homogeneity within and heterogeneity between branches. As shown in Figure 2(D), clustering on the larger graph could not distinguish between samples from different branches far from the centroid due to the abundant connections between branches. Meanwhile, clustering on the smaller graph did identify branches but fragmented each arm into several clusters due to the lack of global information. Surprisingly, when trimmed with minimal curvature criteria, the distinction between branches and the consistency of samples within the same branch

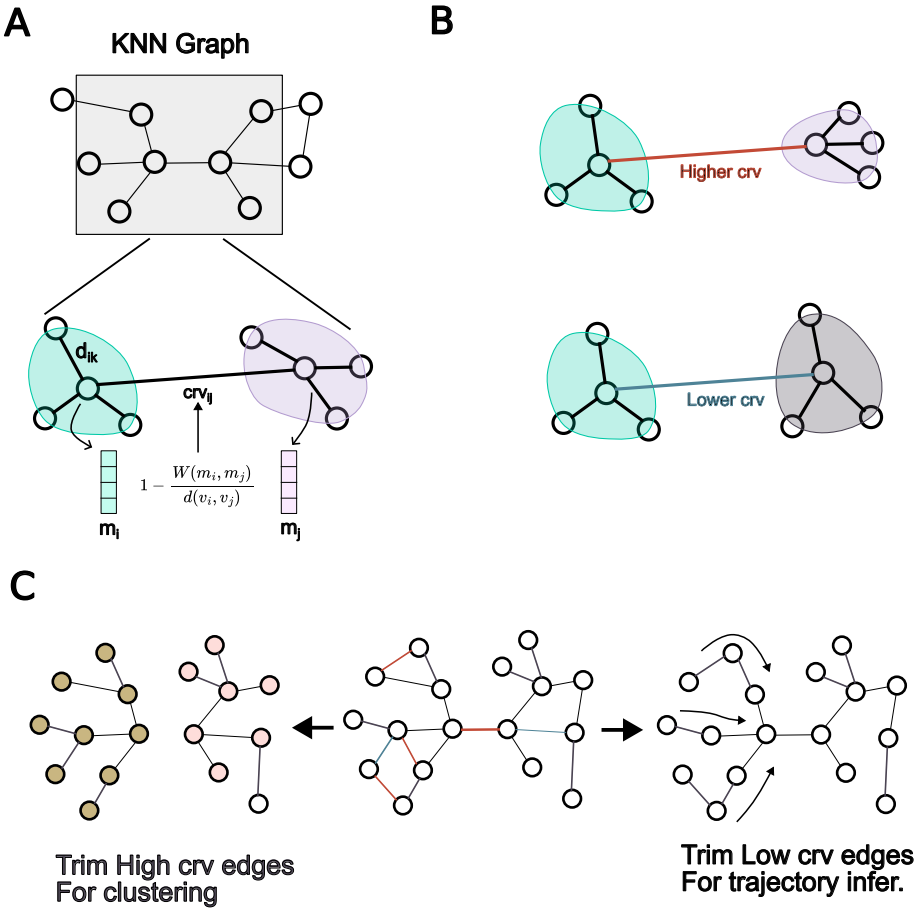

Fig. 1 Overview of curvature based neighboring graph trimming

were partially recovered compared to the naive graph with higher or lower neighborhood sizes. These results from the simulated datasets highlight the significance of the geometric properties of the neighboring graph in clustering and underscore our tool's ability to refine clustering tasks using limited yet informative subgraphs.

2.3 Curvature based trimming improves trajectory inference

To validate the effectiveness of our methods in facilitating trajectory inference tasks, we applied them to both simulated datasets and real-world T cell linkage data. For the simulated datasets, we generated typical trajectory simulations following linear and bifurcation schemes (Methods) and compared the computed diffusion pseudotime with the ground truth simulation results. In the linear dataset, where samples approximate

