bioRχiv

CellFateExplorer: An integrated platform for exploring cell fate

Zhaoyang Huang[1], Haonan Ma[1], Yuchuan Peng[1], Chenguang Zhao[2,*], Liang Yu[1,*]

[1] School of Computer Science and Technology, Xidian University, Xi'an 710071, Shaanxi, China

[2] Department of Rehabilitation Medicine, Xijing Hospital, Fourth Military Medical University, Xi'an, China

Abstract

Trajectory inference algorithms based on single-cell omics data are powerful tools for predicting and dissecting cell differentiation. However, most existing tools are tailored to specific datasets and lack generalizability across diverse data types. To address this limitation, we developed CellFateExplorer, systematically evaluates the performance of *x* trajectory inference methods across *y* datasets. Through an interactive web interface, CellFateExplorer provides guidance on method selection and downstream analysis for specific datasets. In summary, CellFateExplorer is an integrated platform for exploring cell fate.

Keyword: Trajectory inference, Benchmark, Web-based interaction

Introduction

Single-cell sequencing technologies, particularly single-cell RNA sequencing (scRNA-seq), have revolutionized biological research by enabling analysis at single-cell resolution. These technologies provide a robust foundation for studying dynamic processes such as cell differentiation, organ development, and disease progression[1]. Beyond basic cell clustering and annotation, trajectory inference has emerged as a critical computational tool for reconstructing the temporal order of cells within a sample. Visualization techniques have also evolved, ranging from one-dimensional pseudotime (e.g., Palantir[2]) to non-Euclidean tree and graph structures (e.g., Monocle[3]), and vector field representations of RNA velocity (e.g., scVelo[4]).

In recent years, numerous computational tools have been developed to tackle trajectory inference from multiple perspectives. For instance, scVelo leverages spliced/unspliced mRNA ratios to infer cellular dynamics[4], Dynamo maps cells to a single-cell vector field[5], and CytoTrace2 employs interpretable deep learning to predict cell differentiation states and identify key driver gene[6]. As of January 2025, over a hundred trajectory inference methods have been published, each excelling in specific biological contexts or trajectory structures. Consequently, selecting the appropriate method for a given dataset is crucial.

Dynverse, a unified platform, benchmarked 45 trajectory inference methods across 339 dataset[7]. Although it provided a comprehensive evaluation based on accuracy, scalability, and usability, the platform has not been updated in recent years, leaving newer methods, such as those based on RNA velocity, unsupported. Inspired by Dynverse, we developed CellFateExplorer, which retains some design elements from Dynverse while introducing several novel features:

1. Data Integration: Seamlessly integrates single-cell "AnnData" data structures with Dynverse milestone trajectory information, enabling compatibility with the scverse ecosystem of single-cell analysis tools.
2. Benchmarking: Systematically evaluates *x* trajectory inference methods across *y* datasets, covering eight categories of trajectory inference, including pseudotime and RNA velocity, to generate method selection guidelines.
3. Downstream Analysis: Introduces diverse downstream analyses to enrich trajectory results and validate biological significance.
4. Web Interface: Develops web interface based on cellxgene[8], allowing users to interactively select methods and visualize trajectory inference results.

Result

Framework for CellFateExplorer

The model workflow of CellFateExplorer is illustrated in **Fig. 1**. Each module is described as follows:

For data module, the new developed data structure "FateAnnData", built upon "AnnData", enables seamless integration with Scanpy. It incorporates Milestone and Waypoint substructures as reference trajectory labels and includes both real and simulated datasets.

For preprocessing module: In addition to standard preprocessing steps (e.g., matrix normalization, log transformation, and feature selection), prior knowledge such as starting cell states and cell clustering is incorporated to guide trajectory inference.

For method module: Users can execute methods via Python or R scripts or through Docker containers, which simplify deployment by encapsulating dependencies. The framework supports eight output prototypes, with seven optimized from Dynverse and a novel unified wrapper for RNA velocity. Results are evaluated based on accuracy, usability, runtime, and dataset compatibility, providing users with method selection recommendations.

Visualization and Downstream Analysis: Trajectory results are visualized through various plots, including milestone network structures and cell trajectory embeddings. Downstream analyses include driver gene identification (e.g., BEAM[3], Gene2Gene[9])

and gene regulatory network inference (e.g., SCENIC[10], CellOracle[11]). An interactive web interface further enhances user experience by enabling dynamic exploration of results.

