AUTHOR(S)
Sousa, A. G. G., Smolander, J., Junttila, S., & Elo, L. L.

TITLE
Inferring Tree-Shaped Single-Cell Trajectories with Totem

YEAR
2024

DOI
10.1007/978-1-0716-3886-6_9

CITATION
Sousa, A.G.G., Smolander, J., Junttila, S., Elo, L.L. (2024). Inferring Tree-Shaped Single-Cell Trajectories with Totem. In: Azad, R.K. (eds) Transcriptome Data Analysis. Methods in Molecular Biology, vol 2812. Humana, New York, NY. https://doi.org/10.1007/978-1-0716-3886-6_9

VERSION
Accepted Manuscript

Title: *Inferring Tree-Shaped Single-Cell Trajectories with Totem*

Running Head: Totem protocols for single-cell trajectory inference

António G.G. Sousa[1] (aggode@utu.fi), Johannes Smolander[1] (johannes.smolander@utu.fi), Sini Junttila[1] (simaju@utu.fi), Laura L. Elo[1,2,*] (laura.elo@utu.fi)

[1] Turku Bioscience Centre, University of Turku and Åbo Akademi University, Tykistökatu 6, 20520 Turku, Finland

[2] Institute of Biomedicine, University of Turku, 20520 Turku, Finland

* laura.elo@utu.fi

Abstract

Single-cell transcriptomics allows unbiased characterization of cell heterogeneity in a sample by profiling gene expression at single-cell level. These profiles capture snapshots of transient or steady states in dynamic processes, such as cell cycle, activation, or differentiation, which can be computationally ordered into a “flipbook” of cell development using trajectory inference methods. However, prediction of more complex topology structures, such as multifurcations or trees, remains challenging. In this chapter, we present two user-friendly protocols for inferring tree-shaped single-cell trajectories and pseudotime from single-cell transcriptomics data with *Totem*. *Totem* is a trajectory inference method that offers flexibility in inferring both non-linear and linear trajectories, and usability by avoiding the cumbersome fine-tuning of parameters. The QuickStart protocol provides a simple and practical example, whereas the GuidedStart protocol details the analysis step by step. Both protocols are demonstrated using a case study of human bone marrow CD34+ cells, allowing the study of the branching of three lineages: erythroid, lymphoid, and myeloid. All the analyses can be fully reproduced in Linux, MacOS, and Windows operating systems (amd64 architecture) with >8Gb of RAM using the provided docker image distributed with notebooks, scripts, and data in Docker Hub (*elolab/repro-totem-ti*). These materials are shared online under open-source license at https://elolab.github.io/Totem-protocol.

Key words

single-cell RNA-seq, trajectory inference, tree-shaped topology, pseudotime, cell connectivity, Totem, data analysis, bioinformatics

1 Introduction

Advancements in single-cell technologies allow the measurement of hundreds to thousands of molecules across thousands of individual cells in a single sample. The sample may represent a mixture of cell populations collected at a given time, such as bone marrow CD34+ cells during hematopoiesis [1], or a collection of timepoints during a developmental process such as embryogenesis [2]. In either case, the heterogeneity intrinsic to cell populations is likely to represent related cell stages in dynamic processes such as cell cycle, activation, differentiation, or transition states driven by gene expression and/or chromatin accessibility [3]. To create a chronological "flipbook" of cell development, computational prediction of the pseudotime order of these cell snapshots can be used for trajectory inference. Such method has successfully unravelled, for instance, the developmental bifurcation between circulating and germinal center human follicular regulatory T cells from regulatory T cells [4].

Single-cell trajectory inference (TI) methods predict the time-dependent relatedness between cells by modeling molecular data that functionally characterize cell populations, such as gene or protein expression. These methods can be broadly divided into three main categories depending on whether they are applied on the (1) full gene space, (2) reduced gene space, or (3) RNA velocity [5]. Full gene-space methods rely on probabilistic models of gene expression, whereas dimensionality reduction methods reduce the gene space before projecting a spanning tree or graph across cells [5, 6]. RNA velocity uses the ratio of spliced and unspliced transcripts to derive cell state transitions [7]. This chapter focuses on dimensionality reduction methods (see Ding et al. 2022 [5] review for other methods).

Dimensionality reduction based methods have two sequential steps: (1) dimensionality reduction, and (2) trajectory modelling [6]. The first step involves reducing the gene and/or cell space while preserving most of the biological information and eliminating technical noise. This can be accomplished by low-dimensional reduction techniques such as principal component analysis (PCA), multidimensional scaling (MDS), t-distributed stochastic neighbor embedding (t-SNE) [8], uniform

manifold approximation and projection (UMAP) [9], and clustering, which leads to identification of cell centroids (milestones) [5, 6]. The second step involves drawing a spanning tree or graph across the cells or milestones, for example with a Minimum Spanning Tree (MST) and principal curves algorithms [6]. The trajectories inferred can assume several types of topologies, such as linear, bifurcation, multifurcation, tree, cycle, connected, or disconnected, which can be fixed or free, depending on the method used.

According to Saelens et al. 2019 [10], the tree-shaped *Slingshot* TI method was ranked as the overall best method among the 45 TI methods benchmarked on more than 300 data sets. However, a comparison between *Slingshot* and *Totem*, a new tree-shaped TI method developed recently by Smolander et al. 2022 [11], showed that *Totem* outperformed *Slingshot* for non-linear trajectories. The more complex topology structures, such as multifurcations or trees, are still challenging to predict with current methods. Therefore, the focus of this chapter is on the tree-shaped *Totem*'s TI method, which promises to address these challenges.

Totem is a user-friendly tree-shaped TI method implemented as an R software package by Smolander et al. 2022 [11]. It clusters cells *L* times using latent dimensions of the reduced gene space with the CLARA *k*-medoids algorithm [12]. For each clustering result, a MST [13] is built and smoothed using the principal curves algorithm from *Slingshot* [14]. *Totem* introduces a novel concept of cell connectivity, which is the ratio of a cell cluster's connections (edges) by the number of clusters (vertices) and is calculated for every clustering and respective MST result. These cell connectivity values are scaled and averaged across the MSTs to identify branching points and select the best clustering results. In addition, *Totem* was designed to be user friendly by avoiding the cumbersome fine-tuning of parameters during trajectory inference, making it a suitable tool for beginners.

This chapter provides guidelines to perform tree-shaped single-cell trajectory and pseudotime inference analysis from processed single-cell transcriptomics data using *Totem*. We showcase the functionality, flexibility, and user-friendliness of *Totem* using two analysis protocols: QuickStart, a simple and practical protocol, and GuidedStart, a detailed step-by-step protocol. Both protocols are

applied to a real scRNA-seq data set on hematopoiesis. All the analyses can be reproduced in Linux, MacOS and Windows operating systems running under a amd64 architecture with >8Gb of RAM by using a docker image distributed with notebooks, scripts, and data in Docker Hub (*elolab/repro-totem-ti*). An online resource sharing the content materials described herein is available under open-source license at https://elolab.github.io/Totem-protocol.

2 Materials

2.1 Data analysis environment

The protocols presented in this chapter are distributed as R markdown (Rmd) notebooks which can run in the RStudio software. To allow flexibility, the notebooks were converted to R scripts. The notebooks and the respective scripts are provided in a containerized docker image publicly available at Docker Hub (*elolab/repro-totem-ti*), containing Ubuntu 20.04.5 LTS (Focal), R (v.4.2.1), RStudio server (2022.07.2 Build 576), *Totem* (v.0.99.2) R package, and all the system and R libraries, as well as the data required to reproduce the protocols described herein. Instructions to run the container are provided in this protocol and in the github repository https://github.com/elolab/Totem- protocol. The analyses were tested on a server using Slurm scheduler (v.22.05.5) and Singularity (apptainer version 1.1.5-2.el7) container (*elolab/repro-totem-ti*) with 8Gb of RAM memory and 4 CPUs. All the code used in this protocol was made open source under the General Public Licence (GPL) 3.0.

2.2 Case study: hematopoiesis scRNA-seq data set

To showcase the trajectory inference with Totem in practice, we used the scRNA-seq data set of human CD34+ bone marrow cells (replicate 1) from Setty et al. 2019 [15], which involves a known tree-shaped trajectory and is publicly available at the Human Cell Atlas Portal (link: https://data.humancellatlas.org/explore/projects/091cf39b-01bc-42e5-9437-f419a66c8a45). The *anndata* object "*human_cd34_bm_rep1.h5ad"* comprises counts, processed data (e.g. the

dimensionality reduction t-SNE and transformed log-normalized counts), and cell metadata (e.g. cluster identities and Palantir's pseudotime estimates). The *anndata* object was downloaded from the Human Cell Atlas Portal and imported to R as a *SingleCellExperiment* (SCE) class object using the function *readH5AD()* from *zellkonverter* (v.1.8.0 [16]). Counts were log1p normalised and cell type annotations were retrieved from Palantir's github issue: https://github.com/dpeerlab/Palantir/issues/40. The parsed SCE object was exported as a RDS file "*human_cd34_bm_rep1.rds*" and it is distributed in the docker image "*repro-totem-ti*". The object is also available in the Zenodo repository at: https://doi.org/10.5281/zenodo.7845709. All these steps, with the exception of updating the download url due to time limit use, can be reproduced with the script "*download_h5ad_to_SCE_rds_script.R*" (available in the github repository https://github.com/elolab/Totem- protocol).