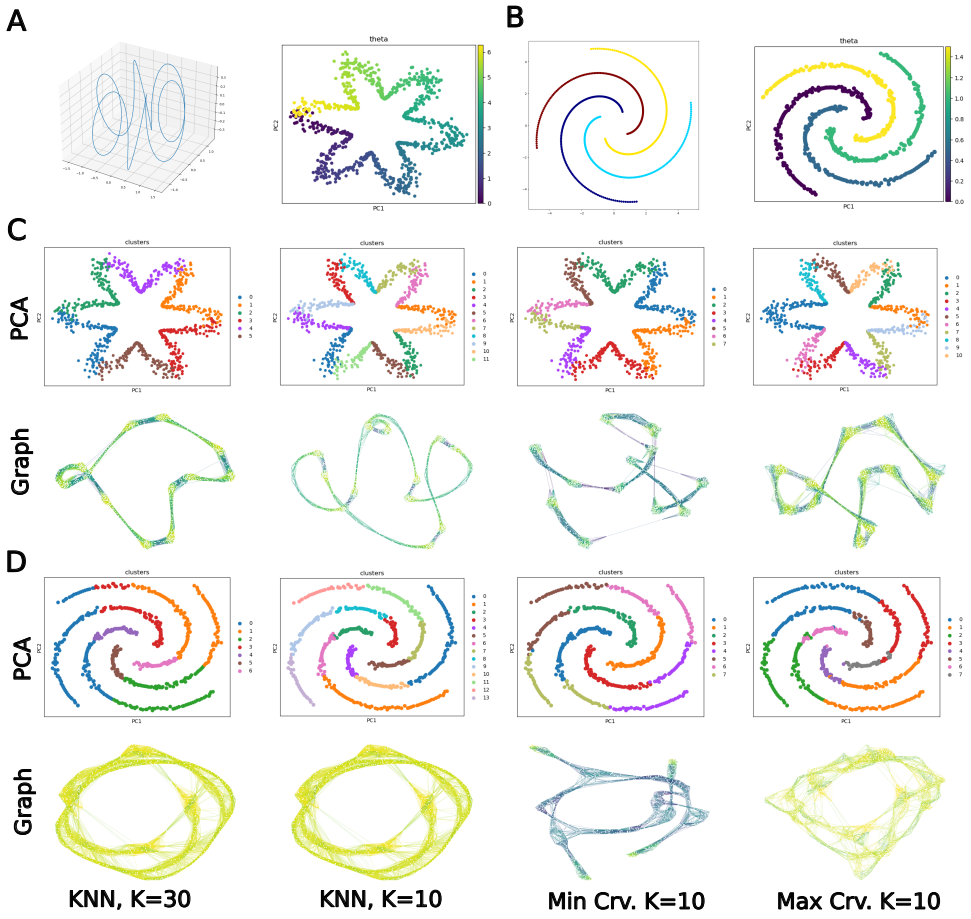

Fig. 2 Curvature based trimming refines clustering on simulated datasets

a continuous smooth manifold (Figure 3(A)), the inference results using a neighboring graph with a large neighborhood size (Figure 3) showed strong consistency with the actual simulation time. However, when we reduced the neighborhood size from 30 to 10, the computed developmental pseudotime exhibited significant divergence from the ground truth. We further illustrated the clustering and PAGA results in Figure S2 and found that reducing the neighborhood size dramatically affected both clustering and trajectory inference tasks. Subsequently, we applied curvature-based trimming to the original large graph and compared both pseudotime (Figure 3) and clustering (Figure S2) results with those from the naive neighboring graphs. As demonstrated, with a small neighborhood size, trajectory inferred on the graph trimmed by the maximal curvature criterion—where most edges between different clusters are preserved—achieved results comparable to those on the full graph. Meanwhile, the clustering results further underscore our method's refinement in clustering tasks by pruning edges with

high curvature. Similarly, we tested our methods on the bifurcation dataset, which is crucial for modeling cell fate decisions in trajectory inference. It was observed that, compared to results inferred on graphs with higher k, the pseudotime computed on graphs with lower neighborhood sizes was less convincing and showed little consistency with the ground truth. Fortunately, by pruning edges with trivial curvature within clusters (Figure S2), the max-curvature trimmed graph successfully supported accurate pseudotime predictions (Figure 2) with only one-third of the information on the graph.

