Reeb Graphs and Towers: Multiscale Skeletons for Data

Andrew J. Steindl
Department of Computer Science
Yale University

Dhananjay Bhaskar
Department of Biomedical Engineering
University of Wisconsin–Madison

Smita Krishnaswamy
Departments of Computer Science and Genetics
Yale University

Abstract

Recently there has been a profusion of tools to analyze high dimensional point cloud data, popularized by the advent and popularity of single cell data in biology. As the number of datasets of this type grow, the need to automate analysis of the data and generate hypotheses grows. However, most of the techniques developed for single-cell data assume a particular type of structure in the data, i.e., typically cluster structure, tree structure, or trajectory structure, and provide specific heuristics to analyze such structure. For example a standard single cell pipeline consists of UMAP visualization followed by Louvain community detection, which assumes the need to visualize neighborhoods in the cellular data, and to separate them into clusters. But deciding which tool to use requires human intervention. Instead, here we provide a method for automating data shape detection from noisy single cell point clouds using the concept of a Reeb graph, called a scReebTower. A Reeb graph is a topological descriptor that uses a TDA-like filtration of the data to detect connected components of a level set of the data. Then each connected component is contracted to a single point resulting in a graph where nodes correspond to critical points and edges relationships between points. Here we propose to construct such graphs by first starting with a data diffusion operator and performing a Vietoris–Rips–like filtration over diffusion scales to reveal multiscale structure. scReebTower first creates a skeleton-Reeb graph of the data at each scale that reveals the fundamental structure of the data beneath the vast amount of noise in single cell data. Next, scReebTower computes the persistence homology of each floor as a quantitative signature of the shape at each scale. This set of signatures then allows classification methods to distinguish between datasets that have cluster structure, tree structure, trajectory structure and undifferentiated (blob–like) structure at a particular scale which naturally suggests downstream analysis tools. Preliminary results show single Reeb graphs and scReebTowers atop single cell embryoid body[1] data and constructed toy data sets.

1 Introduction

The rapid growth of single cell and other high dimensional biological datasets has led to a large increase of analysis tools. Yet most methods *a priori* assume the data is of particular structure, i.e. clusters, trees, or trajectories, and require sensitive hyperparameter choices or expert interpretation.

Biological data often contain geometry that cannot be reduced to simple clusters: branching trajectories (e.g., treatment resistance) and cycles (e.g., circadian or cell-cycle programs) are frequent.

1st Conference on Topology, Algebra, and Geometry in Data Science (TAG-DS 2025).

While dimensionality reduction methods such as PHATE [2] and UMAP [3] preserve this geometry, their outputs remain point clouds. As a result, how regions connect, branch, or form cycles remains implicit and are inferred by visual inspection. What is needed is an explicit *skeleton*: a parsimonious graph that makes this structure tangible and supports automated, unbiased downstream analyses.

We introduce **scReebTowers**, a method for automated data-shape detection that constructs diffusion-refined, Reeb-style skeletons of noisy, high-dimensional point clouds. The approach builds a diffusion operator to capture intrinsic geometry while suppressing noise, then defines a diffusion-geodesic filter to order points along intrinsic directions of variation. Across scales, scReebTowers performs a Vietoris-Rips (VR) filtration implemented through diffusion condensation, which progressively coarsens neighborhoods while preserving manifold structure. At each scale, a Reeb graph summarizes the essential skeleton—its branches, cycles, and connectivity—and graphs are linked across scales to form a multiscale *tower* that tracks how features merge, split, or persist through successive levels of diffusion. The design rests on two principles: a diffusion geometry for robust Reeb construction in noisy, high-dimensional settings, and a multiscale, VR-style filtration realized through diffusion condensation for stable, interpretable structure.

Practically, *scReebTowers* constructs a diffusion operator on the kNN graph to suppress off-manifold shortcuts; derives diffusion-geodesic filters; defines bins via kNN-guided sliding windows; and uses connected components within bins as nodes, with edges drawn by overlaps across consecutive bins. The result is a cleaner, reproducible skeleton in which branches appear as pendant edges and cycles as loops. To capture structure faithfully across scales, we use diffusion condensation to build a *Reeb Tower*: each diffusion scale yields a Reeb graph at its own resolution, and we compute the persistent homology of these floors to quantify how structural features persist or disappear with increasing smoothing.

In the remainder of this paper, we detail the construction and illustrate how branches and cycles can be stably recovered across scales on various data.

1.1 Diffusion Operator Background

The diffusion operator is the row-stochastic matrix

$$P = Q^{-1}W,$$

where W is the affinity matrix and $Q = \mathrm{diag}(W\mathbf{1})$ the degree matrix. P defines a Markov chain on the data graph, with P^t giving t-step transition probabilities. It encodes manifold connectivity and underlies methods such as diffusion maps[4] and diffusion condensation[5][6].