3 Methods

This section describes two *Totem* TI protocols: **QuickStart** and **GuidedStart**. **QuickStart** provides a simpler approach to TI, while **GuidedStart** offers a more detailed description of the parameter functions and intermediate results through intuitive visualisations, to obtain a refined trajectory (see Fig. 1). The **QuickStart** protocol is recommended for situations where one wants to preserve the dimensionality reductions (PCA, t-SNE, UMAP) obtained during a scRNA-seq integration and/or clustering analysis for trajectory inference. This is particularly useful for case studies where one obtains an integrated embedding after a batch-correction analysis, such as in *Seurat* [17] or *Scanorama* [18], overcoming potentially erroneous trajectory predictions that could be driven by batch effects. The **GuidedStart** protocol, on the other hand, is recommended for situations where one wants to perform a distinct dimensionality reduction method (Landmark MDS, diffusion maps [19]). To this end, the **GuidedStart** protocol provides step-by-step guidance on how to perform TI with *Totem*, starting from log-normalised counts till topology projection. Both protocols are demonstrated

using the same data set of the human CD34+ bone marrow cells described in the **Materials** section to explore hematopoiesis, particularly erythroid, lymphoid and myeloid lineages.

3.1 Software installation

A docker image is available on Docker Hub (*elolab/repro-totem-ti*), containing the data, notebooks, scripts, and all software packages required for the analysis. This provides cross-OS, distributable, and reproducible analysis across Linux, MacOS, and Windows operating systems (amd64 architecture). However, all the analyses can also be reproduced using the RStudio desktop/server application or even just the R console, as long as *Totem* R package and its dependencies are installed. In this case, the data needs to be first downloaded from Zenodo in order to run the analyses outside the docker container: https://doi.org/10.5281/zenodo.7845709. Instructions for the installation of the individual software packages are beyond the scope of this chapter as these are provided in the docker image. All the R markdown notebooks and the equivalent R scripts are available in github (https://github.com/elolab/Totem-protocol) for user flexibility.

Installation of docker depends on the operating system and instructions can be found in the official website (https://www.docker.com/). For Ubuntu, the docker application can be installed under the terminal with the *apt-get* package manager toolkit using the following command:

```
apt-get install docker.io
```

3.2 Launching docker `repro-totem-ti` container

The github repository https://github.com/elolab/Totem-protocol contains the "*Dockerfile*" for generating the "*repro-totem-ti*" docker image, as well as instructions for building and converting it to a singularity image, which can facilitate its use in a server environment. In addition, it also provides bash scripts for launching the docker image locally (*run_docker.sh*) or on a server environment with Singularity through the Slurm workload manager (*run_slurm_singularity.sh*). In this section, we

describe the steps to launch the "*repro-totem-ti*" image locally with docker (v.20.10.17, build 100c701).

To allow the container to import/export data/files from the local volumes, these need to be bound to the container. To do this, run the following command in the Linux terminal to create the following directory structure locally under a directory/folder of your choice:

```
mkdir data scripts notebooks results
```

Assuming that docker is installed, pull the "*repro-totem-ti*" image from Docker Hub by typing the command in the Linux terminal:

```
docker pull elolab/repro-totem-ti
```

Then, launch the "*repro-totem-ti*" container in the Linux terminal by binding the "`results`" folder created above to the container's folder with the command:

```
docker run --rm -ti -e PASSWORD=Totem -p 8787:8787 \

-v $PWD/results:/home/rstudio/results \

repro-totem-ti
```

Finally, go to the browser of your choice (e.g., Chrome, Firefox) and type the following link:

```
http://localhost:8787
```

Use the following credentials to login in RStudio Server:

```
Username: rstudio

Password: Totem
```

The **Files, Plots, Packages, Help, Viewer, Presentations** RStudio pane layout displays the directory structure under the **Files** menu, including data, notebooks, results, and scripts. The data folder contains `human_cd34_bm_rep1.rds`, which stores the *SingleCellExperiment* class object of the human CD34+ bone marrow scRNA-seq data set used for all the analyses. The notebooks folder contains R markdown notebooks (Rmd) to reproduce analyses, including the **QuickStart** protocol (`QuickStart_Totem.Rmd`) and the **GuidedStart** protocol (`GuidedStart_Totem.Rmd`), as well as the bibliography used in the Rmd notebooks (`references.bib`). The results folder is empty and intended to save results, while the scripts folder contains R scripts equivalent to Rmd notebooks, including the **QuickStart** (`QuickStart_Totem.R`) and the **GuidedStart** (`GuidedStart_Totem.R`) protocols, and the R script used to generate the `human_cd34_bm_rep1.rds` file (`download_h5ad_to_SCE_rds_script.R`).

3.3 QuickStart protocol

To access the "*QuickStart_Totem.Rmd*" notebook in RStudio, navigate to the **Files** menu and open the **notebooks** folder. This notebook is divided into four parts: (1) Preprocessing; (2) *Totem*'s TI; (3) Visualization; and (4) Pseudotime. The Preprocessing section sets up the environment, imports and prepares data for *Totem*. The *Totem*'s TI section contains a two-step workflow which includes clustering and multiple spanning trees (MSTs), followed by smoothing the selected clustering/MSTs. The Visualization section of the notebook visualizes *Totem*'s cell connectivity, clustering, and trajectory's topology. Finally, the Pseudotime section performs re-rooting and compares *Totem*'s pseudotime estimates to those of *Palantir*.

The set of code blocks highlighted in the next subsections can be run by clicking over the start button icon that appears in every R chunk of the "*QuickStart_Totem.Rmd*" notebook, or by putting the mouse cursor over the line of code to be run and pressing CTRL+ENTER keys. Alternatively, the code can be copy/pasted into the R console.

3.3.1 Preprocessing

First, load the R packages required to run the analyses into the R environment:

```
library("dplyr")
library("Totem")
library("scater")
library("ggplot2")
library("SingleCellExperiment")
```

Set the seed to ensure reproducibility:

```
set.seed(1204)
```

Import the scRNA-seq data set of human CD34+ bone marrow cells as `SingleCellExperiment` R object:

```
sce <- readRDS(file = "../data/human_cd34_bm_rep1.rds")
```

The `sce` object contains the assays `counts`, `logcounts`, `scaled`; the cell metadata `clusters`, `palantir_pseudotime`, `palantir_diff_potential`, `cell_types_long`, `cell_types_short`; the dimensionality reduction `tsne`; and several metadata such as `cluster_colors`. In total, the data has 14624 genes and 5780 cells.

Prepare the data to Totem by filtering non-expressed genes:

```
sce <- PrepareTotem(object = sce)
```

3.3.2 *Totem*'s TI

The mandatory input to Totem is a `SingleCellExperiment` object, which in this case is `sce`. This object needs to compulsorily contain the `logcounts` assay (which can be checked with `assayNames(sce)` function) and at least one dimensionality reduction (which can be checked with `reducedDimNames(sce)`). *Totem* uses the first dimensionality reduction result in `reducedDimNames(sce)` for clustering, which in this case is `tsne` (see **Note 1**). Thus, it is important to make sure that the first dimensionality reduction result is the right one to use for clustering.

The two-step `Totem`'s TI workflow consists of:

1. CLARA (*k*-medoids) clustering and multiple spanning trees (MSTs): `RunClustering()`
2. Smoothing selected clustering/MSTs with the principal curves algorithm: `SelectClusterings()` followed by `RunSmoothing()`

To run the two-step *Totem*'s TI workflow with default options (see **Note 2**), set the seed to maintain reproducibility and use the following code:

```
set.seed(123)
sce <- RunClustering(object = sce) %>%
     SelectClusterings(object = .) %>%
     RunSmoothing(object = .)
```

All the results are stored in the `sce` object as a list of elements (`cell.connectivity`, `all.clustering`, `all.clustering.scores`, `selected.clustering`, `dynwrap_trajectory`, `slingshot_trajectory`) in the metadata slot named totem (see **Note 3**).

3.3.3 Visualization

To explore the results and support decisions, *Totem* provides several visualization functions to explore cell connectivity (`VizCellConnectivity()`, see **Note 4**), clustering/MST (`VizMST()`), and trajectory and pseudotime (`VizSmoothedTraj()`).

