Université Libre de Bruxelles

Methods in Bioinformatics
INFO-F-439

Project report
-
Quantifying the clusterness and trajectoriness of single-cell RNA-seq data

Authors:
GRÉGOIRE Jean-Nicolas (000446638)
VANDERVAEREN Kevin (000546274)

Date:
May 28, 2025

Contents

1 Introduction

Single-cell RNA sequencing (scRNA-seq) has transformed our understanding of cellular diversity by enabling gene expression profiling at single-cell resolution. Two main types of algorithms are commonly used to analyze scRNA-seq data: clustering and trajectory inference. Clustering algorithms are typically used to identify distinct cellular populations based on gene expression profiles, while trajectory inference methods aim to reconstruct dynamic biological processes by ordering cells along developmental or differentiation continua. However, these two approaches often produce markedly different results when applied to the same dataset, leading to ambiguities in data interpretation. For instance, a dataset may appear as distinct clusters under a clustering algorithm, but exhibit a smooth progression under trajectory inference, making it unclear which analysis better reflects the biological reality.

To address this, Lim and Qiu (2024) [1] proposed a quantitative framework for assessing whether a dataset is more "cluster-like" or "trajectory-like." Their method introduces five scoring metrics based on pairwise distances, persistent homology, vector directionality, Ripley's K, and connectivity to objectively characterize the geometric structure of scRNA-seq data.

In this project, we reproduce their approach and apply it to both synthetic and real datasets. We will evaluate the ability of these metrics to correctly distinguish between cluster-like and trajectory-like data structures. Our implementation is available here.

2 Related Work

Earlier studies have explored the challenges of interpreting scRNA-seq data geometry. Saelens et al. (2019) [2] conducted a comprehensive comparison of 45 trajectory inference methods, highlighting the variability in performance and the importance of method selection based on dataset characteristics. Their work underscores the need for objective criteria to guide the choice between clustering and trajectory inference methods in scRNA-seq analysis.

Similarly, Zhang et al. (2020) [3] reviewed clustering methods for scRNA-seq data, outlining their strengths and limitations, and emphasizing the difficulty of robustly identifying distinct cell populations.

These studies demonstrate the complexity of scRNA-seq data interpretation and set the stage for more recent quantitative approaches like the one proposed by Lim and Qiu (2024) [1].

3 Materials and methods

3.1 Datasets

To evaluate and reproduce the methodology proposed by Lim and Qiu [1], we used both simulated datasets and the same real single-cell RNA-seq (scRNA-seq) datasets that they used.

In first place, we used a *simulated datasets* in which we recreated four types of synthetic data as described in the original paper: *Clear clusters, Clear trajectories, Noisy clusters & Noisy trajectories.* The data were generated using either Gaussian mixtures for cluster-like structures or parametric curves (e.g., sine functions) with added noise for trajectory-like patterns. We implemented these simulations using Python, libraries `numpy` and `scikit-learn`.

Then, for the *real scRNA-seq datasets* we reused a subset of publicly available datasets mentioned in the original GitHub repository provided by the authors, available here. These include the Planaria, Mouse Bone Marrow, and Natural Killer T-cell datasets. Each files contains a long list of genes representing various biological functions, such as cell metabolism, immune function, stress response, and extracellular matrix remodeling, all of which are crucial for understanding cell behavior, particularly in single-cell analyses. Each gene corresponds to an individual feature in the dataset. These datasets were preprocessed using standard scRNA-seq procedures: filtering low-quality cells, normalizing by library size, and selecting highly variable genes.

In addition, we tested our implementation on a separate dataset from a different study to assess the robustness of the pipeline. However, finding a version in the correct format (gene-by-cell matrix) compatible with our pipeline required significant preprocessing and metadata alignment. We used the human brain dataset originally published by Darmanis et al. (2015) [4], which was also cited in the review by Zhang et al. (2020) [3] on clustering methods for single-cell RNA-seq.

3.2 Methodology

The implementation was done in Python using Jupyter notebooks, with core dependencies including `ripser, scikit-learn, and numpy` and a modified version of *scanpy*.