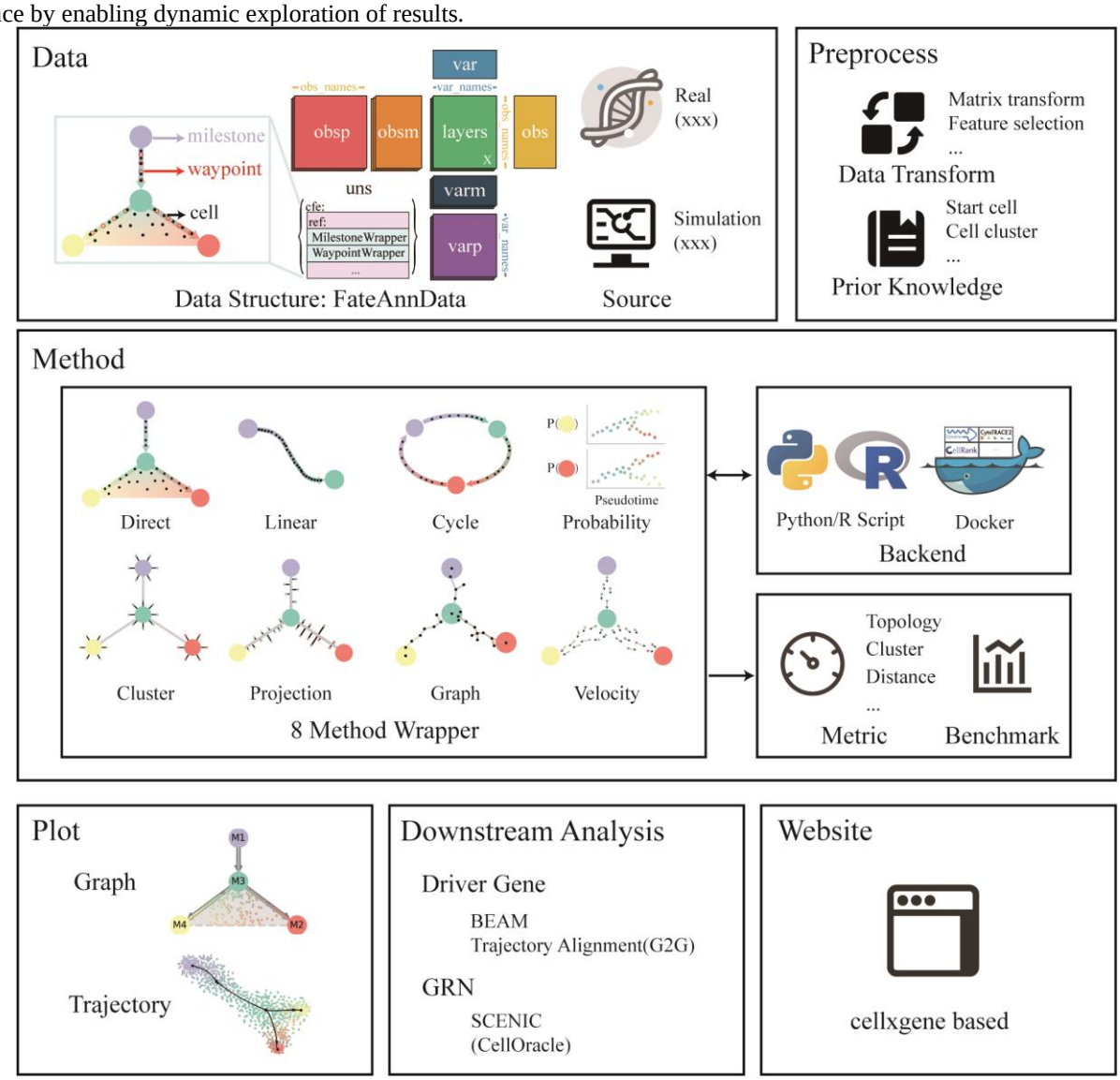


Figure 1: Framework of CellFateExplorer. (A) Data structure and sources. **(B)** Preprocess module, including data transformation and prior knowledge integration. **(C)** Method module, featuring eight prototype wrappers, backend execution via scripts or Docker, and benchmarking metrics. **(D)** Visualization module for cells and milestones. **(E)** Downstream analysis module, covering driver gene and GRN (gene regulatory network) inference. **(F)** Website module for interactive analysis.

Benchmark

Benchmarking results for *y* methods across *x* datasets are summarized in Figure 2.

Figure 2: Benchmark for y method on x dataset.

Guideline

Based on systematic evaluation, we provide method selection guidelines for specific datasets, as illustrated in Figure 3.

Figure 3: Guidelines for method selection.