To visualize the cell connectivities, retrieve the `tsne` dimensionality reduction using the `reducedDim()` function and plot them using the `VizCellConnectivity()` function (Fig. 2):

```
dim_red <- reducedDim(sce, "tsne")
VizCellConnectivity(object = sce, viz.dim.red = dim_red)
```

To visualize the best selected clustering/MST result, retrieve its name and visualize it using the `VizMST()` function (Fig. 3A):

```
select.cluster <- names(metadata(sce)$totem$slingshot_trajectory)
VizMST(object = sce, clustering.names = select.cluster,
       viz.dim.red = dim_red)
```

For comparison, the projection can also be visualized together with cell types from the original study (Fig. 3B):

```
plotReducedDim(sce, dimred = "tsne",
               color_by = "cell_types_short") +
     scale_color_manual(name= "Cell Types",
                        values=as.character(unlist(metadata(sce)$clust
                        er_colors)[!duplicated(levels(sce$cell_types_s
                        hort))])) +
     ggtitle("Cell types") +
     theme_void() +
```

```
    theme(legend.position = "bottom")
```

3.3.4 Pseudotime

To visualize Totem's pseudotime, define the root of the cell trajectory to cluster 15, which is one of the clusters that correspond to the more immature HSCs (hematopoietic stem cells, see **Note 5**), and use the function `VizSmoothedTraj()` (Fig. 4A):

```
root.cluster <- 15
sce <- ChangeTrajRoot(object = sce,
                      traj.name = select.cluster,
                      root.cluster = root.cluster)
VizSmoothedTraj(object = sce, traj.names = select.cluster,
                viz.dim.red = dim_red, plot.pseudotime = TRUE)
```

To compare *Totem*'s pseudotime with that of Palantir from the original study, visualize Palantir's pseudotime using the function `plotReducedDim()` (Fig. 4B):

```
plotReducedDim(sce, dimred = "tsne",
               colour_by = "palantir_pseudotime") +
               ggtitle("Palantir's pseudotime")) +
               theme_void() +
               theme(legend.position = "bottom")
```

Despite the good correlation between the pseudotimes obtained using *Totem* and Palantir and the correct prediction of terminal states once the root was provided, the lymphoid lineage was wrongly predicted. CLP (common lymphoid progenitor) was predicted to diverge from MyP (myeloid progenitor) instead of HSC (hematopoietic stem cells), as it should have been. One of the reasons that may explain this wrong prediction is that the low-dimensional representation used for clustering may

not fairly represent the relationship between cell types. The CLP population appears farther apart from the HSC than the MyP cluster, making it more likely to branch from the latter than the former. In general, the user should be critical about lineages towards cell populations/clusters that do not show a continuous development in the dimensionality reduction projection used for clustering.

3.4 GuidedStart protocol

To access the "*Guided_Totem.Rmd*" notebook in RStudio, navigate to the **Files** menu and open the **notebooks** folder. This notebook is divided into nine parts: (1) Prepare data; (2) Feature selection; (3) Dimensionality reduction; (4) Clustering (& MST); (5) Cell connectivity; (6) Select clusterings; (7) Smoothing MSTs; (8) Define root; and (9) Pseudotime. Each one of these parts is summarised below:

1. Prepare data: Importing data to R and preparing it for analysis with *Totem.*
2. Feature selection: Selecting 2K highly variable features for dimensionality reduction with *scran.*
3. Dimensionality reduction: Dimensionality reduction for clustering (PCA) and visualization (UMAP) using *Totem.*
4. Clustering and MST: CLARA *k*-medoids clustering and MSTs using *Totem.*
5. Cell connectivity: Visualizing and interpreting cell connectivity estimated by *Totem.*
6. Select clustering: Selecting and visualizing top six clustering and MST results using *Totem.*
7. Smooth MST: Smoothing the selected MSTs with the principal curves algorithm using *Totem.*
8. Define a root: Defining the most probable root for the inferred trajectory.
9. Pseudotime: Visualizing *Totem*'s pseudotime and comparing it with Palantir's pseudotime.

3.4.1 Preprocessing

The first step of the **GuidedStart** protocol is exactly the same as the **(1) Preprocessing** step in the **QuickStart** protocol. However, the dimension reduction `tsne` is removed from the `sce` object to focus on exploring other dimensionality reduction methods for clustering and visualization:

```
set.seed(1204)
sce <- readRDS(file = "../data/human_cd34_bm_rep1.rds")
reducedDim(sce, "tsne") <- NULL
sce <- PrepareTotem(object = sce)
```

3.4.2 Feature selection

The next step of the protocol is feature selection. It is not mandatory but highly recommended to avoid selection of uninformative genes, such as those with low abundance or invariant expression across different cells, and to increase the computational speed of downstream steps due to the reduced dimensionality. Selection of 2K most highly variable genes (HVG) can be done using the `scran` R package:

```
var.genes <- scran::modelGeneVar(sce)
hvg <- scran::getTopHVGs(var.genes, n = 2000)
```

3.4.3 Dimensionality reduction

After feature selection, perform a PCA for clustering with 50 PCs using the `hvg` selected previously:

```
sce <- RunDimRed(object = sce,
                 dim.red.method = "pca",
                 dim.red.features = hvg,
                 dim.reduction.par.list = list(ndim=50))
```

In order to see which PCs explain most of the biological variability in the data, an elbow plot can be created based on the standard deviation of the PCs along with visualization of the cells projected onto the first PCs (Fig. 5):

```
reducedDim(sce, "pca") %>%
     apply(X = ., MARGIN = 2, FUN = function(x) sd(x)) %>%
     as.data.frame(.) %>%
     `colnames<-`("Standard Deviation") %>%
     mutate("PCs" = factor(1:nrow(.), levels = 1:nrow(.))) %>%
     ggplot(data = ., mapping = aes(x = PCs,
                                    y = `Standard Deviation`)) +
          geom_point() +
          theme_bw()


reducedDim(sce, "pca") %>%
     as.data.frame(.) %>%
     mutate("Cell_ID" = row.names(.)) %>%
     cbind(., colData(sce)) %>%
     ggplot(data = ., mapping = aes(x=comp_1,
                                    y=comp_2,
                                    color=cell_types_short)) +
          geom_point() +
          labs(x = "PC1", y = "PC2") +
          scale_color_manual(values=as.character(unlist(metadata(s
     ce)$cluster_colors)[!duplicated(levels(sce$cell_types_short))
     ])) +
          theme_bw()
```

To perform clustering, the first six PCs were selected based on the previous visualizations (see **Note 1**):

```
pick.pcs <- 1:6
reducedDim(sce, "pca") <- reducedDim(sce, "pca")[, pick.pcs ]
```

To create a two-dimensional representation of the data for visualization, UMAP is selected as the dimensionality reduction method using the same options as with PCA and saved to the variable `dim_red`. The projection can then be visualized together with cell types from the original study (Fig. 6).

```
set.seed(123)
sce <- RunDimRed(object = sce,
                 dim.red.method = "umap",
                 dim.red.features = hvg,
                 dim.reduction.par.list = list(ndim=2,
                                               pca_components = 6))
dim_red <- reducedDim(sce, "umap")
plotReducedDim(object = sce, dimred = "umap",
               colour_by = "cell_types_short", point_size=0.5) +
     scale_color_manual(name=            "Cell            Types",
values=as.character(unlist(metadata(sce)$cluster_colors)[!duplicat
ed(levels(sce$cell_types_short))])) +
     theme_void()
```

Finally, the UMAP reduction is removed from the `sce` object to retain only `pca`:

```
reducedDim(sce, "umap") <- NULL
```

3.4.4 Clustering and MST

To cluster the single-cell data, use the CLARA algorithm (*k*-medoids) on the PCA-reduced data 10K times (`N.clusterings = 10000`), ranging from 3 to 20 clusters (`k.range = 3:20`, see **Note 2**) and excluding clusters with less than 5 cells (`min.cluster.size = 5`):

```
set.seed(123)
sce <- RunClustering(object = sce, k.range = 3:20,
                     min.cluster.size = 5, N.clusterings = 10000)
```

3.4.5 Cell connectivity

To visualize the cell connectivity, the function `VizCellConnectivity()` is used, with the UMAP dimension reduction as input (stored in the variable `dim_red`):

```
VizCellConnectivity(object = sce, viz.dim.red = dim_red)
```

This information can be used to support the decision on the most likely trajectory (Fig. 7, see **Note 4**). Additionally, the cell connectivity can be used to select the clustering method (see the next section).

3.4.6 Select clusterings

The next step is to select the top clustering results based on several metrics and inspected for further visualizations (see **Note 6**). Select the top six clustering results based on the variance ratio criterion and cell connectivity (method 3) for MST calculation:

```
sce <- SelectClusterings(sce, selection.method = 3,
                         selection.N.models = 6,
                         selection.stratified = FALSE,
                         prior.clustering = NULL)
```

After selecting the top six clustering results, they can be visualized using the `VizMST()` function (Fig. 8):

```
select.clusters <- ReturnTrajNames(sce)
VizMST(object = sce, clustering.names = select.clusters,
       viz.dim.red = dim_red)
```

3.4.7 Smoothing MSTs

Smooth the selected top six MSTs with the principal curves algorithm and visualize the smoothed trajectories:

```
sce <- RunSmoothing(sce)
smooth.msts.names <- ReturnTrajNames(sce)
VizSmoothedTraj(object = sce,
                traj.names = smooth.msts.names,
                viz.dim.red = dim_red,plot.pseudotime = FALSE)
```

From the top six smoothed MST trajectories presented (Fig. 9), the only difference is in the MoP and DCP lineages and the branching from these lineages to the CLP (see Fig. 6). The CLP needs to diverge before the MyP, and MoP and DCP should diverge from MyP. The clustering/MST result `8.135` seems to meet these expectations and is selected below as the elected trajectory.