3.2.1 Step 1: Data Preprocessing

All datasets (simulated and real) were reduced to 20 principal components using PCA to standardize dimensionality across samples. For both datasets we applied a trimming step based on a quantile threshold of both the total distance (td) and outlier distance (od) to remove extreme cells that could distort the geometry. Finally a simple normalization along each features was processed on both datasets.

3.2.2 Step 2: Distance Matrix

We computed the geodesic distances between cells using Diffusion Pseudotime (DPT) from the `scanpy` package. To handle non-finite entries in the DPT distance matrix, positive infinite values were capped at 1.5× the maximum observed finite distance, while negative infinite values were replaced with $-1\times$ the minimum observed finite distance to ensure consistency.

3.2.3 Step 3: Scoring Metrics

Each dataset was then evaluated using the following five metrics:

- *Entropy of pairwise distances*: Entropy was computed on a histogram of pairwise DPT distances.
- *Persistent homology entropy*: Using the `ripser` library, we computed 0-dimensional persistent homology and measured the entropy of merge distances.
- *Vector magnitude score*: After applying K-means (with $K = 5\%$ of sample size), we followed vector chains across cluster centers and computed the

average magnitude over 5 runs.

- *Ripley's K function*: For both real and uniformly random point clouds in the same convex hull, we computed and compared $k(t)$ values using geodesic distances.
- *Degrees of connectivity*: For increasing k-nearest-neighbor values (5% to 95% of dataset size), we built mutual neighbor graphs and computed reachability for each node. The final score is the area under the curve of median reachability across k-values.

3.2.4 Step 4: UMAP Projection

After scoring 6.000 simulated datasets, we applied UMAP to the resulting score matrix to define a $2D$ geometric landscape. Real scRNA-seq datasets were then scored and projected onto this same landscape to evaluate their structural similarity to known cluster-like or trajectory-like patterns with at most some rotational difference depending of the number of simulated datasets.

3.2.5 Pipeline and Dataset Overview

The pipeline described in Figure 1 was designed by the authors to ensure full reproducibility. All code used in this work is available upon request or can be adapted directly from the authors' repository with minor modifications for local datasets and parameter tuning.

4 Results

At the outset of our work, we hypothesized, based on early observations, that reproducing the pipeline under the same conditions should yield results consistent with those reported by Lim and Qiu [1]. Our primary objective was to verify the reproducibility of the information presented in their article and to assess the validity of the proposed framework. Additionally, we noted that certain figures included in the original publication were not that same as the ones generated in the Jupyter notebooks.

4.1 Quantifying Clusterness and Trajectoriness

We first generated 6.000 simulated datasets equally divided across the four structural categories, then each dataset was evaluated using the five proposed metrics. As shown in Figure 2, each metric clearly differentiated cluster-like from trajectory-like structures. In particular, as expected, trajectory-like datasets consistently showed higher values for entropy-based scores, while cluster-like datasets had stronger contrast in Ripley's K and lower reachability.

4.2 UMAP Embedding of Geometric Landscape

In Figure 3 we see that clear clusters and clear trajectories are positioned at opposite corners of the landscape, while noisy cases occupy transitional regions. The smooth gradients observed in the UMAP confirm that the metrics collectively describe a continuous spectrum between clusterness and trajectoriness. Note that due to the stochastic nature of UMAP and its invariance to rotation and reflection, our projected $2D$ geometric landscape appears as a mirrored version of that in the original paper. This does not impact the interpretability or consistency of the clusterness and trajectoriness patterns.

4.3 Projection of Real scRNA-seq Datasets

In Figure 4a, our projections are covering the full spectrum of the clusterness-trajectoriness space, with biological datasets like Planaria and NK cells projecting closer to the cluster-like region, and others like fibroblast or bone marrow differentiation aligning with trajectory-like regions. The violin plots in Figure 5 are also showing that the real data distributions align well with those from simulated datasets, confirming that the scoring system performs reliably on real scRNA-seq data.