Web UI for interactive trajectory analysis

The web interface, built on cellxgene, facilitates interactive exploration of single-cell data. Users can visualize cell labels, gene expression distributions, and perform differential expression analysis. CellFateExplorer enhances this interface by overlaying differentiation trajectories on dimensionality reduction plots and providing method selection guidance and metrics performance.

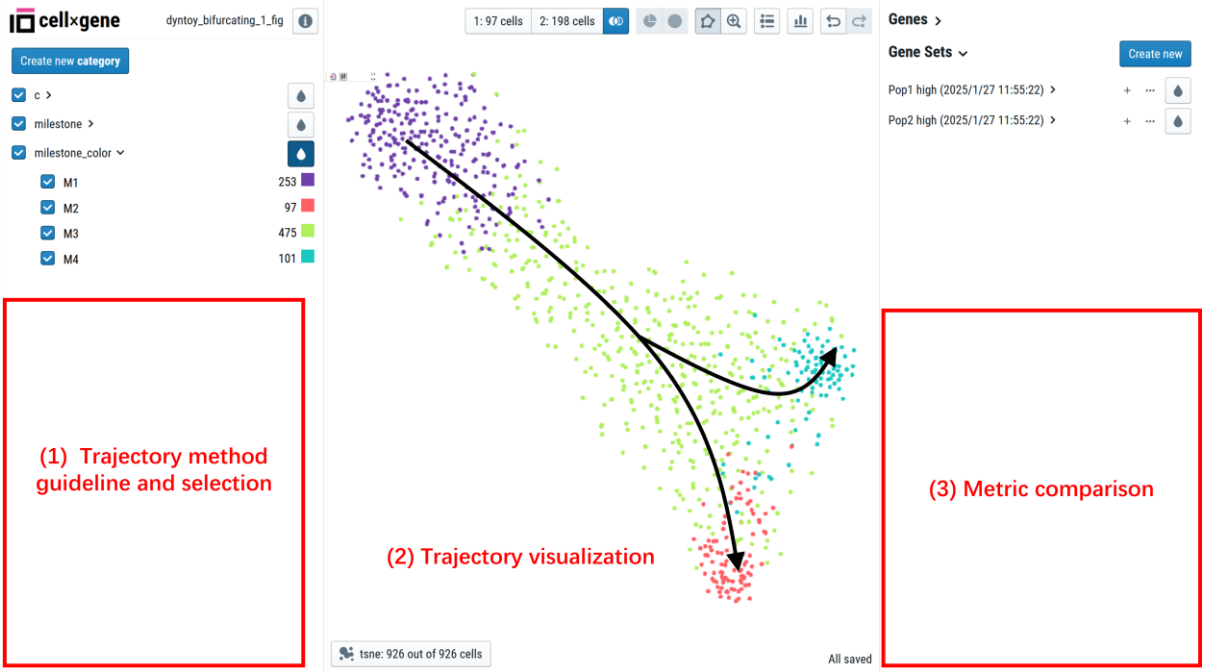


Figure 4: Web UI for CellFateExplorer

Case Study and Downstream Analysis

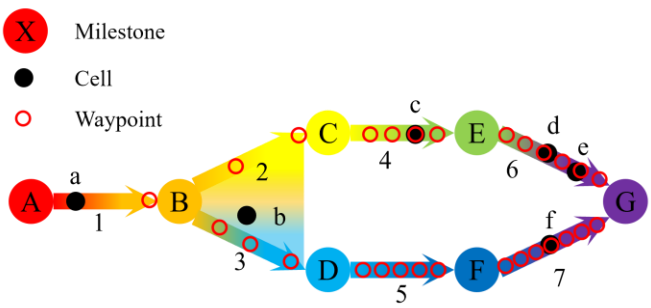
For simulation data ……

For real data ……

Methods

Milestone network structure

The milestone network and cell percentage provide an abstract representation of cell trajectories as shown in **Fig. 5**. Milestones represent key nodes in the trajectory, forming a graph structure, while percentages indicate the strength of cell assignments to trajectory edges. Waypoints are sampled at equal intervals along milestone edges, facilitating metric computation and mapping between topological structures and cell embeddings.