3.4.8 Define a root

The following defines the root of the trajectory of the clustering/MST result `8.135` as the cluster number 2, which corresponds to the most immature HSCs (see **Note 5**):

```
select.traj <- "8.135"
root.cluster <- 2
sce <- ChangeTrajRoot(object = sce, traj.name = select.traj,
```

```
                    root.cluster = root.cluster)
```

3.4.9 Pseudotime

Plot the selected trajectory highlighting the pseudotime (Fig. 10A):

```
  VizSmoothedTraj(object = sce,
                  traj.names = select.traj,
                  viz.dim.red = dim_red, plot.pseudotime = TRUE)
```

To compare *Totem*'s pseudotime (Fig. 10A) with *Palantir*'s pseudotime, visualize Palantir's pseudotime using the function `plotReducedDim()` (Fig. 10B):

```
reducedDim(sce, "umap") <- dim_red
plotReducedDim(sce, dimred = "umap",
               colour_by = "palantir_pseudotime") +
     ggtitle("Palantir's pseudotime")) +
     theme_void() +
     theme(legend.position = "bottom")
```

The pseudotime and trajectory obtained with *Totem* agrees well with *Palantir*. Contrary to the inferred trajectory with the t-SNE projection in the **QuickStart** protocol, by using a more adequate dimensionality reduction projection, such as PCA, *Totem* was able to correctly predict the lineage related to the CLP (see **Note 1**).

The trajectory of interest obtained with Totem can be further used to perform downstream analyses such as the identification of genes supporting that same trajectory with third party tools developed for this purpose. *Totem* allows converting the trajectory of interest to formats compatible with the *dyno* [10] and *tradeSeq* [21] for trajectory differential gene expression (see **Note 7**).

4 Notes

Note 1: Dimensionality reduction to use for clustering and visualization

The minimal requirement to run *Totem*'s TI method is a dimensionality reduction embedding. This can be obtained either from an upstream data analysis, such as integration or clustering as shown in **QuickStart**, or *de novo* from the normalized assay (`logcounts`), as shown in **GuidedStart**. In either case, the low-dimensional representation chosen for clustering should fairly represent the dynamic process being studied, such as cell cycle, activation, or differentiation, as it greatly impacts the inferred trajectories. Choosing the appropriate dimensionality reduction method and the number of dimensions is a difficult task. Typically, two or more dimensions need to be selected depending on the data set and the dimensionality reduction method chosen [6, 20]. The first dimension may represent the differentiation gradient and the second the cell cycle [6]. As demonstrated in the **QuickStart** protocol, it is important to note that using the same dimensionality reduction embedding for clustering and visualization might not be adequate. While t-SNE might be a proper method to visualize cell populations, other methods such as PCA might be more suitable for cell trajectory clustering [20]. *Totem* does not deal with disconnected trajectories and, thus, the user should be very critical and doubtful about branchings towards milestones that appear segregated in the low-dimension projection, as it is the case of the common lymphoid progenitor population in both protocols.

Note 2: Optimizing parameters: merge clusters

By default, the Totem clustering function `RunClustering()` uses a target range of clusters from 3 to 20 (`k.range = 3:20`). However, this range may not be optimal for each data set, particularly if the data set involves unbalanced cell clusters. In such cases, it is recommended to provide a higher target range of clusters that can be merged later. More information can be found at https://github.com/elolab/Totem-benchmarking/blob/main/Totem.html#merging-clusters.

Note 3: Exporting and accessing the *Totem* results

Totem saves its results as a list under the metadata slot named `totem`. These can be accessed typing the following:

```
metadata(sce)$totem
```

The `sce` object with all the results stored on it can be exported using the function `saveRDS`:

```
saveRDS(object = sce, file = "../results/sce_guickstart.rds")
```

Note 4: Supporting the decision of trajectory selection

Cell connectivity can help support the decision about which topology to select among the top ranked candidates. It is recommended that the user cross-checks the top inferred topologies with the cell connectivity values to ensure that they match the expectations, that is, if the highest and lowest connectivity values correspond to branching points and leaf/end points, respectively, in the inferred trajectory.

Note 5: Optimizing parameters: root

Totem randomly assigns a root to the trajectory during the inference, and the user should specify the root milestone or cell centroid afterward for a given topology to obtain a proper directed trajectory and pseudotime.

Note 6: *Totem*'s limitations

The aim of *Totem* is not necessarily to find the correct topology at first, but rather to identify a good set of candidate topologies that the user can evaluate based on their expectations and cell connectivity, as demonstrated in **GuidedStart**. It is important to note that *Totem* cannot infer topologies that include cycles, disconnected, converging or diverging trajectories and, thus, any attempt to do so may result in incorrect trajectory predictions.

Note 7: Downstream analyses: converting *Totem*'s trajectory to `dynwrap` and `PseudotimeOrdering` objects for differential gene expression

The trajectory of interest obtained with *Totem* can be converted into a `dynwrap` (using the function `ReturnDynwrapObject()`) or `PseudotimeOrdering` (with `ReturnSlingshotObject()`) object in order to perform trajectory differential gene expression analysis with the third party tools *dyno* [10] and *tradeSeq* [21], respectively:

```
dynwrap.obj <- ReturnDynwrapObject(sce,traj.name=select.traj)
slingshot.obj <- ReturnSlingshotObject(sce,
                                       traj.name=select.traj)
```

The functions take as input the `SingleCellExperiment` object and the name of the trajectory (`traj.name` parameter; `select.traj` variable corresponds to the trajectory `"8.135"` in the **GuidedStart** protocol).

Acknowledgements

The authors thank the Elo lab for useful discussions and support, particularly Mats Perk and Tomi Suomi. AGGS has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No.: 955321. This research was also supported by University of Turku, Åbo Akademi University, Turku Graduate School (UTUGS).

References

1. Pellin D, Loperfido M, Baricordi C et al (2019) A comprehensive single cell transcriptional landscape of human hematopoietic progenitors. Nat Commun 10(1):2395
2. Qiu C, Cao J, Martin BK et al (2022) Systematic reconstruction of cellular trajectories across mouse embryogenesis. Nat Genet 54(3):328–341
3. Dalton S (2015) Linking the cell cycle to cell fate decisions. Trends Cell Biol 25(10):592–600
4. Kumar S, Fonseca VR, Ribeiro F et al (2021) Developmental bifurcation of human T follicular regulatory cells. Sci Immunol 6(59):eabd8411
5. Ding J, Sharon N, Bar-Joseph Z (2022) Temporal modelling using single-cell transcriptomics. Nat Rev Genet 23(6):355–368

6. Cannoodt R, Saelens W, Saeys Y (2016) Computational methods for trajectory inference from single-cell transcriptomics. Eur J Immunol 46(11):2496–2506
7. La Manno G, Soldatov R, Zeisel A et al (2018) RNA velocity of single cells. Nature 560(7719):494–498
8. Van der Maaten L, Hinton G (2008). Visualizing data using t-SNE. J Mach Learn Res 9(11)
9. McInnes L, Healy J, Melville J (2018) Umap: Uniform manifold approximation and projection for dimension reduction. arXiv preprint arXiv:1802.03426
10. Saelens W, Cannoodt R, Todorov H et al (2019) A comparison of single-cell trajectory inference methods. Nat Biotechnol 37(5):547–554
11. Smolander J, Junttila S, Elo LL (2022) Totem: a user-friendly tool for clustering-based inference of tree-shaped trajectories from single-cell data. bioRxiv doi:10.1101/2022.09.19.508535
12. Maechler Martin (2019) Finding groups in data: Cluster analysis extended Rousseeuw et al. R package version 2(0):242–248
13. Paradis E, Schliep K (2019) ape 5.0: an environment for modern phylogenetics and evolutionary analyses in R. Bioinformatics 35(3):526–528
14. Street K, Risso D, Fletcher RB et al (2018) Slingshot: cell lineage and pseudotime inference for single-cell transcriptomics. BMC Genom 19:1–16
15. Setty M, Kiseliovas V, Levine J et al (2019) Characterization of cell fate probabilities in single-cell data with Palantir. Nat Biotechnol 37(4):451–460
16. Zappia L, Lun A (2022) zellkonverter: Conversion Between scRNA-seq Objects. R package version 1.8.0 https://github.com/theislab/zellkonverter. Accessed 06 April 2023
17. Stuart T, Butler A, Hoffman P et al (2019) Comprehensive integration of single-cell data. Cell 177(7):1888–1902
18. Hie B, Bryson B, Berger B (2019) Efficient integration of heterogeneous single-cell transcriptomes using Scanorama. Nat Biotechnol 37(6):685–691

