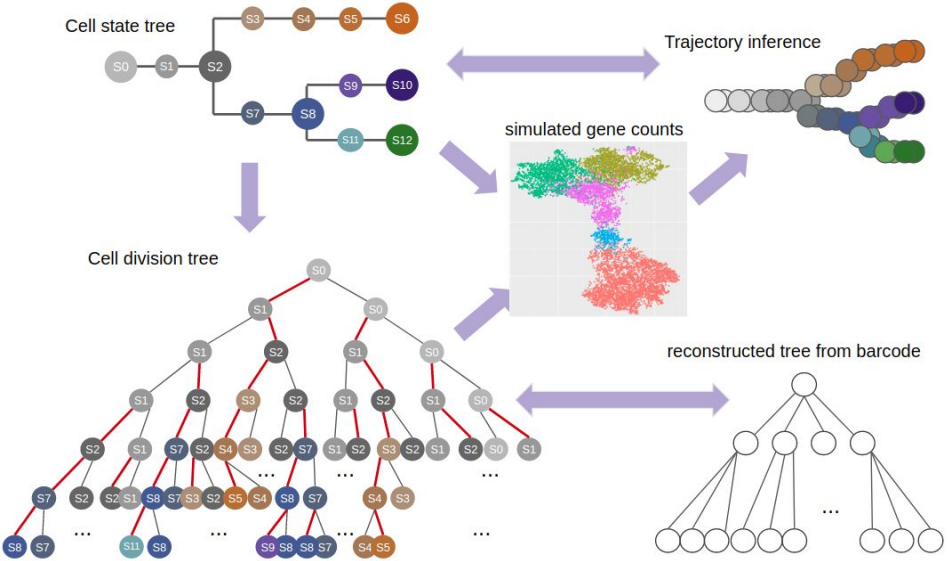
TedSim: Temporal Dynamics Simulation of single-cell RNA sequencing data

Xinhai Pan, Xiuwei Zhang

The Zhang CompBio Lab, School of Computational Science and Engineering, Georgia Institute of Technology

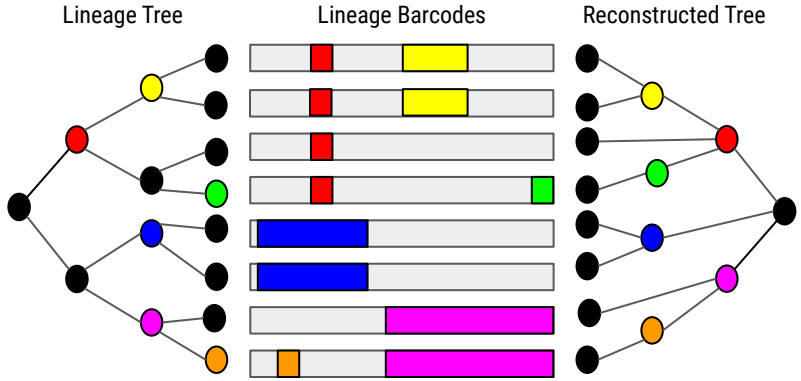
Overview of TedSim: Connecting two aspects of temporal analysis

Recently, the combined single-cell RNA sequencing (scRNA-seq) and CRISPR/Cas9 genome editing technologies have enabled readouts of gene expression data and lineage barcodes simultaneously. Both the lineage tracing and the trajectory inference frameworks aim at deciphering the changes of cells during cell temporal dynamics. We ask the question how these two problems are related, given that they are based on different designs and assumptions. Here, we present TedSim (temporal dynamics simulator), a simulator of simultaneous scRNA-seq and CRISPR/Cas9 genome editing that models the biological process of cell division and differentiation to form populations of cells and cell lineage tree. TedSim simulates the asymmetric cell division events guided by a cell state tree which lead to different cell types.



TedSim simulates CRISPR/Cas9 induced lineage barcodes

With a cell-lineage tree, TedSim is able to simulate the process of accumulation of CRISPR/Cas9 induced scars, which can be used to reconstruct the lineage using various tree reconstruction algorithms.



Benchmarking tree reconstruction algorithms

Using simulated barcode data, we are able to benchmark the accuracy of tree reconstruction algorithms by comparing reconstructed tree to the division tree (ground truth).