milestone_network

	from	to	length	directed
1	A	B	1	TRUE
2	B	C	2	TRUE
3	B	D	3	TRUE
4	C	E	4	TRUE
5	D	F	5	TRUE
6	E	G	6	TRUE
7	F	G	7	TRUE

divergence_regions

	divergence_id	milestone_id	is_start
1	BCD	B	TRUE
2	BCD	C	FALSE
3	BCD	D	FALSE

percentages

	cell_id	milestone_id	percentage
1	a	A	0.8
2	a	B	0.2
3	b	B	0.3
4	b	C	0.2
5	b	D	0.5
6	c	C	0.0
7	c	E	1.0
8	d	E	0.5
9	d	G	0.5
10	e	E	0.9
11	e	G	0.1
12	f	F	0.6
13	f	G	0.4

Figure 5: Representation Milestones, cell and waypoint concept graph. The milestone network is composed of a weighted graph structure, with its data stored in the "milestone_network" table. Information regarding local branching regions is stored in the "divergence_regions" table, while the specific positions of cells along the edges are recorded in the "percentage" table.

The remaining seven wrappers in **Fig. 1** are transformed into the "Direct" show in **Fig. 5** through a specific methods.

Trajectory methods showlist

Table 1: trajectory methods table

Name	Wrapper type
PAGA[12]	directed
Slingshot[13]	directed
Palantir[2]	linear
CytoTrace[6]	linear
Angle	cycle

CellRank[14]	probability
SCUBA[15]	cluster
MST	projection
Monocle[3]	graph
scVelo[4]	velocity
Dynamo[5]	velocity

Metric description

Benchmarking metrics are categorized into topology, clustering, and distance metrics.

Topology metric

Topological metrics are exclusively employed to assess the structure of the reference and predicted milestone networks, without considering the specific positions of cells.

The isomorphism metric is used to determine whether the reference and predicted network structures are isomorphic. This can be directly implemented using the "*networkx.is_isomorphic*" function from the Python package NetworkX.

The edge flip metric quantifies the transformation required to convert the reference network into the predicted network by adding or deleting edges. The metric is calculated as the number of edge modifications divided by the total number of edges in both networks. This problem is equivalent to the maximum common edge subgraph problem, which is known to be NP-hard.

Cluster metric

Clustering metrics are applied to assess the consistency between the reference clustering and the clustering results derived from the predicted milestone network. Milestones and edges are key entities in the milestone network that define the trajectory, and cells can be intuitively mapped to their corresponding milestones or edges to form new clusters, as illustrated in **Fig. 6**.

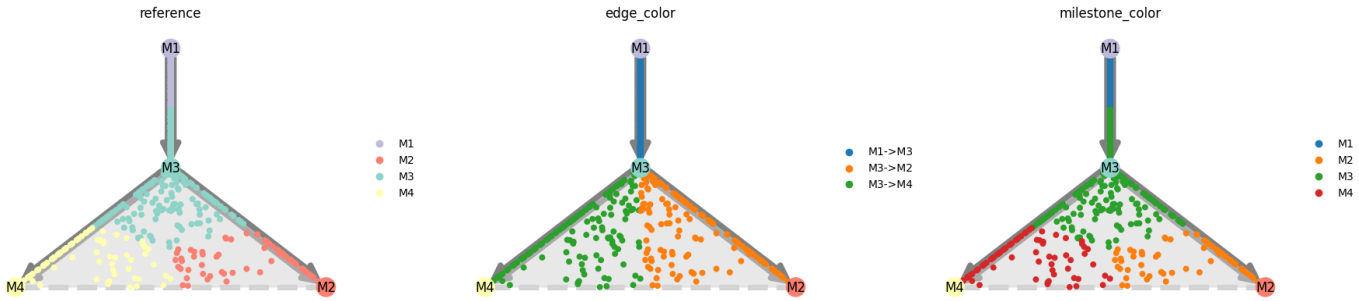


Figure 6: Graph plot with different cell cluster. (A) Reference cluster. **(B)** Cells are mapped towards the nearest milestone. **(C)** Cells are mapped towards the highest milestone percentage edge.

This is a classic supervised clustering evaluation task. While scikit-learn provides a variety of clustering metrics[16]. Here, the new clusters C are evaluated against the known reference clusters C' by a score, the harmonic mean of recovery and relevance, which represent the maximum average Jaccard similarity between the new clusters and the reference clusters, as defined by the following formula:

$$Jaccard(c,c') = \frac{|c \cap c'|}{|c \cup c'|}$$

$$Recovery = \frac{1}{|C|}\sum_{c \in C} \max_{c' \in C'} Jaccard(c,c')$$

$$Relevance = \frac{1}{|C'|}\sum_{c' \in C'} \max_{c \in C} Jaccard(c,c')$$

$$F1 = \frac{2}{\frac{1}{Recovery} + \frac{1}{Relevance}}$$

A higher score indicates greater similarity between the new clusters and the reference. For the two types of new clusters (based on milestones and edges), the corresponding metrics are denoted as $F1_{milestone}$ and $F1_{edge}$, respectively.