We then moved on to a real-world dataset containing 3505 samples within the T-cell developmental lineage to demonstrate our method's capability in accelerating trajectory inference tasks. The dataset is sampled from a post-transplantation T-cell differentiation lineage from *BALB/c* mice and mainly includes exhausted T (Tex) cells and their precursor Tpex cells (Figure 2E). By performing RNA-velocity inference, the trend of differentiation from Tpex to Tex cells has been identified. Here, we aim to replicate this trajectory using Diffusion Pseudotime (DPT) with the different graphs mentioned earlier. As presented in Figure 2F, DPT results on graph with a sufficient neighborhood size reveal the transition from Tpex to Tex samples, while pseudotime on a graph with a narrow neighborhood size collapses. However, by trimming the graph using the maximal curvature criterion, reasonable pseudotime inference results are achieved even with less edge information. It is also noteworthy that although graphs trimmed by minimal curvature do not provide convincing trajectory data, the clustering results based on such graphs indicate the existence of a novel subgroup within the data. Differential expression analysis of this newly identified cluster further suggests higher expression of the genes *CCL5* and *Trbv3*, which are target expressions of effective T cells. These results further underscore the capability of curvature-informed trimming to facilitate downstream analysis across different topics and contribute to the ongoing advancement of single-cell research.

3 Methods

3.1 Oliver-Ricci Curvature

In differential geometry, curvature is a measure of the deviation of a geometric object such as a surface or curve from being flat or straight. In the context of graph theory, curvature concepts are adapted to study the shape and structure of graphs. Curvature on graphs provides insights into the graph's connectivity, robustness, and overall topology. Scalar curvature is a more specific type of curvature used primarily in the study of Riemannian manifolds, which are smooth surfaces. In these manifolds, scalar curvature describes how the volume of small balls in the manifold deviates from those in Euclidean space. Translating this concept to graphs, scalar curvature can be thought of as a measure that captures the local clustering or neighborhood density around a vertex[8]. It reflects how tightly or loosely connected the neighborhood of a vertex is, compared to a regular lattice or tree structure.

Ollivier-Ricci curvature is a discrete analogue of Ricci curvature, originally defined for Riemannian manifolds. Ricci curvature provides a way to measure the extent to which the volume of small geodesic balls in a manifold deviates from that in the