19. Coifman RR, Lafon S, Lee AB et al (2005) Geometric diffusions as a tool for harmonic analysis and structure definition of data: Diffusion maps. Proc Natl Acad Sci USA 102(21):7426-7431
20. Sun S, Zhu J, Ma Y et al (2019) Accuracy, robustness and scalability of dimensionality reduction methods for single-cell RNA-seq analysis. Genome Biol 20(1):1–21
21. Van den Berge K, Roux de Bézieux H, Street K et al (2020) Trajectory-based differential expression analysis for single-cell sequencing data. Nat Commun 11(1):1201

Figures

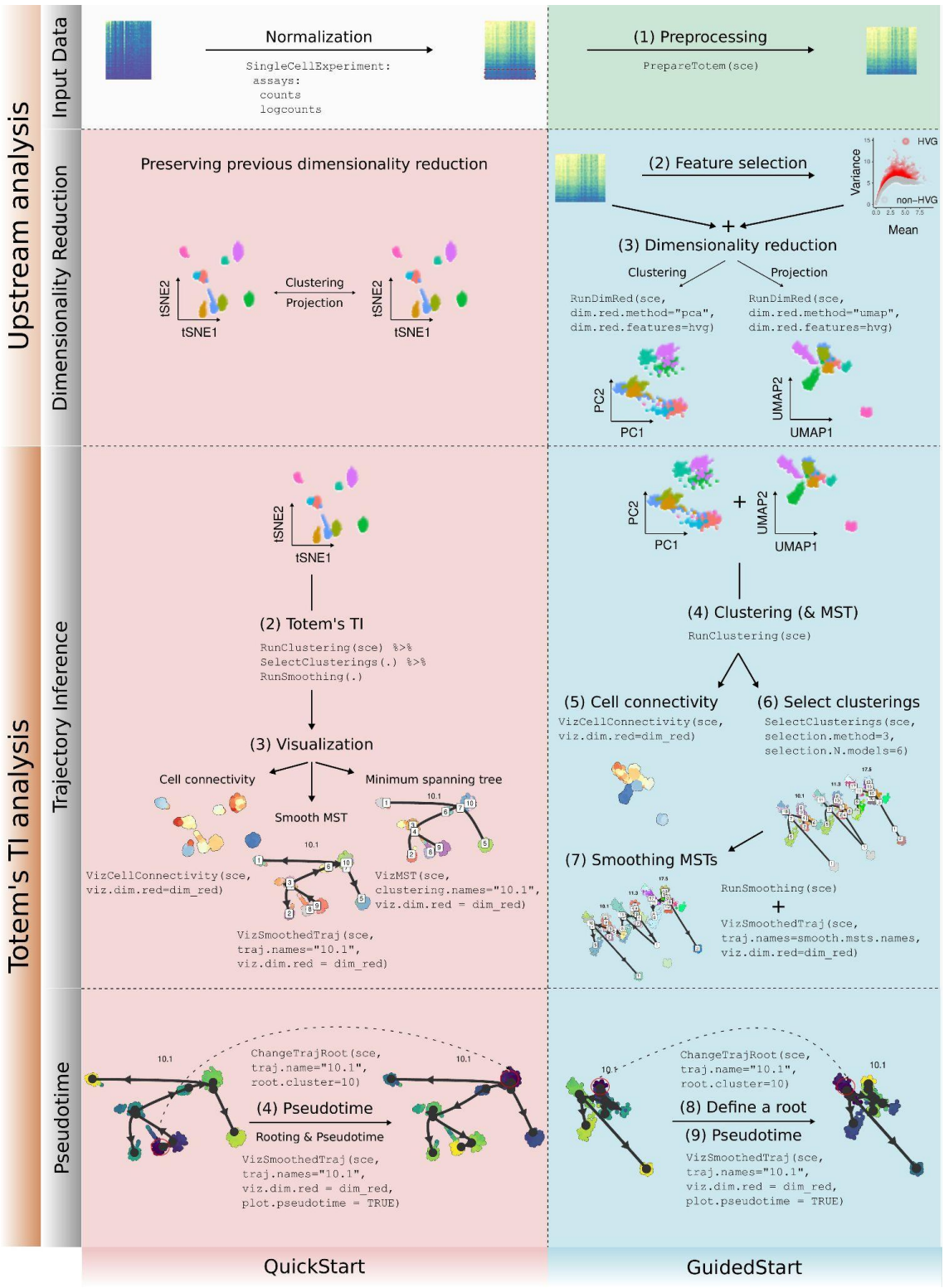

Figure 1. Summary of the **QuickStart** and **GuidedStart** data analysis protocols for the inference of cell trajectory with Totem. The main protocol steps were numbered and the respective Totem's functions highlighted.

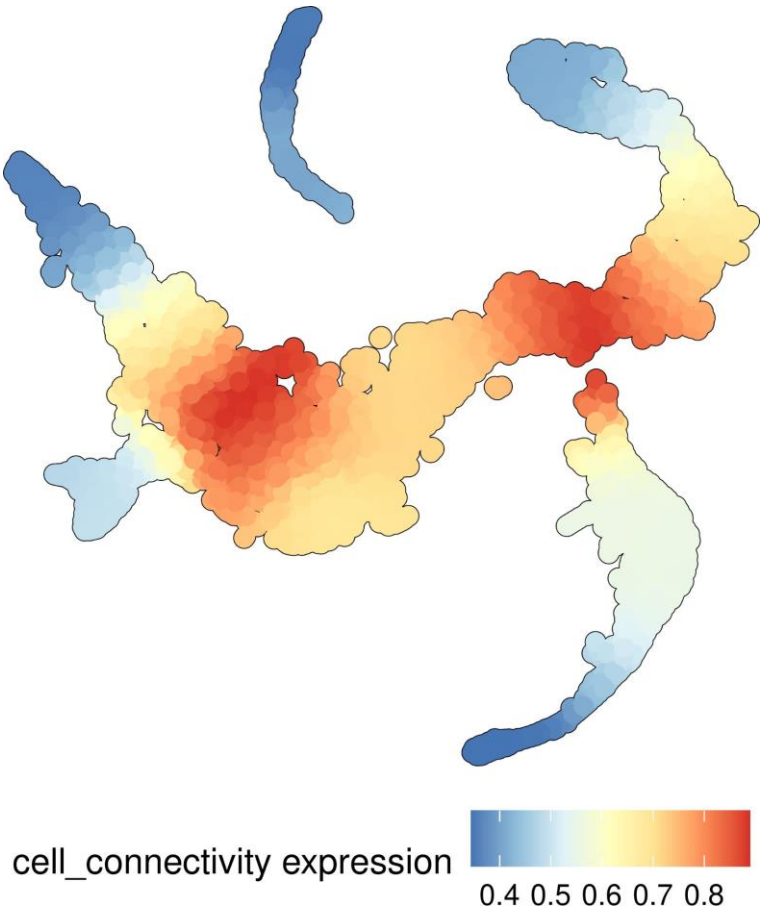

Figure 2. Cell connectivity of the human CD34+ bone marrow cells projected onto t-distributed stochastic neighbor embedding (t-SNE). The cell connectivity highlights the averaged ratio of a cell cluster's connections to the number of clusters calculated across the minimum spanning trees obtained after clustering with `RunClustering()`. Each cell is represented by a dot. Higher (red shade colors) the connectivity farther the distance from the leafs/ending points of the trajectory a cell is and, thus, more likely to represent branching points.