We see in 4b that the red triangle, representing the projection of the scRNA-seq of the human brain is in a cluster zone. Confirming that the scoring system also performs reliably on another real scRNA-seq dataset.

5 Discussion

During the reproduction of Lim and Qiu's framework, we encountered several technical and methodological challenges that highlight limitations in the original implementation. These issues impacted both our understanding of the method and the reproducibility of results, prompting us to make specific modifications.

One of the main difficulties was the overall lack of clarity in the original code. The code suffered from very limited documentation and contained large sections of duplicated logic, making it difficult to trace. Additionally, several "magic numbers" appeared throughout the code (e.g., hardcoded thresholds or constants) without justification or explanation, limiting transparency and interpretability.

Another important issue was the lack of clear technical specifications to ensure a consistent execution environment. Although the authors provided a list of Python package requirements, they omitted the version of Python used. After investigation, we determined that their code was developed under Python 3.8, an outdated version that presented compatibility issues with current libraries. We therefore adapted the implementation to work with Python 3.12, which required several adjustments.

We also identified inconsistencies in reproducibility related to random number generation. The original pipeline did not include fixed seeds across all stochastic processes, such as bootstrap sampling and K-means clustering.

We made a key adjustment to their data resampling strategy. In the original pipeline, bootstrap sampling was applied on data that had already undergone an earlier bootstrap stage. We opted to simplify this by applying bootstrap sampling only once, directly on the original datasets but over multiple iterations. This change reduced redundancy and brought the procedure closer to standard statistical practice, without compromising the variability required to evaluate the metrics.

Finally, a significant limitation of the original pipeline was its computational inefficiency. To address this, we used `numpy` methods instead of standard functions, which greatly reduced the execution time. Furthermore, we reduced the cost of the metric vector by eliminating all overhead calculations. Thanks to this improvement, we were able to accelerate the computation by a factor of 20 compared to the original sequential implementation.

In summary, our adapted pipeline offers improved clarity, reproducibility, and computational performance while preserving the original objectives.

References

[1] Hong Seo Lim and Peng Qiu. "Quantifying the clusterness and trajectoriness of single-cell RNA-seq data". In: *PLoS Computational Biology* 20.2 (2024), e1011866.

[2] Wouter Saelens et al. "A comparison of single-cell trajectory inference methods". In: *Nature biotechnology* 37.5 (2019), pp. 547–554.

[3] Shixiong Zhang et al. "Review of single-cell RNA-seq data clustering for cell-type identification and characterization". In: *Rna* 29.5 (2023), pp. 517–530.

[4] Spyros Darmanis et al. "A survey of human brain transcriptome diversity at the single cell level". In: *Proceedings of the National Academy of Sciences* 112.23 (2015), pp. 7285–7290.