2 Methods

2.1 Reeb Graph Construction

To construct an approximated Reeb graph, our pipeline proceeds through the following stages: construction of a k-nearest neighbor graph, definition of a diffusion-based filter, adaptive sliding-window binning, local clustering, and graph assembly. Collectively, this process can be viewed as a Vietoris-Rips (VR) style filtration on the diffusion-refined kNN graph, in which connected components of overlapping neighborhoods are contracted to nodes, forming an approximate Reeb graph at each scale.

kNN Affinity Graph Given an input embedding, we construct a k-nearest neighbor (kNN) graph. The parameter k is the primary hyperparameter of our pipeline and can be set adaptively to preserve a desired number of connected components. We select the largest k up to a specified maximum, or we select the minimum k that maintains this connectivity. From the kNN graph we compute affinities between nodes using a Gaussian kernel with local scaling[7]:

$$W_{ij} = \exp\left(-\frac{\|x_i - x_j\|^2}{\sigma_i\sigma_j + \varepsilon}\right),$$

where $x_i, x_j \in \mathbb{R}^d$ are data points, $\|x_i - x_j\|$ is their Euclidean distance, σ_i is the distance from x_i to its k-th nearest neighbor (normalizing for density variation), and ε is a small constant for numerical

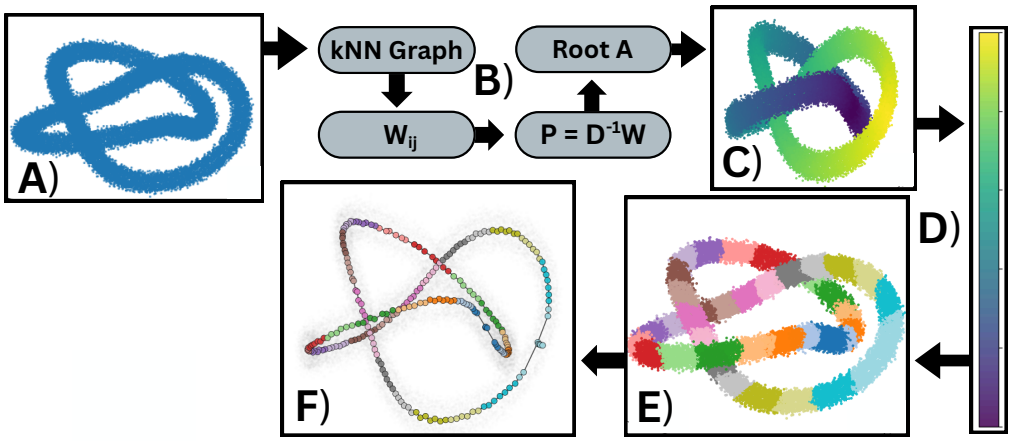

Figure 1: Schematic of our pipeline. **(A)** Input point cloud. **(B)** Construction of the kNN graph, computation of affinities, degree matrix, diffusion operator, and selection of Root A. **(C)** Computation of the diffusion geodesic filter. **(D)** Ordering of points by the filter. **(E)** Adaptive kNN-based binning of the point cloud in filter space. **(F)** Final Reeb graph approximation.

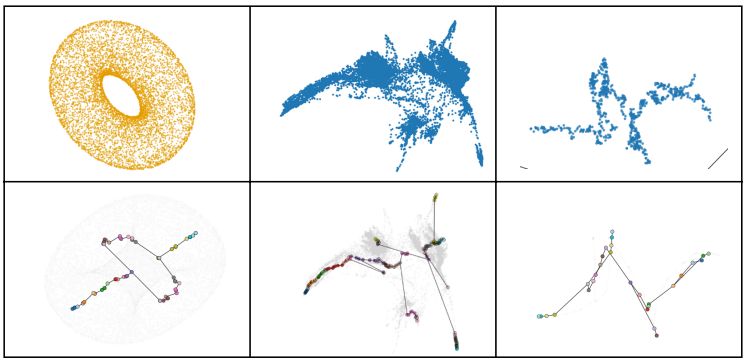

Figure 2: Reeb Graph Approximations on a torus, PHATE[2] embedding of EBD single cell data[1], and a DLA tree.

stability. The resulting affinity matrix W is symmetric and sparse, containing nonzero entries only for kNN pairs.

We then row-normalize this affinity matrix to obtain the diffusion operator[4], P which underlies all subsequent steps. The k-nearest neighbor graph, affinity matrix, and diffusion operator are computed exactly as in PHATE and diffusion condensation [2, 6].