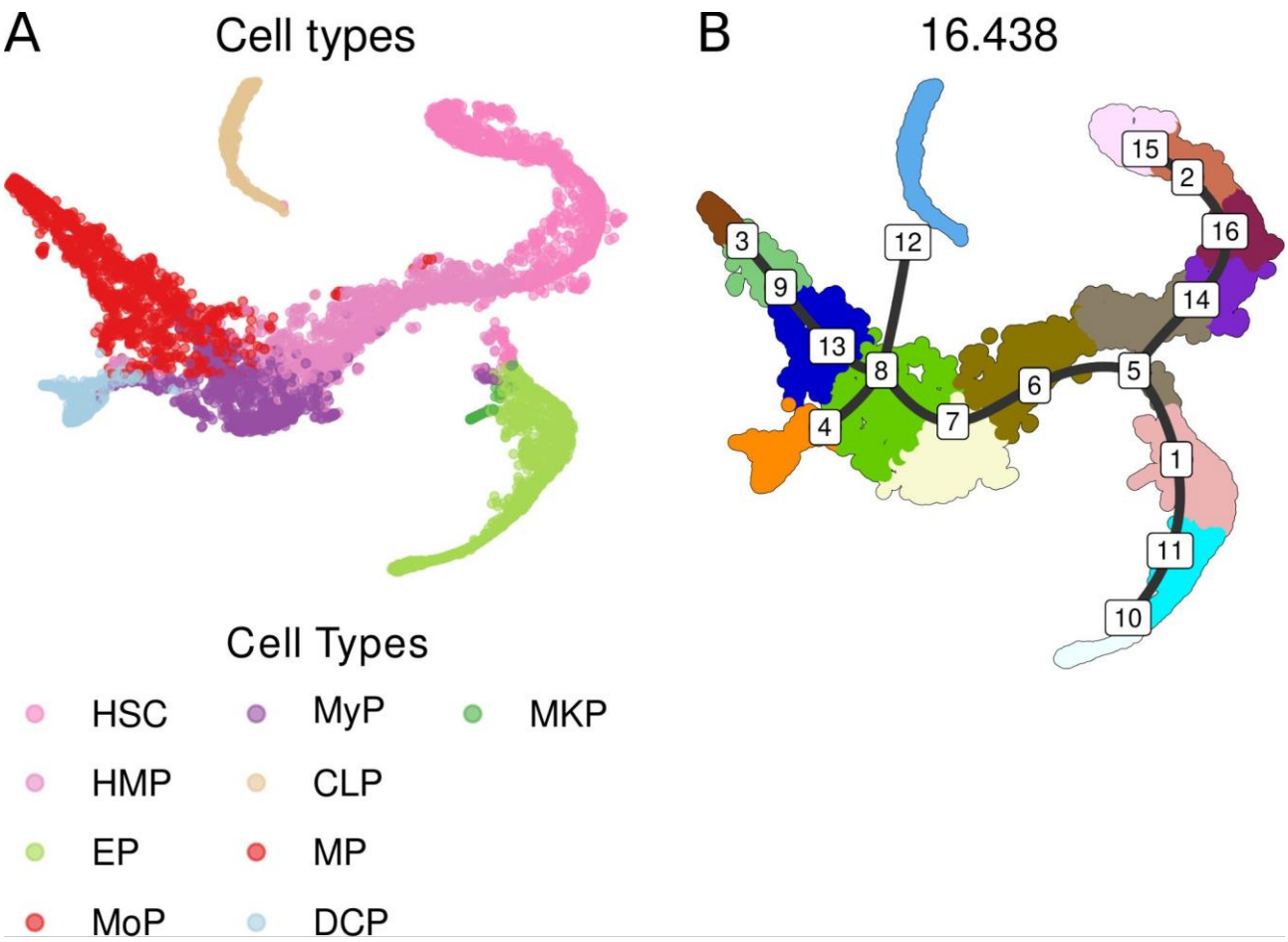

Figure 3. Human CD34+ bone marrow cells projected onto t-distributed stochastic neighbor embedding (t-SNE), illustrating **(A)** the best minimum spanning tree (MST) result from *Totem* and **(B)** the cell types from the original study. HSC (hematopoietic stem cells), HMP (hematopoietic multipotent progenitors), EP (erythroid progenitors), MoP (monocyte progenitors), MyP (myeloid progenitors), CLP (common lymphoid progenitors), DCP (dendritic cell progenitors), MKP (megakaryocyte progenitors).

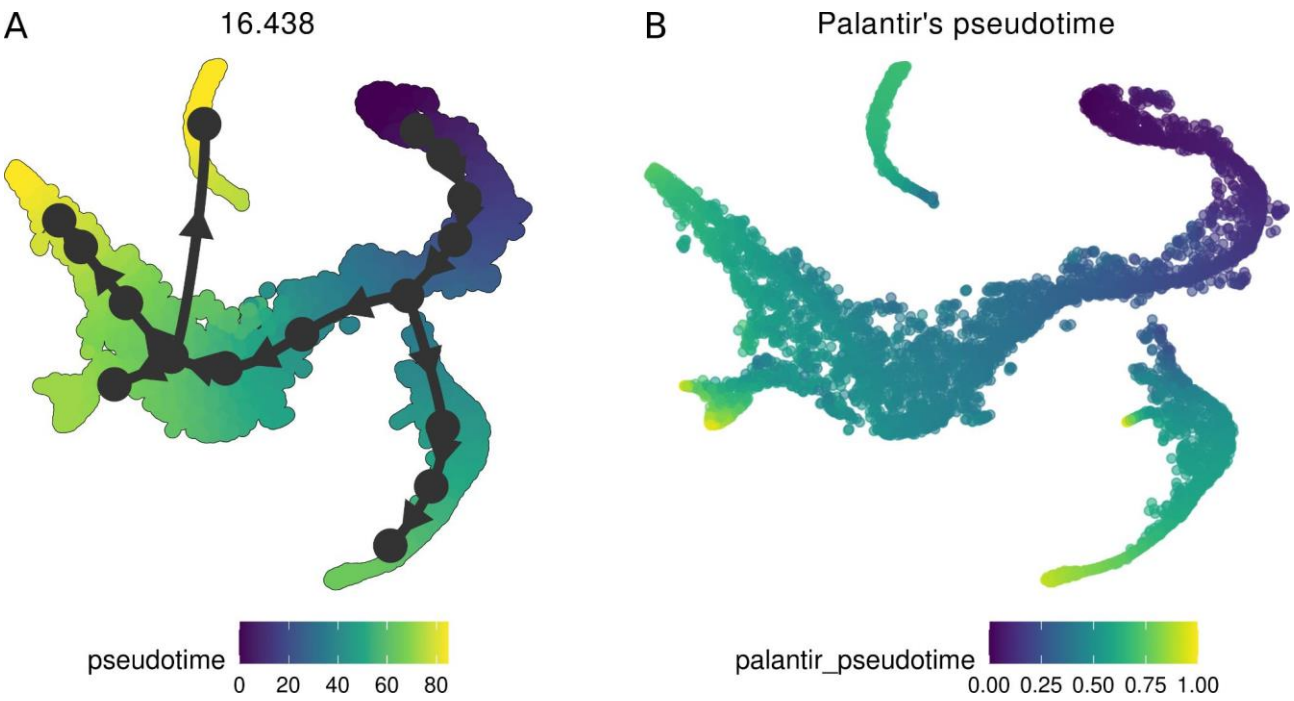

Figure 4. Human CD34+ bone marrow cells projected onto t-SNE highlighted by the pseudotime obtained using **(A)** *Totem* and **(B)** Palantir used in the original study.

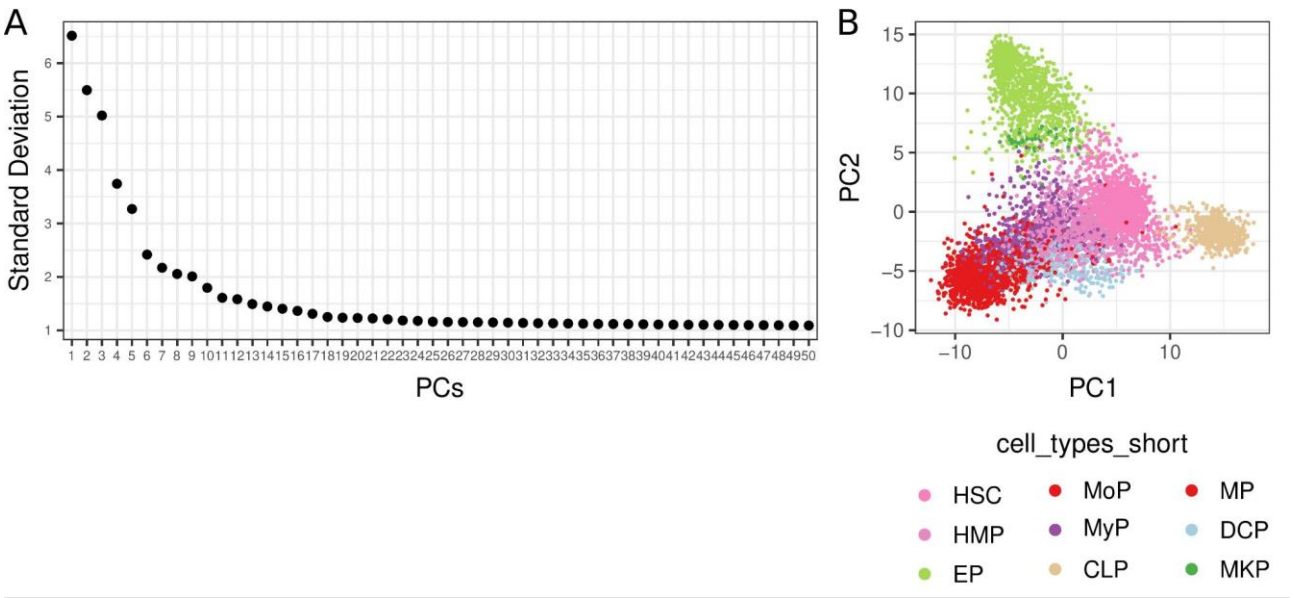

Figure 5. Biological variation of human CD34+ bone marrow cells explored through principal component analysis (PCA). The elbow plot **(A)** shows the standard deviation of each principal component and can help selecting a suitable number of principal components (PCs) for downstream analysis. The PCA plot **(B)** visualizes the cells projected onto the first two PCs, with each cell coloured according to the cell type from the original study. HSC (hematopoietic stem cells), HMP (hematopoietic multipotent progenitors), EP (erythroid progenitors), MoP (monocyte progenitors), MyP (myeloid progenitors), CLP (common lymphoid progenitors), DCP (dendritic cell progenitors), MKP (megakaryocyte progenitors).