A Figures

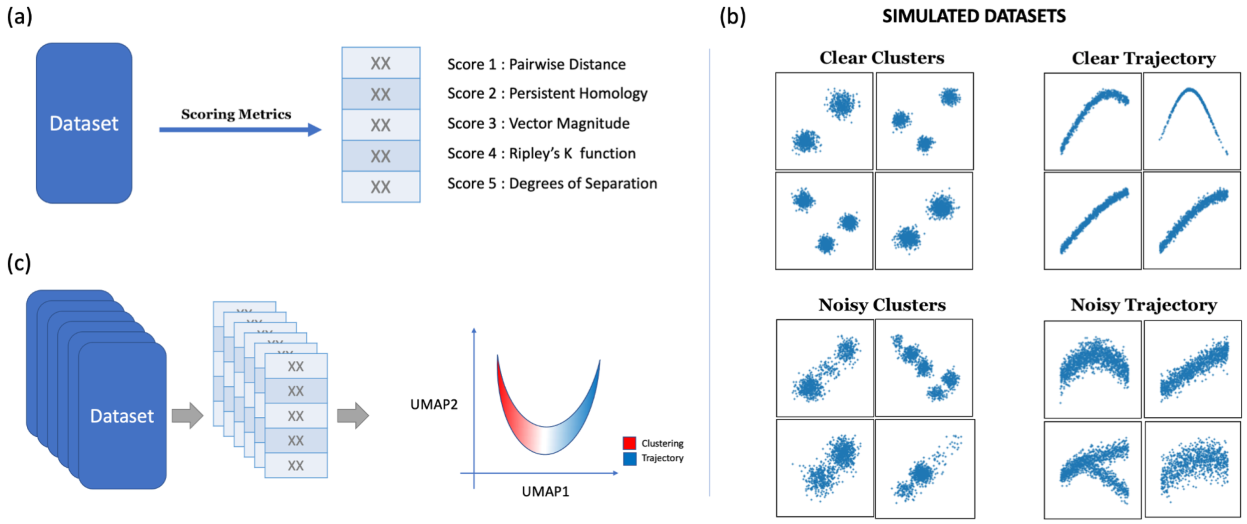


Figure 1: **Overview of the proposed pipeline and simulated data used**. (a) Given a dataset, five different scoring metrics are used to quantify the dataset. The output is five numerical scores. (b) Scatter plots visualizing a few examples of simulated datasets. The simulated datasets are two-dimensional. (c) A multitude of simulated datasets was scored by the scoring metrics, and the scores are projected to UMAP space. `https://doi.org/10.1371/journal.pcbi.1011866.g003`