Diffusion-based Filter To capture the intrinsic geometry of the data while suppressing noisy shortcuts, we use a diffusion-informed geodesic metric. Starting from the diffusion operator P, we compute its symmetrized form $P_{\text{sym}} = \frac{1}{2}(P + P^\top)$ and define an edge-cost matrix

$$\text{cost}_{ij} = -\log P_{\text{sym},ij},$$

so that sparse connections incur higher traversal cost. We then compute pairwise distances between points as shortest path distances on this weighted cost graph, yielding a diffusion geodesic metric that follows the manifold structure while reducing sensitivity to sampling noise [4, 8]. Within each connected component, we identify approximate diameter endpoints by two sweeps of Dijkstra's algorithm (seed $\to A \to B$). The distances from the first endpoint A define the scalar filter function f, which orders points for subsequent binning.

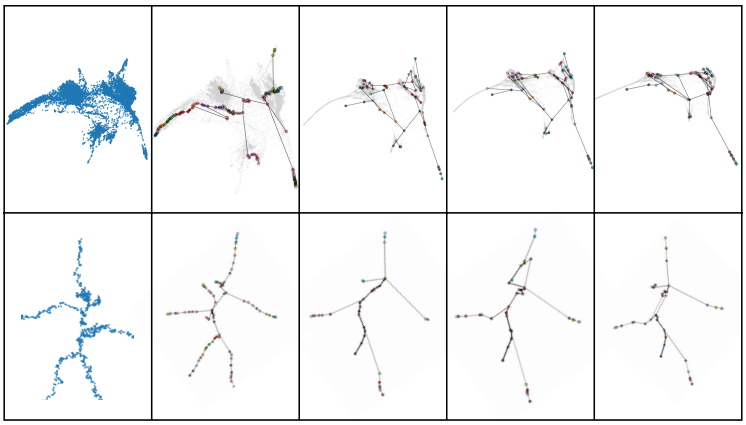

Figure 3: Floors of two Reeb towers. The top row is single cell data, the bottom row is a DLA tree.

Binning via Sliding Windows We construct overlapping sliding windows along f with widths and steps derived from the *kNN graph* using a hyperparameter k_{bin} which we set to 95% of the k from the kNN graph. For each connected component of the kNN graph we order the points by f and let $\text{rank}(i)$ denote the rank of point i. Then for every point i, we compute $d_k(i)$ which is the rank difference between i and its k_{bin}-th nearest neighbor. We then set

$$Q = \text{percentile}_{95}\{d_k(i)\}, \quad W_{\text{init}} = \left\lceil \alpha \cdot \frac{2m}{m-1} \cdot Q \right\rceil,$$

where α is padding for the windows and m is the desired overlap (we use $m = 5$). Windows step with stride $\text{step} = \lfloor W_{\text{init}}/m \rfloor$. This way the windows are sized to minimize cutting across local neighborhoods, ensuring that most k_{bin} nearest neighbors remain within the same bin.

To reduce density bias, we measure the span of each window in the filter coordinate f and cap all widths at the median. This prevents windows in sparse regions from stretching disproportionately.

Clustering and Graph Assembly Within each bin, we induce the subgraph from the kNN graph and extract its connected components. These per-bin components become the nodes of the Reeb graph approximation. Nodes are connected if they share any data points across consecutive bins.

2.2 Multiscale Reeb Towers

Because a single Reeb graph approximation is sensitive to the resolution at which it is built, we extend the construction across multiple diffusion scales to capture structure robustly at different levels of granularity.

Diffusion Condensation Backbone We build multiscale structure out of *diffusion condensation*, an iterative process that repeatedly applies the diffusion operator to contract nearby points while preserving global manifold geometry [6, 5]. Early iterations retain fine detail, whereas later iterations yield progressively coarser representations, providing a natural backbone for constructing Reeb graphs across scales.

Recursive Reeb Graphs At each diffusion condensation step, nearby points are merged into clusters. To construct the next floor of the tower, each cluster is assigned a filter value equal to the average of its member points. These clusters are then treated as new "superpoints" on which we reapply the same Reeb graph procedure as at the base layer: computing a global kNN graph, binning according to the filter, and extracting connected components within each bin to define the nodes. This process yields a Reeb graph approximation at the new diffusion scale. Repeating this procedure across successive condensation iterations produces a hierarchy of Reeb graphs, one floor per scale, capturing progressively coarser structure in the data.

Persistence Analysis After constructing Reeb graphs across diffusion scales, we compute the persistent homology of each floor's graph to quantify feature stability. This provides a quantitative summary of how structural features persist across the filtration scales.