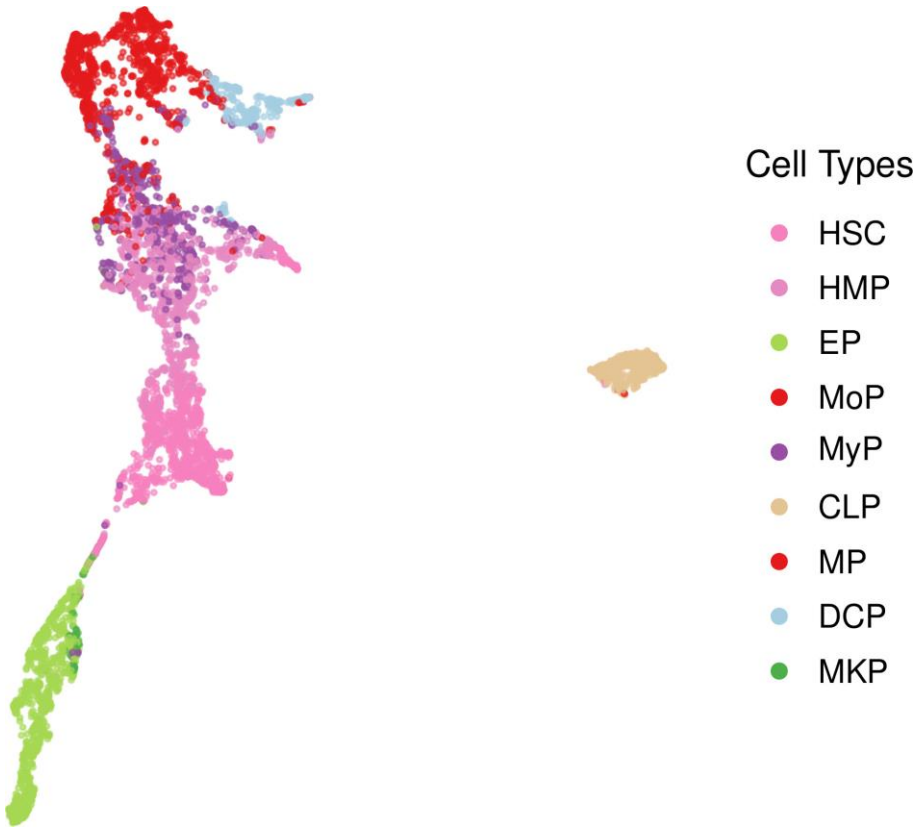

Figure 6. Human CD34+ bone marrow cells projected onto uniform manifold approximation and projection (UMAP) highlighted by cell type. HSC (hematopoietic stem cells), HMP (hematopoietic multipotent progenitors), EP (erythroid progenitors), MoP (monocyte progenitors), MyP (myeloid progenitors), CLP (common lymphoid progenitors), DCP (dendritic cell progenitors), MKP (megakaryocyte progenitors).

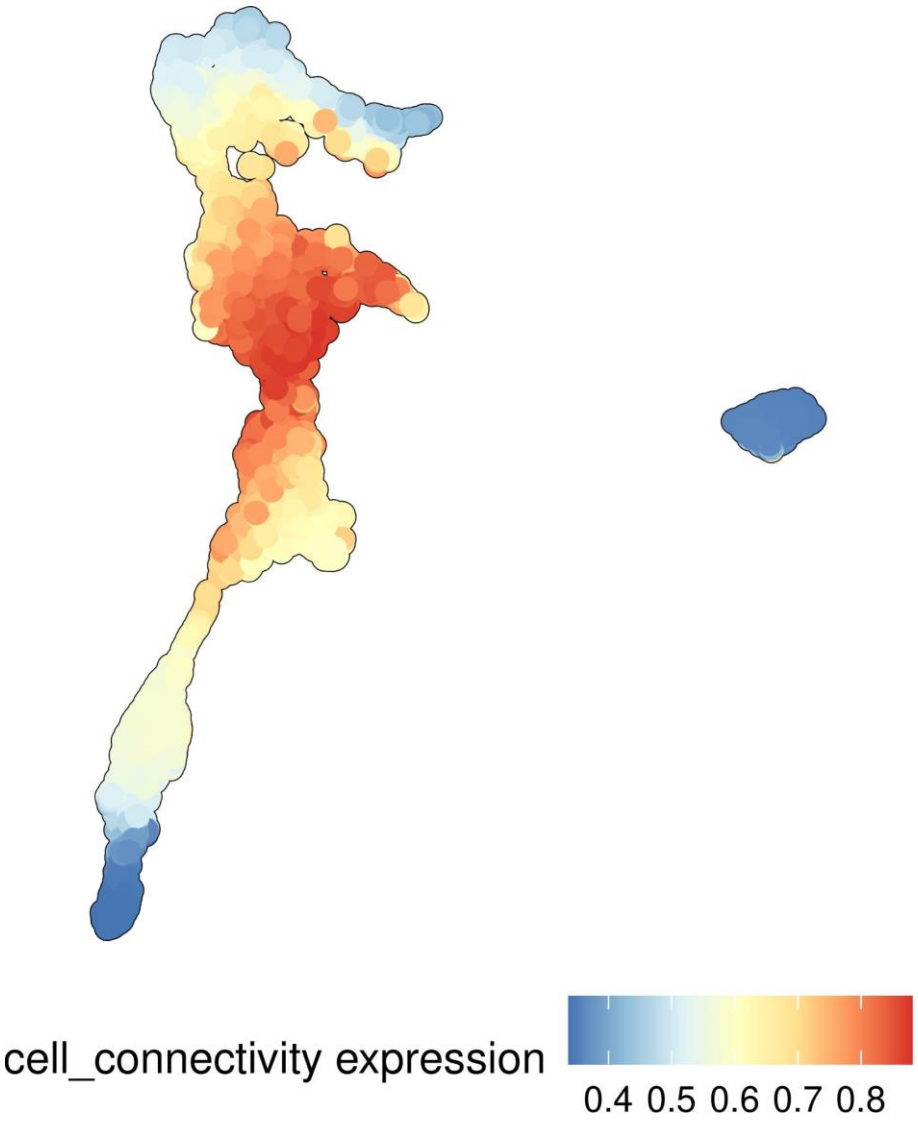

Figure 7. Cell connectivity of the human CD34+ bone marrow cells projected onto uniform manifold approximation and projection (UMAP). The cell connectivity highlights the averaged ratio of a cell cluster's connections to the number of clusters calculated across the minimum spanning trees obtained after clustering with `RunClustering()`. Each cell is represented by a dot. Higher (red shade colors) the connectivity farther the distance from the leafs/ending points of the trajectory a cell is and, thus, more likely to represent branching points.

