Cytopath

Release 0.1.9

Revant Gupta

Jul 18, 2023

CONTENTS

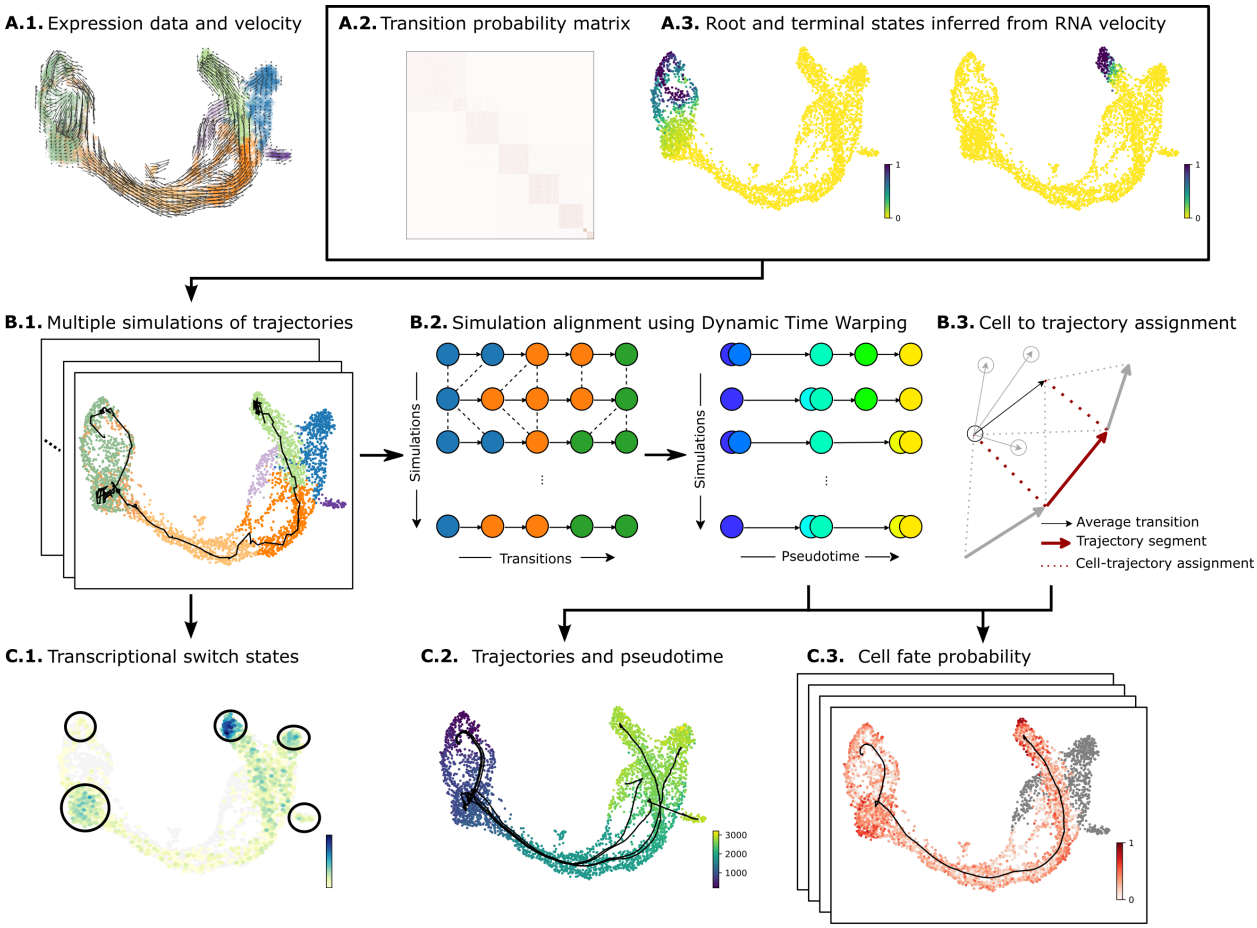
Note: This project is under active development.

Cytopath is a method for trajectory inference with single-cell RNA sequencing data. Transcriptional activity information from RNA velocity of single cells is used to define a Markov chain model; simulation of this model yields an ensemble of possible differentiation trajectories that are used to estimate the lineage path.

Cytopath can infer trajectories with or without root/terminal state supervision. No topological constraints (e.g. a tree structure) are placed on the inference as each trajectory is modelled independently. The number of trajectories to be inferred can either be defined or estimated in an unsupervised fashion. Subsequent statistical analysis reveals the topological and molecular characteristics of the differentiation process.

Cytopath can model complex behaviours like cycling and convergence as well as co-occurring combinations of multiple processes. Read more here.

Check out the *Notebooks* section for a demonstration of Cytopath on publicly available datasets.