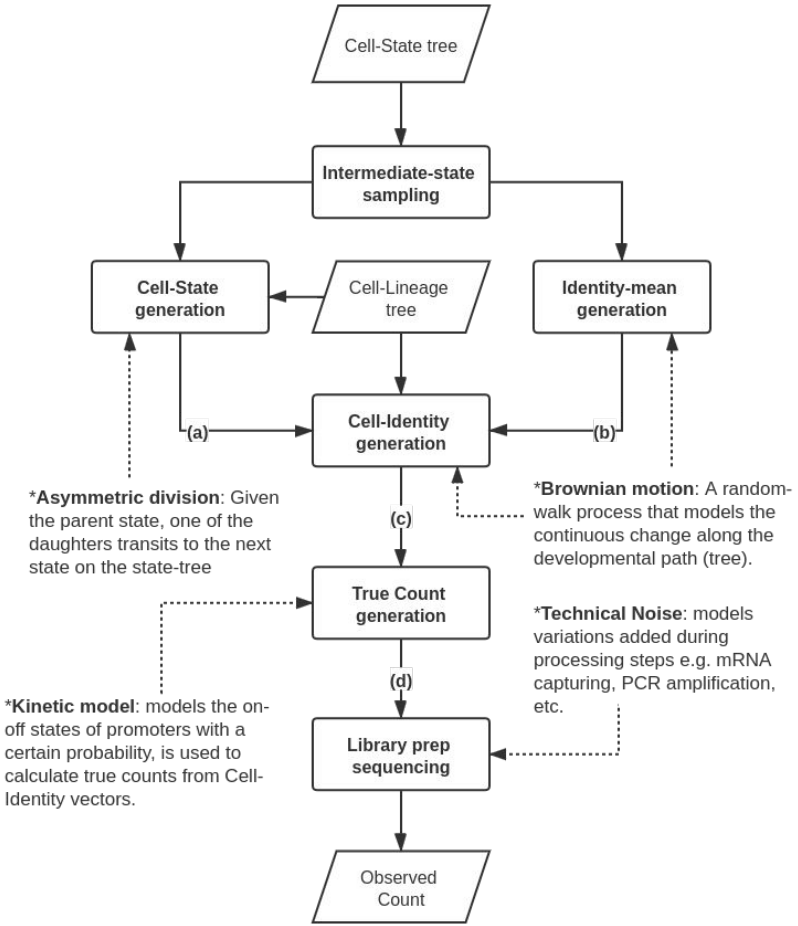
Genetic dropouts can happen when multiple cuts are induced at the same time, which results in missing data in the barcodes. Based on the following results, all selected tree reconstruction algorithms tend to perform much better without dropouts.

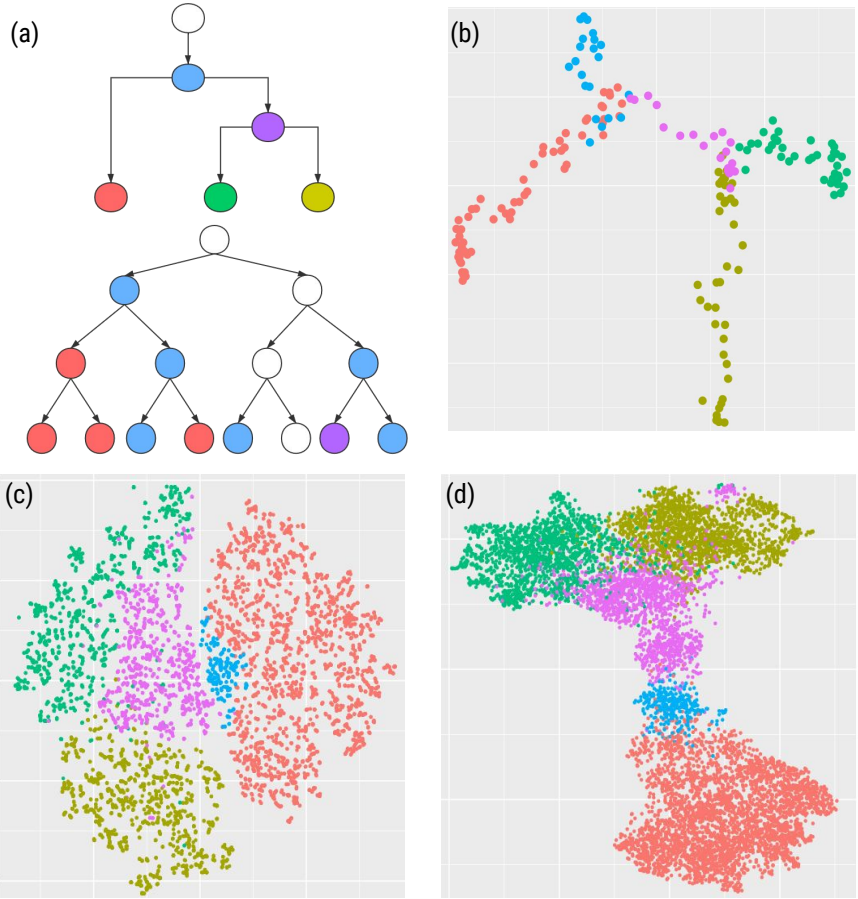
Table 1: The different sets of parameters for the synthetic clusters