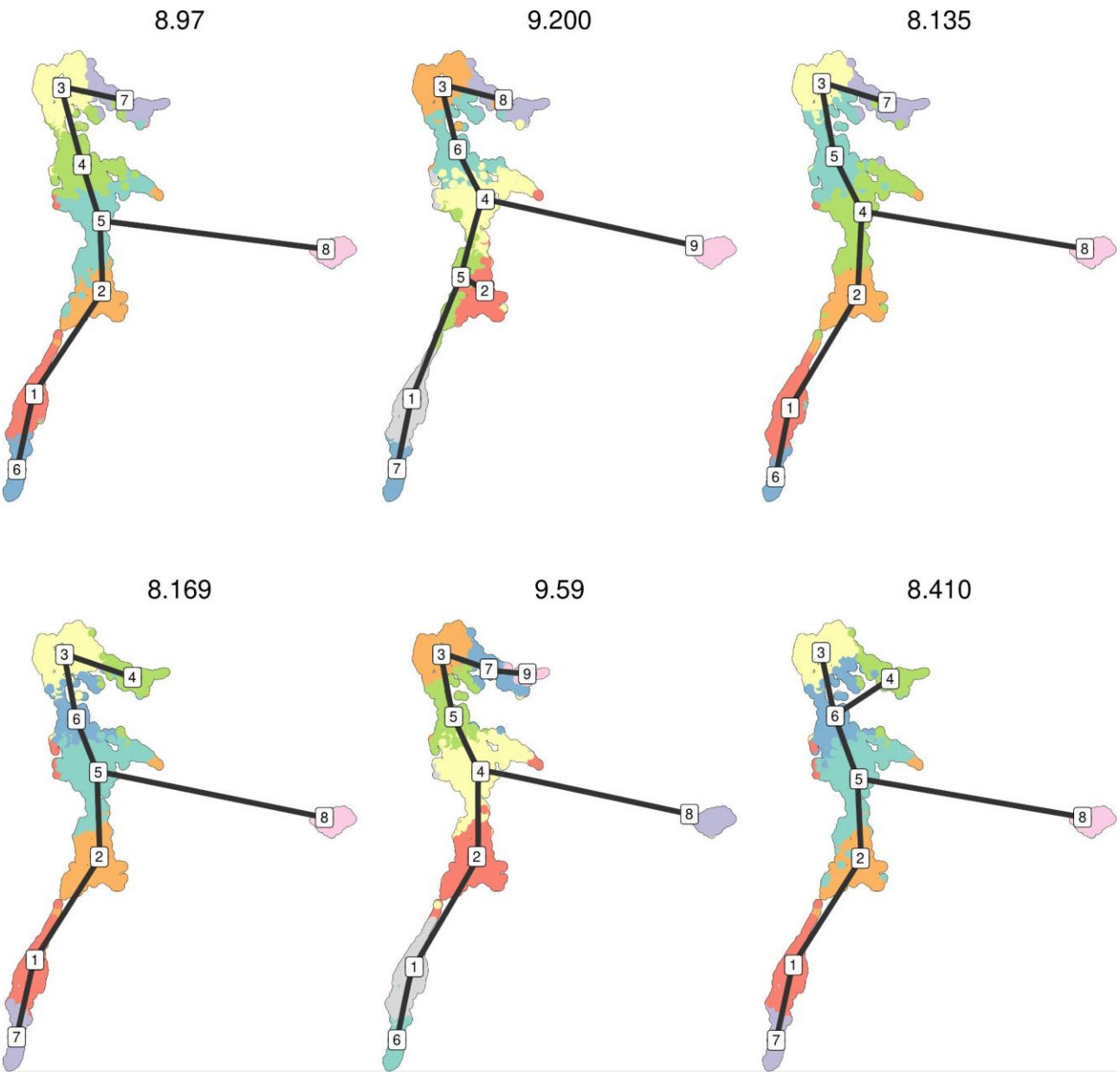

Figure 8. Six top-ranked MST topologies based on the variance ratio criterion and cell connectivity, highlighted across the human CD34+ bone marrow cells projected onto uniform manifold approximation and projection (UMAP).

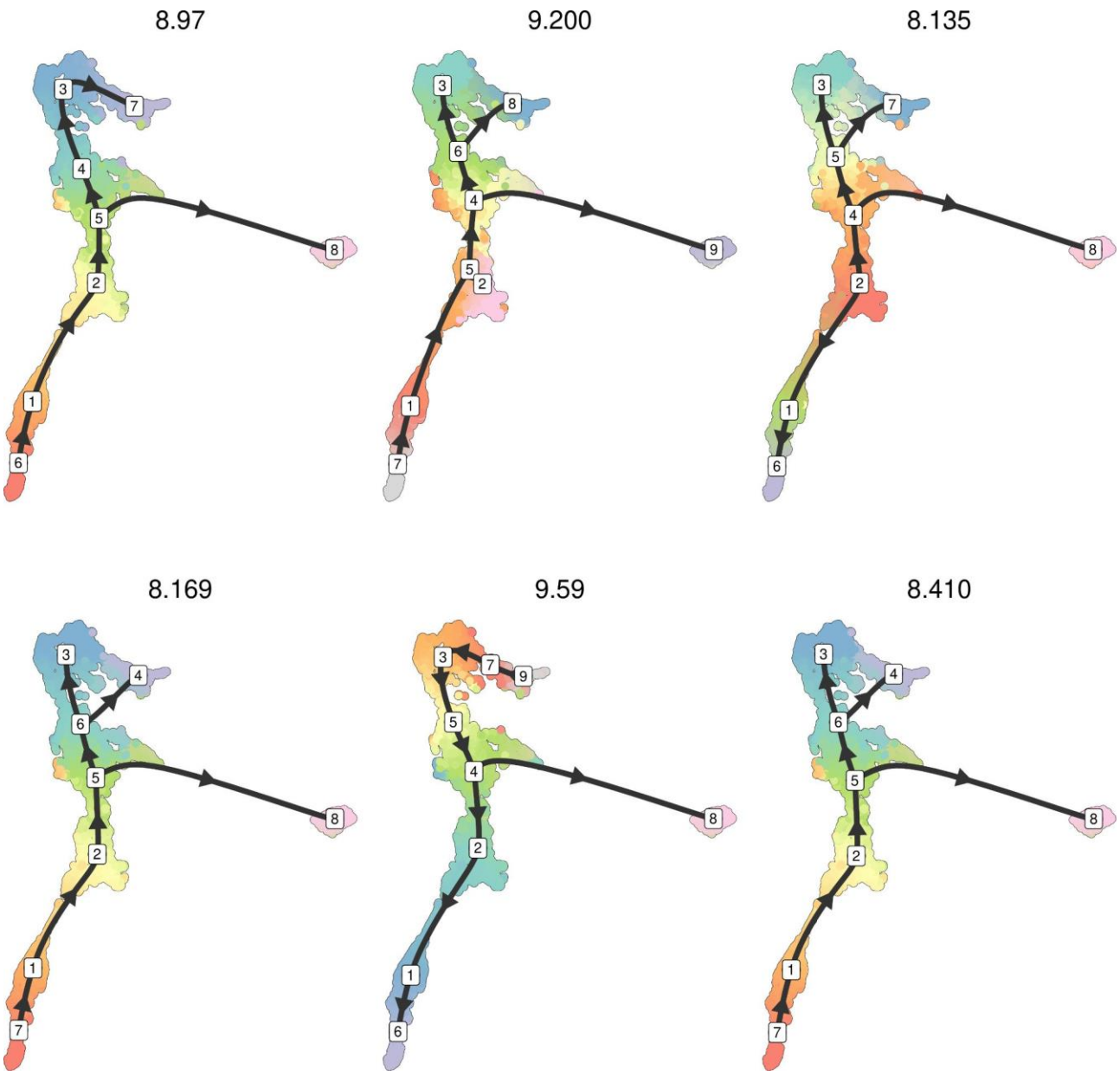

Figure 9. Smoothed trajectories for the six top-ranked MST topologies across the human CD34+ bone marrow cells projected onto uniform manifold approximation and projection (UMAP).

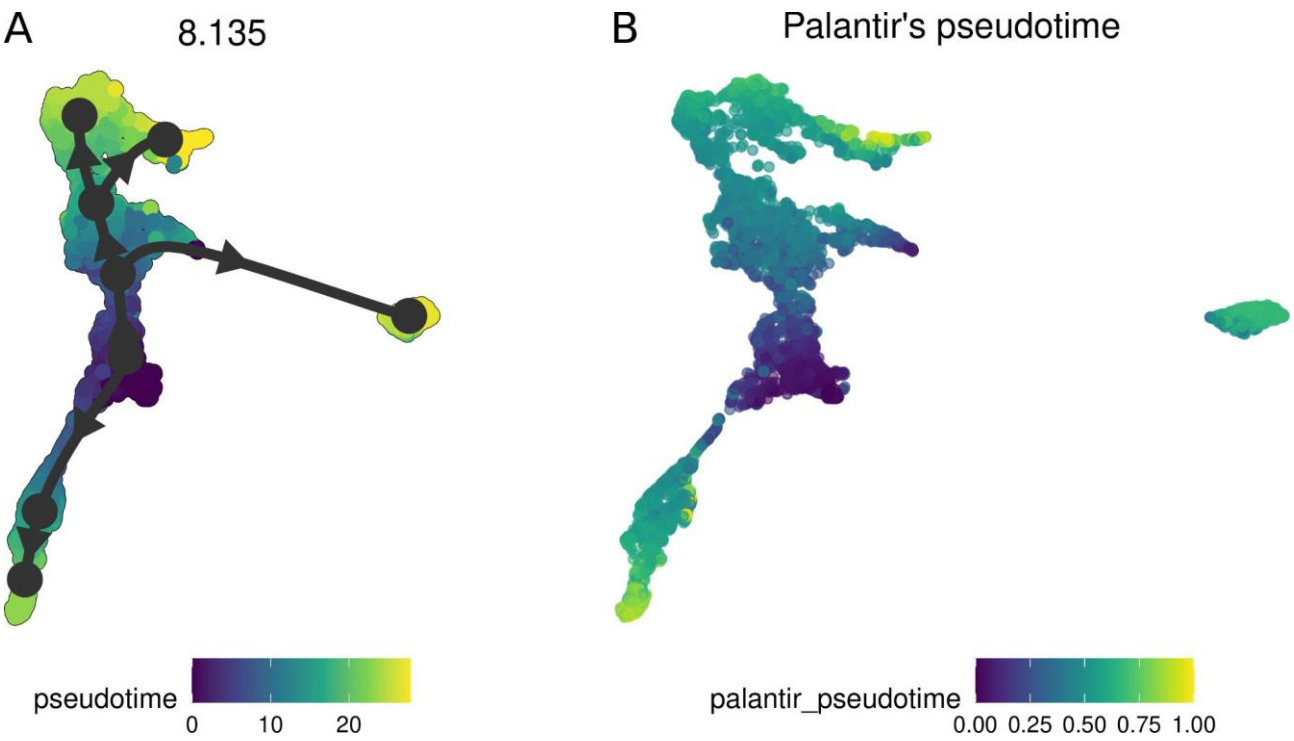

Figure 10. Human CD34+ bone marrow cells projected onto uniform manifold approximation and projection (UMAP) highlighted by the pseudotime obtained using **(A)** *Totem* and **(B)** Palantir used in the original study.