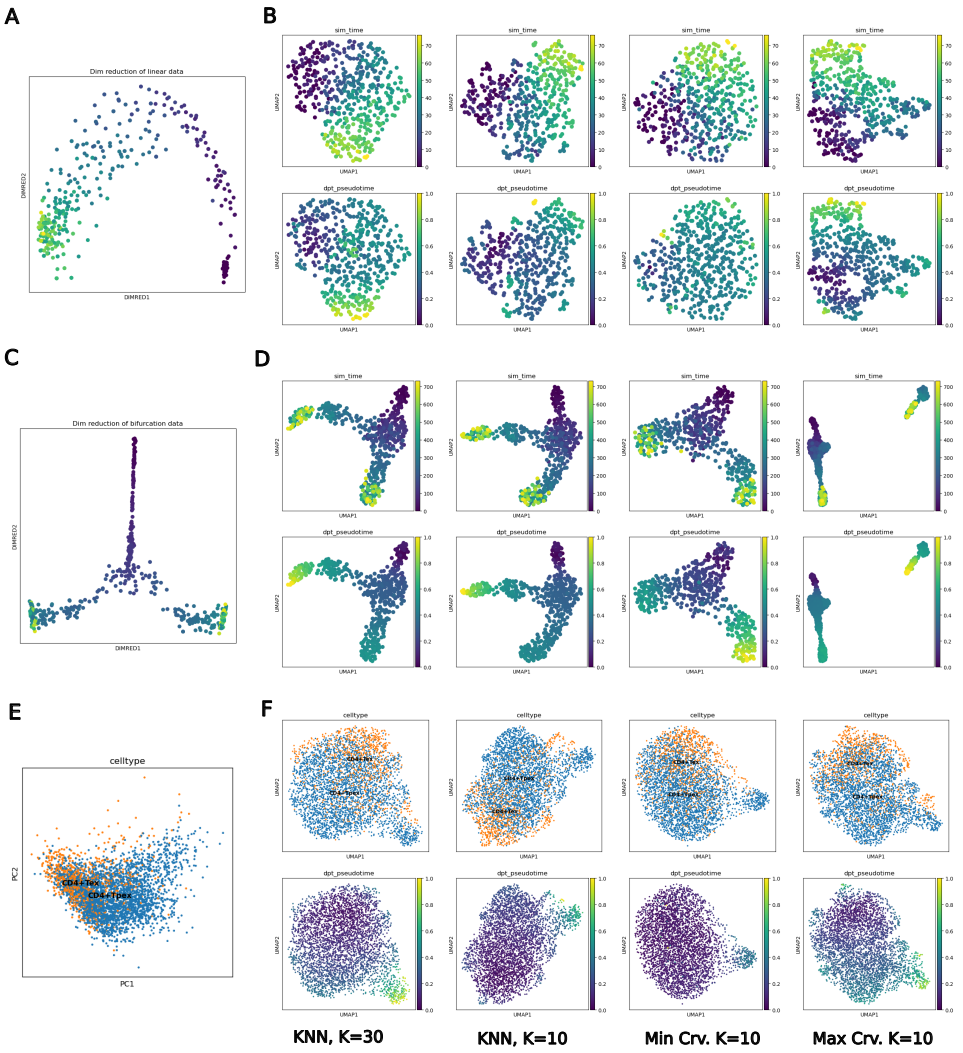

Fig. 3 Curvature based trimming improves trajectory inference

Euclidean space, averaged over all directions. In the discrete setting of graph theory, Ollivier-Ricci curvature measures the connectivity and distribution of shortest paths around nodes and edges in a graph. Consider a weighted graph $G = (V, E)$ with vertex set V and edge set E. The Ollivier-Ricci curvature $ric(i, j)$ between two adjacent vertices i and j is defined as:[9]

$$ric(i, j) = 1 - \frac{W(m_i, m_j)}{d(i, j)} \tag{1}$$

where $d(i,j)$ is the on graph shortest distance between v_i and v_j, and $W(m_i, m_j)$ is the Wasserstein distance (or earth mover's distance) between the measure m_i and m_j on node respectively. Given our aim to evaluate change of neighbors, we set m_i as transition measure of a random walk on graph where $m_{ij} = \delta_{ij}/d(i,j)$. As for $W(m_i, m_j)$ we apply sinkhorn-knopp's method which solve the optimal transport problem as

$$\begin{aligned} W(a,b) = \min_{\Pi \in \mathbb{R}_+^{m \times n}} \quad & \sum_{i,j} \Pi_{i,j} C_{i,j} \\ \text{s.t.} \quad & \Pi 1 = a; \Pi^T 1 = b; \Pi \geq 0 \end{aligned} \tag{2}$$

where C is the cost matrix with entries C_{ij} denotes distance from v_i to v_j and Π denotes for coupling matrix indicating the mass of transport from a_i to b_j.

3.2 Curvature based Trimming

For the input adjacency distance graph, we first compute the olliver-ricci curvature for each valid edge. Given that $ric_{ij} = 0$ means the neighborhood of node i, j is flat along edge e_{ij} and $ric_{ij} < 0, ric_{ij} > 0$ denotes different kinds of torsion of data manifold (sphere for positive and hyperbolic for negative respectively), during trimming we first rank all edges $e_{i,\cdot}$ of node i by the absolute value of the computed curvature and select the top or bottom edges under different scenario.

3.3 Simulated Data Generation

Here we briefly introduce simulated dataset used in this article. Two spiral datasets are generated by polar curves: For closed spiral dataset [10] we sample 10^3 points $\{x_i\}$ evenly spaced with respect to arc-length along the spiral parameterized by

$$\begin{bmatrix} x(t) \\ y(t) \\ z(t) \end{bmatrix} = \begin{bmatrix} \cos(t)(0.5\cos(6t) + 1) \\ \sin(t)(0.4\cos(6t) + 1) \\ 0.4\sin(6t) \end{bmatrix}, \quad t \in [0, 2\pi]. \tag{3}$$

We then generated a random matrix $R \in \mathbb{R}^{100 \times 3}$ such that $R^\top R = I$, i.e. the columns are orthogonal. Each point $x_i \in \mathbb{R}^3$ was then linearly embedded in the 100-dimensional ambient space with additive noise via $\hat{x}_i = Rx_i + \eta_i$. The noise model we use is

$$\eta_i = \frac{Z_i}{\|Z_i\|}\rho(\theta), \quad \rho(\theta) = 0.05 + 0.95\frac{1 + \cos(6\theta)}{2}. \tag{4}$$

where Z_i follows the standard Gaussian distribution. In order to eliminate the effect of scaling on the effect of the spread parameter ε, we normalised all input data so that $N^{-2}\sum_{i,j=1}^{N} \|\hat{x}_i - \hat{x}_j\|^2 = 1$. As for multi-arm spiral dataset [11] we chose 4 arms started at angles $\theta = \pi/2, \pi, 3/2\pi, 2\pi$, and for each initial angle θ the spiral arm was parameterised as

$$r(t;\theta) = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} t\cos(t) \\ t\sin(t) \end{bmatrix}, t \in [t_0, t_1]. \tag{5}$$