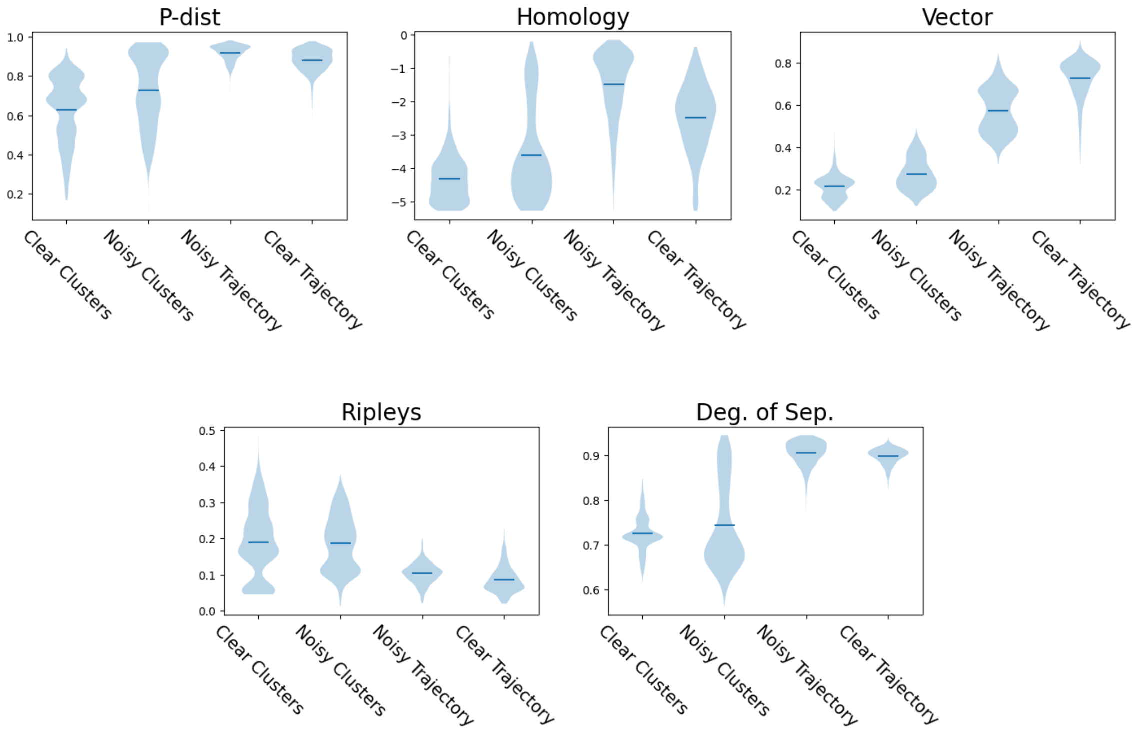
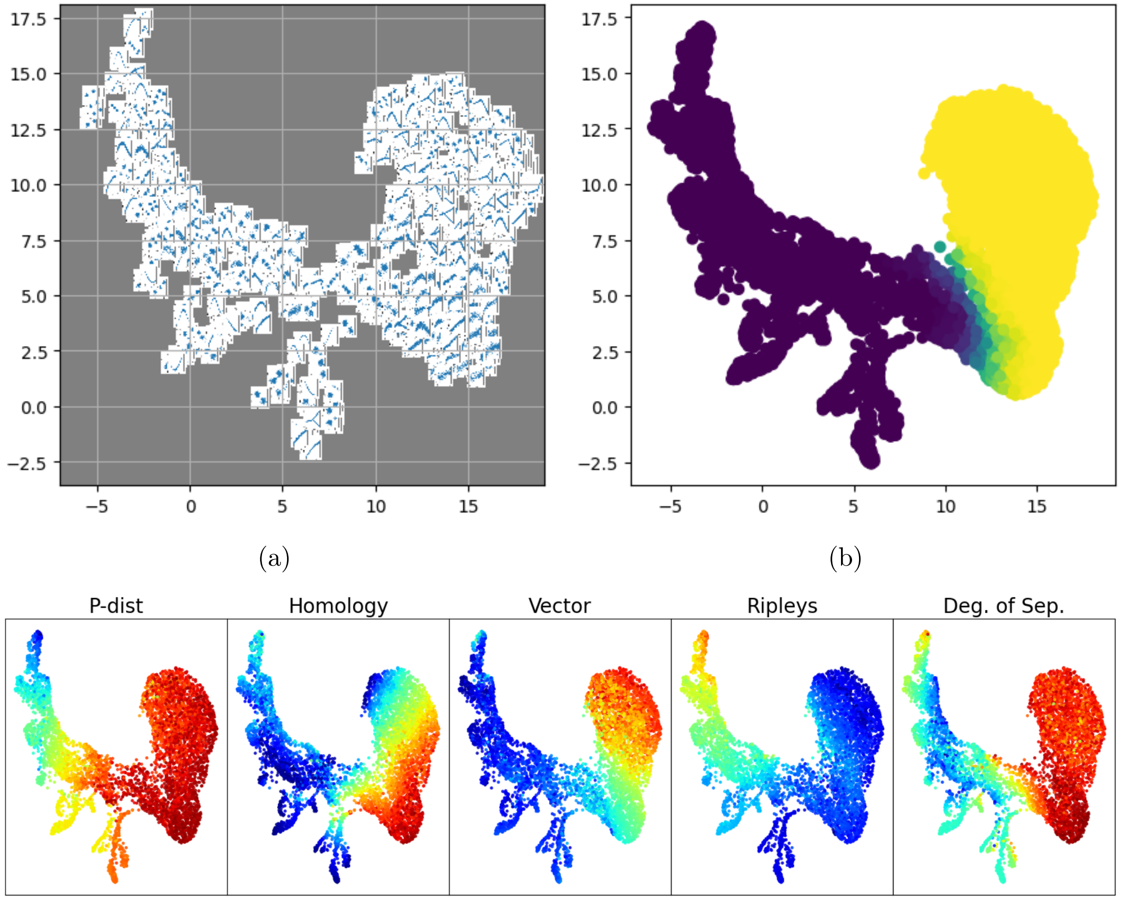


Figure 2: **Differences between datasets**: Violin plots showing the scores across the datasets, the blue lines representing the median of the distributions ($n = 1500$).