CHAPTER

ONE

INSTALLATION

Cytopath requires Python 3.9. We recommend using Miniconda and setting up a new conda environment.

```
conda create -y -n cytopath_env python==3.9
conda activate cytopath_env
```

1.1 PyPI

Install cytopath from PyPI using pip

```
pip install cytopath
```

1.2 Dependencies

Direct dependencies of Cytopath will be installed automatically in the step above however, additional dependencies of scvelo will need to be installed manually.

```
conda install -c conda-forge python-igraph louvain
```

1.3 Jupyter notebook

To run the tutorials install jupyter notebooks.

```
conda install notebook
```

CHAPTER

TWO

USAGE

The entire process of inferring trajectories with Cytopath is divided into two steps. In the first step, a transition probability matrix is used to simulate possible differentiation paths. In the second step, the simulations are used to estimate trajectories from root states to each terminal region (cluster containing terminal states).

2.1 Pre-processing

We assume that the user has already performed velocity analysis at this point and has the processed anndata object in memory. Following is an example of an anndata processed with scvelo.

```
AnnData object with n_obs × n_vars = n_cells × n_genes
    obs: 'root_cells', 'end_points', 'louvain'
    var: 'velocity_gamma', 'velocity_qreg_ratio', 'velocity_r2', 'velocity_genes'
    uns: 'neighbors', 'pca', 'umap', 'velocity_graph', 'velocity_graph_neg', 'velocity_
↪params'
    obsm: 'X_pca', 'X_umap', 'velocity_umap'
    varm: 'PCs'
    layers: 'Ms', 'Mu', 'ambiguous', 'matrix', 'spliced', 'unspliced', 'variance_velocity
↪', 'velocity'
    obsp: 'connectivities', 'distances'
```

Store the transition probability matrix,

```
adata.uns['T_forward'] = scv.utils.get_transition_matrix(adata)
```

2.2 Markov chain sampling

The sampling procedure requires a transition probability matrix (parameter: matrix_key), a clustering that will be used to define terminal regions (parameter: cluster_key) and root/terminal state probabilities (stored under root_cells and end_points obs keys). If a clustering is not specified then Louvain is used by default.

The auto_adjust parameter will automatically and dynamically select values for technical parameters of the sampling process based on the properties of the dataset. We advise keeping this parameter set to True.

Additionally, set the number of vCPUs (parameter: num_cores) for significantly faster performance. If you wish to return a copy of the anndata object then set copy parameter to True.

```
import os
import cytopath

cytopath.sampling(adata, auto_adjust=True, matrix_key = 'T_forward', cluster_key =
↪'louvain',
                  num_cores=os.cpu_count()-1, copy=False)
```

If the user wishes to manually specify root and terminal cell states, then use the following,

```
end_points = np.array(adata.obs.end_points > 0.99)[0] # Numerical index; use any␣
↪selection criteria
root_cells = np.array(adata.obs.root_cells > 0.99)[0] # Numerical index; use any␣
↪selection criteria

cytopath.sampling(adata, auto_adjust=True, matrix_key = 'T_forward', cluster_key =
↪'louvain',
                  end_points=end_points, root_cells=root_cells, num_cores=os.cpu_count()-
↪1, copy=False)
```

Finally, entire clusters can be designated as root or terminal regions as a means of incorporating biological know-how not reflected in velocity-based selection of root/terminal states

```
# Example designation of root/terminal clusters
end_clusters = ['end_cluster_1', 'end_cluster_2']
root_clusters = ['root_cluster_1']

cytopath.sampling(adata, auto_adjust=True, matrix_key = 'T_forward', cluster_key =
↪'louvain',
                  end_clusters=end_clusters, root_clusters=root_clusters, num_cores=os.
↪cpu_count()-1, copy=False)
```

2.3 Trajectory inference

Trajectory inference with default parameters will require the anndata to contain a PCA embedding (uns: 'X_pca') and UMAP (uns: 'X_umap'). The latter can be substituted for any 2D embedding being used to visualise the data (parameter: basis).

```
cytopath.trajectories(adata, num_cores=os.cpu_count()-1)
```

2.4 Plotting

```
cytopath.plot_trajectories(adata, basis='umap')
```

2.5 Inference output

The trajectories inferred by Cytopath are composed of segments. Cells are aligned to these trajectory segments to determine their relative position along the trajectory (pseudotime) and relative association with multiple trajectories (cell-fate).

The complete inference output containing all cell-trajectory alignments is stored under the following key,

```
adata.uns['trajectories']['cells_along_trajectories_each_step']
```

Inference output summarised at the single cell level is stored under.

```
adata.uns['trajectories']['cells_along_trajectories']
```

Simulations are stored under,

```
adata.uns['samples']
```

Trajectory coordinates are stored under,

```
adata.uns['trajectories']
```

CHAPTER

THREE

NOTEBOOKS

Demonstration of Cytopath on publicly available single cell RNA sequencing datasets: cytopath-notebooks