We set $t_0 = 1, t_1 = 5$ and sampled 200 points along each arm to be uniformly spaced with respect to the arc-length of the branch. Following the approach above, the clean data x_i were then embedded into 100 dimensions and noise was added following $\hat{x}_i = Rx_i + \eta_i$ with $\eta_i = (1 - \sin(3t_i)^4)Z_i/\|Z_i\|$.

For simulated expression dataset we use DynGen[12] to generate counts data. For both scheme (Linear/Bifurcation) we generate 500 samples with 50 targets/hks set for each sample. Model backbone and other hyper-parameters are set to be default as in its documentations.

4 Discussion

In this work, we propose a curvature-based neighboring graph trimming method to address the trade-off between computational efficiency and performance in neighborhood graph-dependent clustering and trajectory inference. By pruning edges that are less informative based on computed Olliver-Ricci curvature, we facilitate downstream analysis using a sparser neighboring graph without affecting the final results. Our methods were tested on four simulated datasets and one real-world dataset, revealing that after trimming, clustering or trajectory inference results are preserved or even improved with fewer edges. For clustering tasks, minimal curvature trimming leads to more informative graphs for subsequent algorithms. By pruning edges connecting clusters, we can denoise the original neighboring graph and identify clusters with higher homogeneity. In simulated datasets, minimally trimmed graphs exhibit more compact clustering results. In the real-world T-cell dataset, such trimmed graphs suggest the existence of a novel cluster, which has been validated by differential expression analysis. Conversely, maximal curvature trimming results in better trajectory inference compared to ordinary graphs with narrower neighborhood sizes.

However, several questions remain for further discussion. Firstly, the computation of Olliver-Ricci curvature involves solving an optimal transport problem that requires iterative solvers. For large-scale datasets, this can be time-consuming, potentially more so than direct analysis. Secondly, the contribution of different trimming schemes to downstream analysis has yet to be fully understood. Additionally, the significance of Ricci curvature on discrete data structures like graphs should be examined cautiously.

Despite these challenges, our work provides new insights into neighboring graph trimming by introducing geometric metrics to facilitate downstream analysis. By pruning less informative edges, our method allows for more adaptive neighboring graph construction and the application of algorithms across various fields on large-scale datasets.

Declarations

- Data and code availability: Code of this article could be found in https://github.com/michaelGuo1204/ProjectsComBio