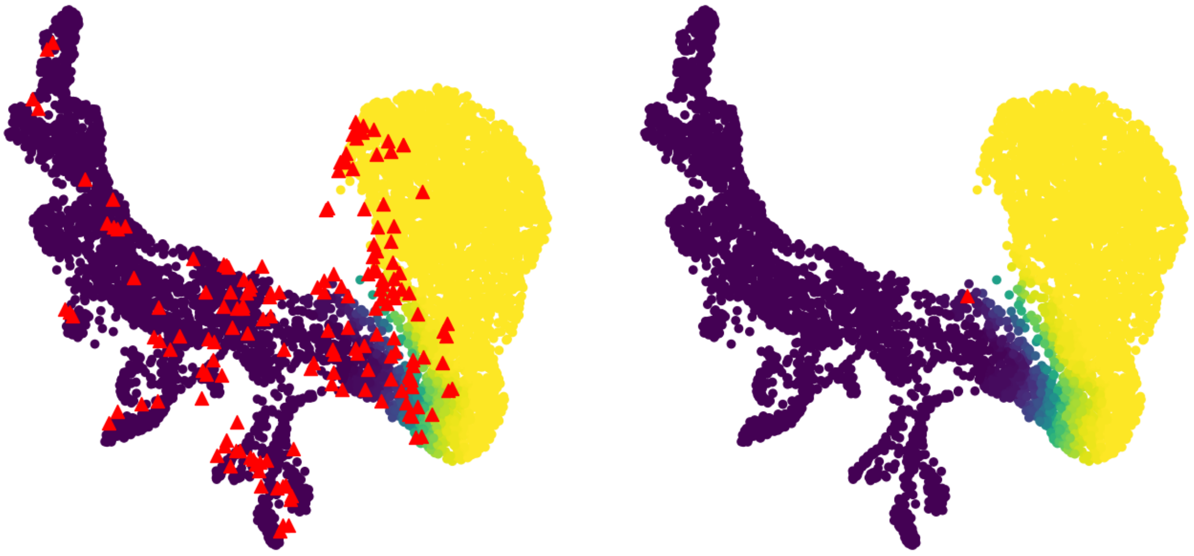


(a)

(b)

(c)

Figure 3: **Geometric landscape of clusterness and trajectoriness**: In figure 3a each dot in the UMAP represents one simulated dataset visualized by the scatter plot of the simulated dataset itself. In figure 3b dots in the UMAP plot were colored by the proportion of neighboring dots belonging to clear or noisy trajectory-like data. The boundary on the colored UMAP shows the separation between each type of datasets. Figure 3c is a colored visualizations by values of each of the five scoring metrics, showing variations of the scores in the UMAP space. Red represents higher values, and blue represents lower values.



(a)

(b)

Figure 4: **Projections of the real scRNA-seq datasets used by Lim and Qiu and the human brain onto the simulated geometric landscape.** In Figure 4a each red triangle represents the projection of one of the 169 real scRNA-seq datasets. In Figure 4b the red triangle is in a cluster zone (see Figure 3a) as expected.

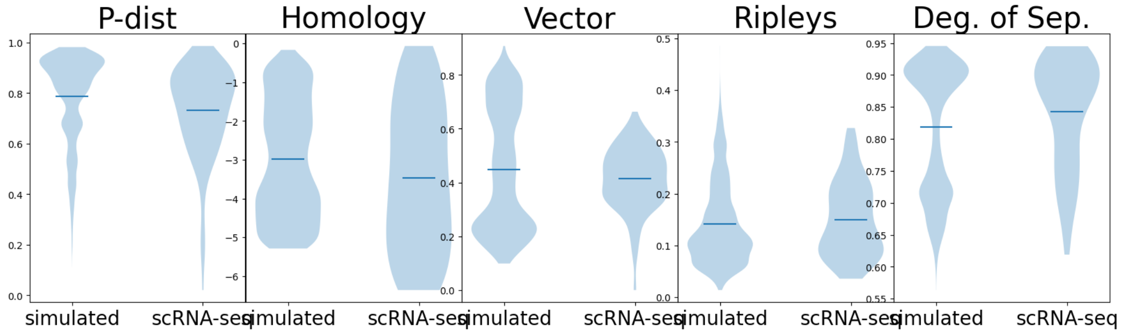


Figure 5: **Violin plots of the scoring metrics in the simulated versus the real scRNA-seq data.** The distribution of scores are similar between simulated and real datasets.