Distance

Distance metrics measure the consistency of paired-cell distance between reference and predicted trajectories.

Dissussion

CellFateExplorer provides a comprehensive ecosystem for trajectory inference, offering developers an easy-to-compare and integrate interface and users a rich selection of methods and guidance. Future developments may include:

Expansion for data and method: Extending to multi-omics data and integrating methods for proteomics (e.g., protacce[17]), spatial transcriptomics (e.g., stlearn[18], and epigenomics (e.g., multivelo[18]).

Downstream Analysis: Leveraging the large-scale corpus database search functionality of cellxgene[8], which enables the mapping of local tissue structures to broader global tissue contexts. This capability allows for the prediction of future global differentiation trajectories, providing users with a more comprehensive understanding of cellular dynamics across different scales. By integrating these features, CellFateExplorer not only facilitates detailed local analysis but also extends its predictive power to global tissue-level insights, offering a more holistic view of cell fate and differentiation processes.

In summary, CellFateExplorer establishes a robust platform for systematic cell trajectory analysis, with ample room for future improvements.

Availability

All source codes are available at https://github.com/HuangDDU/CellFateExplorer. Document can be accessed by https://cellfateexplorer.readthedocs.io. Pipy package will be released in the near future.

Reference

1. Gulati, G.S., et al., *Profiling cell identity and tissue architecture with single-cell and spatial transcriptomics.* Nature Reviews Molecular Cell Biology, 2024: p. 1-21.
2. Setty, M., et al., *Characterization of cell fate probabilities in single-cell data with Palantir.* Nature biotechnology, 2019. **37**(4): p. 451-460.
3. Qiu, X., et al., *Single-cell mRNA quantification and differential analysis with Census.* Nature methods, 2017. **14**(3): p. 309-315.
4. Bergen, V., et al., *Generalizing RNA velocity to transient cell states through dynamical modeling.* Nature biotechnology, 2020. **38**(12): p. 1408-1414.
5. Qiu, X., et al., *Mapping transcriptomic vector fields of single cells.* Cell, 2022. **185**(4): p. 690-711. e45.
6. Kang, M., et al., *Mapping single-cell developmental potential in health and disease with interpretable deep learning.* bioRxiv, 2024: p. 2024.03. 19.585637.
7. Saelens, W., et al., *A comparison of single-cell trajectory inference methods.* Nature biotechnology, 2019. **37**(5): p. 547-554.
8. Program, C.C.S., et al., *CZ CELLxGENE Discover: a single-cell data platform for scalable exploration, analysis and modeling of aggregated data.* Nucleic Acids Research, 2025. **53**(D1): p. D886-D900.
9. Sumanaweera, D., et al., *Gene-level alignment of single-cell trajectories.* Nature Methods, 2025. **22**(1): p. 68-81.
10. Aibar, S., et al., *SCENIC: single-cell regulatory network inference and clustering.* Nature methods, 2017. **14**(11): p. 1083-1086.
11. Kamimoto, K., et al., *Dissecting cell identity via network inference and in silico gene perturbation.* Nature, 2023. **614**(7949): p. 742-751.
12. Wolf, F.A., et al., *PAGA: graph abstraction reconciles clustering with trajectory inference through a topology preserving map of single cells.* Genome biology, 2019. **20**: p. 1-9.
13. Street, K., et al., *Slingshot: cell lineage and pseudotime inference for single-cell transcriptomics.* BMC genomics, 2018. **19**: p. 1-16.
14. Weiler, P., et al., *CellRank 2: unified fate mapping in multiview single-cell data.* Nature Methods, 2024: p. 1-10.
15. Marco, E., et al., *Bifurcation analysis of single-cell gene expression data reveals epigenetic landscape.* Proceedings of the National Academy of Sciences, 2014. **111**(52): p. E5643-E5650.
16. Pedregosa, F., et al., *Scikit-learn: Machine learning in Python.* the Journal of machine Learning research, 2011. **12**: p. 2825-2830.
17. Gorin, G., V. Svensson, and L. Pachter, *Protein velocity and acceleration from single-cell multiomics experiments.* Genome biology, 2020. **21**: p. 1-6.
18. Pham, D., et al., *Robust mapping of spatiotemporal trajectories and cell–cell interactions in healthy and diseased tissues.* Nature communications, 2023. **14**(1): p. 7739.