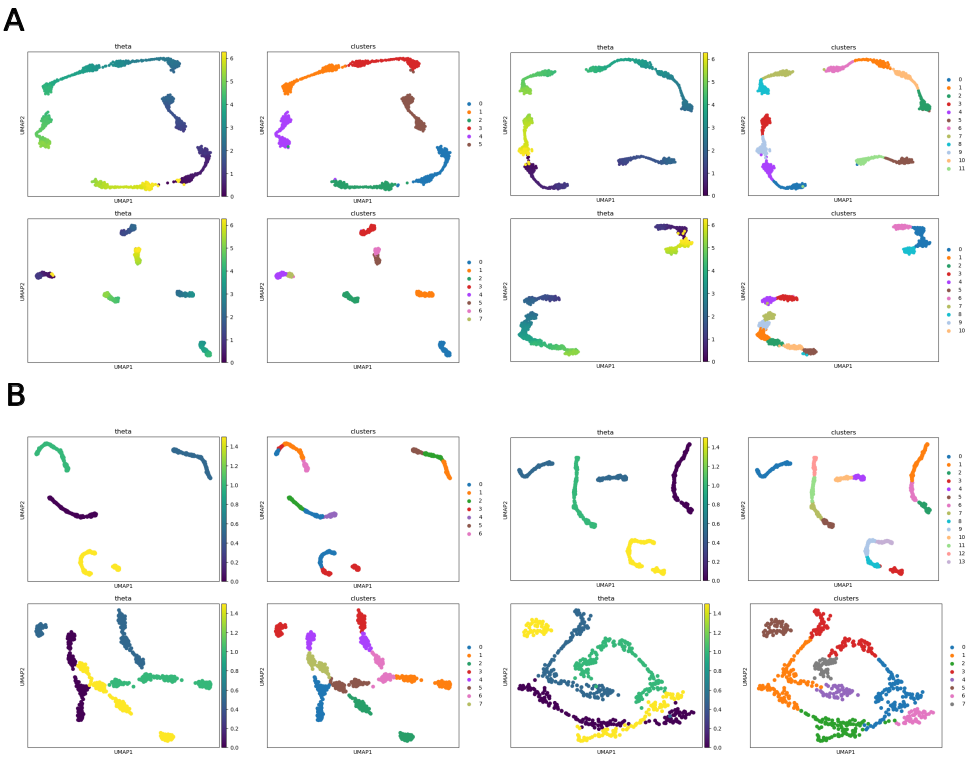

Fig. 1 UMAP visualization of clustering results of spiral datasets given different graph structure

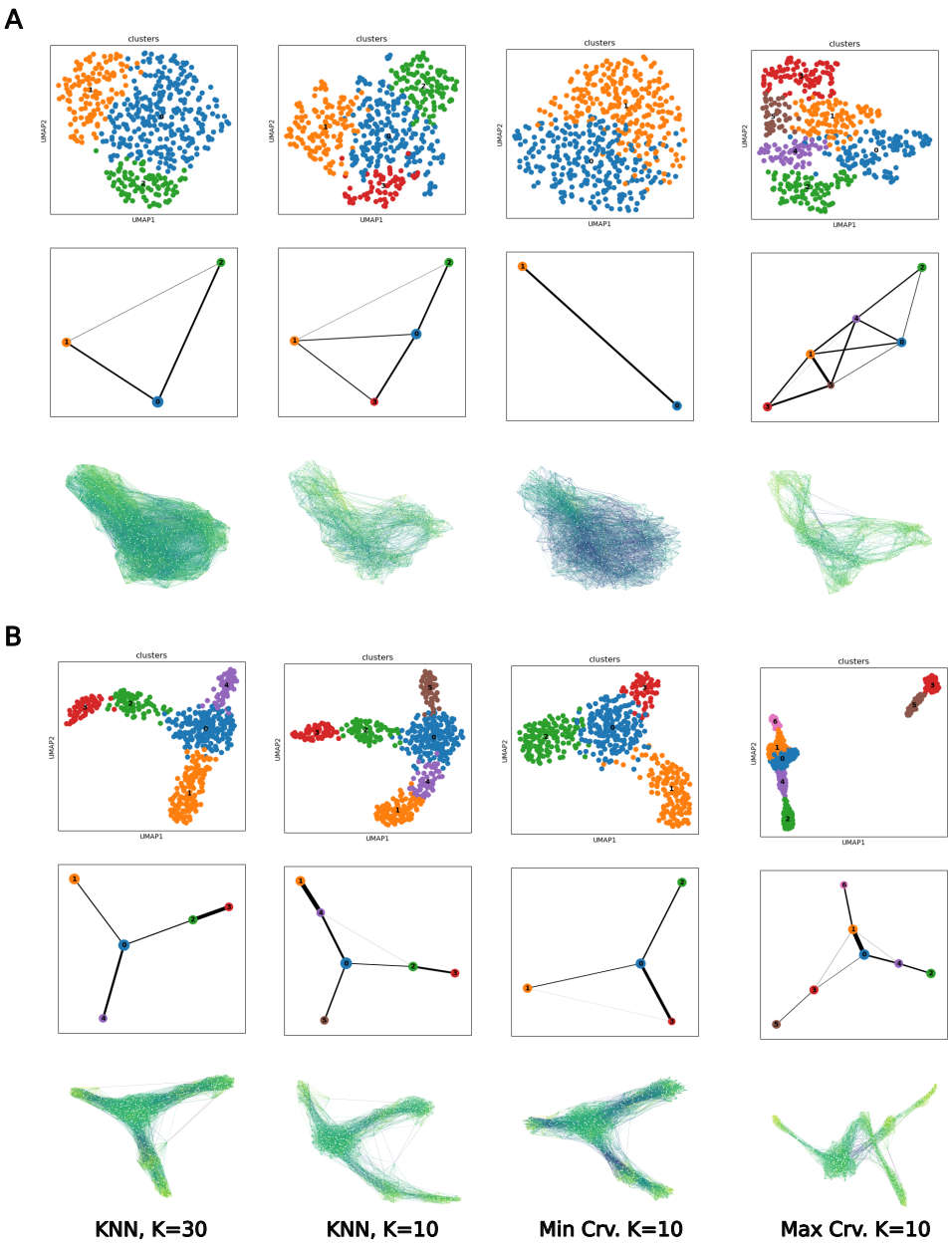

Fig. 2 UMAP visualization of clustering results and PAGA trajectory inference of simulated linage datasets given different graph structure