References

[1] Kevin R. Moon, David van Dijk, Zheng Wang, Scott Gigante, Daniel B. Burkhardt, William S. Chen, Kristina Yim, Antonia van den Elzen, Matthew J. Hirn, Ronald R. Coifman, Natalia B. Ivanova, Guy Wolf, and Smita Krishnaswamy. Visualizing structure and transitions for biological data exploration. *bioRxiv*, 2019.

[2] Kevin R. Moon, David van Dijk, Zheng Wang, Scott Gigante, Daniel B. Burkhardt, William S. Chen, Kristina Yim, Antonia van den Elzen, Matthew J. Hirn, Ronald R. Coifman, Natalia B. Ivanova, Guy Wolf, and Smita Krishnaswamy. Visualizing structure and transitions in high-dimensional biological data. *Nature Biotechnology*, 37(12):1482–1492, 2019.

[3] Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018.

[4] Ronald Coifman and Stéphane Lafon. Diffusion maps. *Applied and Computational Harmonic Analysis*, 21:5–30, 07 2006.

[5] Guillaume Huguet, Alexander Tong, Bastian Rieck, Jessie Huang, Manik Kuchroo, Matthew Hirn, Guy Wolf, and Smita Krishnaswamy. Time-inhomogeneous diffusion geometry and topology, 2023.

[6] Nathan Brugnone, Alex Gonopolskiy, Mark W. Moyle, Manik Kuchroo, David van Dijk, Kevin R. Moon, Daniel Colon-Ramos, Guy Wolf, Matthew J. Hirn, and Smita Krishnaswamy. Coarse graining of data via inhomogeneous diffusion condensation. In *2019 IEEE International Conference on Big Data (Big Data)*, page 2624–2633. IEEE, December 2019.

[7] Lihi Zelnik-manor and Pietro Perona. Self-tuning spectral clustering. In L. Saul, Y. Weiss, and L. Bottou, editors, *Advances in Neural Information Processing Systems*, volume 17. MIT Press, 2004.

[8] Laleh Haghverdi, Maren Büttner, F. Alexander Wolf, Florian Buettner, and Fabian J. Theis. Diffusion pseudotime robustly reconstructs lineage branching. *Nature Methods*, 13(10):845–848, Oct 2016.

[9] Gurjeet Singh, Facundo Memoli, and Gunnar Carlsson. Topological Methods for the Analysis of High Dimensional Data Sets and 3D Object Recognition. In M. Botsch, R. Pajarola, B. Chen, and M. Zwicker, editors, *Eurographics Symposium on Point-Based Graphics*. The Eurographics Association, 2007.

A Technical Appendices and Supplementary Material

B Related Work

Existing methods such as clustering, trajectory inference, and Mapper [9] describe local or hierarchical relationships but are not grounded in the geometry of the data manifold itself. Classical Reeb graph approximations can capture global structure but are unstable on noisy, high-dimensional point clouds. In contrast, *scReebTowers* builds a diffusion-refined graph and performs a multiscale filtration to robustly extract the manifold's one-dimensional scaffold, providing a principled foundation for downstream analyses such as clustering or trajectory detection.

C limitations

The current implementation is sensitive to non-uniform sampling and local density variation, which can distort the filter function and lead to uneven binning. This can be mitigated through optimal-transport quantile normalization of the filter values, though it remains an open challenge for highly

irregular data. The method may also require preprocessing to remove sparse, off-manifold points that can mislead the diffusion-geodesic filter. Additionally, tuning the overlap of sliding windows affects graph connectivity, too little overlap can fragment components. To mitigate this we can over-bin and simplify. Finally, computing large kNN graphs can be computationally expensive for high-dimensional or large-scale datasets.

D Future Work

Vertical Edges We connect nodes across consecutive floors whenever they share underlying points, assigning each vertical edge a weight

$$w(v_i, u_j) = \frac{|v_i \cap u_j|}{|v_i|}$$

which measures the fraction of the lower node's points that merge into the coarser node. This probabilistically summarizes how local structures coalesce as you ascend the tower. Identifying meaningful interpretations or quantitative uses for these vertical edges remains an open question and a direction for future work.

Algorithmic Improvements Future work will focus on refining the algorithmic pipeline, including improving stability across hyperparameters, and optimizing computational efficiency for large datasets. Additional work will address current limitations aiming to make scReebTowers a more general and reliable framework for multiscale topological analysis. Most future improvements will focus on the *scReebTower* framework rather than the underlying Reeb graph approximation itself, though enhancements to both are possible. In particular, we aim to better integrate diffusion condensation with the Reeb graph pipeline.