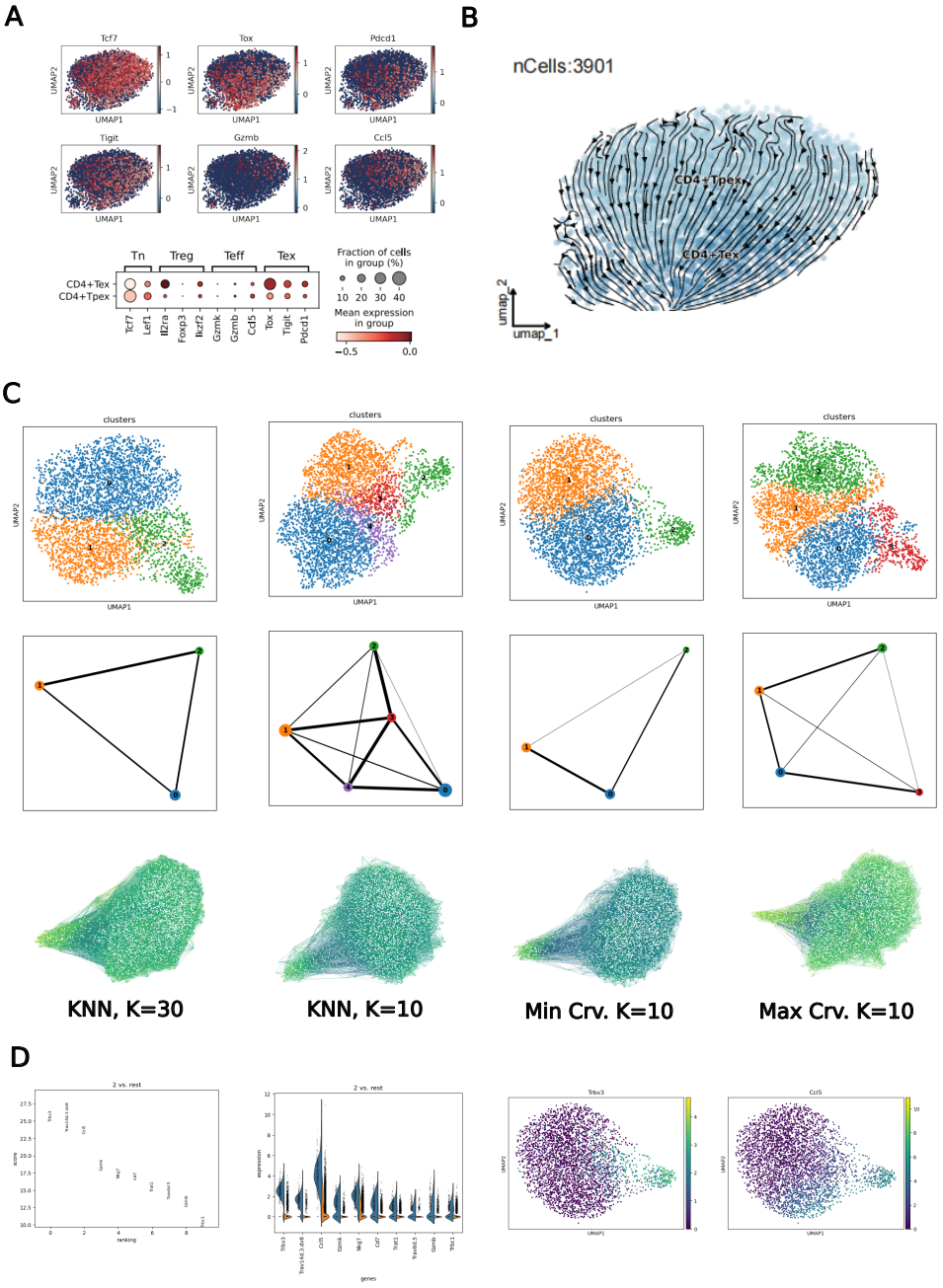

Fig. 3 Real world dataset and its clustering and PAGA inference results.

References

[1] Kobak, D., Berens, P.: The art of using t-SNE for single-cell transcriptomics. Nature Communications **10**(1), 5416 (2019) https://doi.org/10.1038/s41467-019-13056-x . Accessed 2024-06-14

[2] McInnes, L., Healy, J., Melville, J.: UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. arXiv. arXiv:1802.03426 [cs, stat] (2020). https://doi.org/10.48550/arXiv.1802.03426 . http://arxiv.org/abs/1802.03426 Accessed 2024-06-14

[3] Bergen, V., Lange, M., Peidli, S., Wolf, F.A., Theis, F.J.: Generalizing RNA velocity to transient cell states through dynamical modeling. Nature Biotechnology **38**(12), 1408–1414 (2020) https://doi.org/10.1038/s41587-020-0591-3 . Accessed 2024-06-14

[4] Qiu, X., Zhang, Y., Martin-Rufino, J.D., Weng, C., Hosseinzadeh, S., Yang, D., Pogson, A.N., Hein, M.Y., Hoi (Joseph) Min, K., Wang, L., Grody, E.I., Shurtleff, M.J., Yuan, R., Xu, S., Ma, Y., Replogle, J.M., Lander, E.S., Darmanis, S., Bahar, I., Sankaran, V.G., Xing, J., Weissman, J.S.: Mapping transcriptomic vector fields of single cells. Cell **185**(4), 690–71145 (2022) https://doi.org/10.1016/j.cell.2021.12.045 . Accessed 2024-06-14

[5] Haghverdi, L., Büttner, M., Wolf, F.A., Buettner, F., Theis, F.J.: Diffusion pseudotime robustly reconstructs lineage branching. Nature Methods **13**(10), 845–848 (2016) https://doi.org/10.1038/nmeth.3971 . Accessed 2024-06-14

[6] Sha, Y., Qiu, Y., Zhou, P., Nie, Q.: Reconstructing growth and dynamic trajectories from single-cell transcriptomics data. Nature Machine Intelligence **6**(1), 25–39 (2024) https://doi.org/10.1038/s42256-023-00763-w . Accessed 2024-06-14

[7] Singhal, V., Chou, N., Lee, J., Yue, Y., Liu, J., Chock, W.K., Lin, L., Chang, Y.-C., Teo, E.M.L., Aow, J., Lee, H.K., Chen, K.H., Prabhakar, S.: BANKSY unifies cell typing and tissue domain segmentation for scalable spatial omics data analysis. Nature Genetics **56**(3), 431–441 (2024) https://doi.org/10.1038/s41588-024-01664-3 . Accessed 2024-06-14

[8] Samal, A., Sreejith, R.P., Gu, J., Liu, S., Saucan, E., Jost, J.: Comparative analysis of two discretizations of Ricci curvature for complex networks. Scientific Reports **8**(1), 8650 (2018) https://doi.org/10.1038/s41598-018-27001-3 . Accessed 2024-06-14

[9] Siconolfi, V.: Ricci curvature, graphs and eigenvalues. Linear Algebra and its Applications **620**, 242–267 (2021) https://doi.org/10.1016/j.laa.2021.02.026 . arXiv:2102.10134 [math]. Accessed 2024-06-14

[10] Matsumoto, T., Zhang, S., Schiebinger, G.: Beyond kNN: Adaptive, Sparse

Neighborhood Graphs via Optimal Transport. arXiv. arXiv:2208.00604 [cs, stat] (2022). https://doi.org/10.48550/arXiv.2208.00604 . http://arxiv.org/abs/2208.00604 Accessed 2024-06-14

[11] Budninskiy, M., Abdelaziz, A., Tong, Y., Desbrun, M.: Laplacian-optimized diffusion for semi-supervised learning. Computer Aided Geometric Design **79**, 101864 (2020) https://doi.org/10.1016/j.cagd.2020.101864 . Accessed 2024-06-14

[12] Cannoodt, R., Saelens, W., Deconinck, L., Saeys, Y.: Spearheading future omics analyses using dyngen, a multi-modal simulator of single cells. Nature Communications **12**(1), 3942 (2021) https://doi.org/10.1038/s41467-021-24152-2 . Accessed 2024-06-14