	Cassiopeia-Greedy		Cassiopeia-Hybrid		Hamming+FastME		DCLEAR-WH	
	RF dist	Triplets	RF dist	Triplets	RF dist	Triplets	RF dist	Triplets
mutation rate: 0.1, no dropout	**0.1032**	0.8923	0.1573	**0.9588**	0.1058	0.7173	0.2341	0.7154
mutation rate: 0.1, dropout rate: 0.005	0.5608	0.8493	0.5450	**0.8549**	**0.5269**	0.6528	0.6033	0.5885

TedSim models the cell division and differentiation process

Below is the workflow of TedSim generating the gene expression data given the input of cell-state tree and cell-lineage tree.



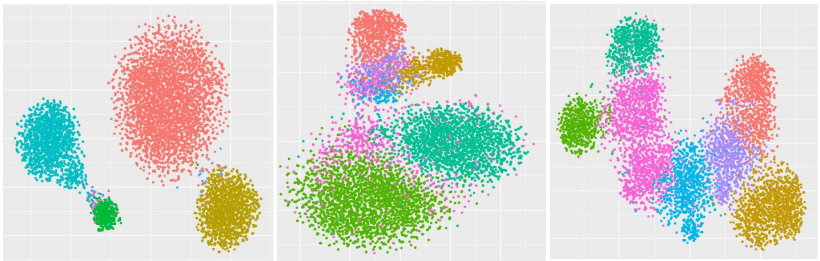


(a), Cell state tree and Cell states on lineage (toy example). (b), Identity-mean values from Brownian motion, visualized by PCA (PC1 vs. PC2). (c), Cell-identity values, visualized by tSNE. (d), True count values, a continuous population visualized by tSNE.

TedSim generates both discrete and continuous populations

TedSim is able to simulate both discrete and continuous populations of cells under the same framework with different parameters.
1. More intermediate states (smaller interval) correspond to continuous changes between cell types
2. Lower asymmetric division rate results in more balanced composition of cell states

$p_a = 1$ $p_a = 0.7$ $p_a = 0.4$

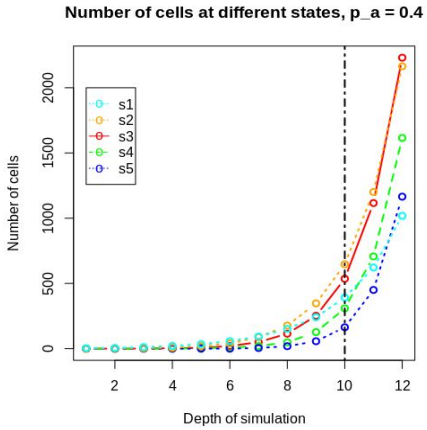


Finding the depth with the best-balanced composition of cell states

Considering a cell division tree of depth d and a single sequential n-state chain $\{s_1, s_2,...,s_n\}$, given asymmetric division rate p_a , the expected number of cells for different states can be derived as following:

$$N_d(s_i) = p_a N_{d-1}(s_{i-1}) + (2 - p_a)N_{d-1}(s_i)$$

For p_a = 0.4, we can simulate and plot the number of cells at different depths. The balance of cell types can be achieved before d=10 while getting as many cells as possible.



Reconstruct cell-state tree from simulated counts

When cells are collected at the right depth, and under certain scenarios (including the density of states sampled on the state tree and the asymmetric division rate), we can obtain continuous populations that reflect the temporal changes of all cell types on the state tree.

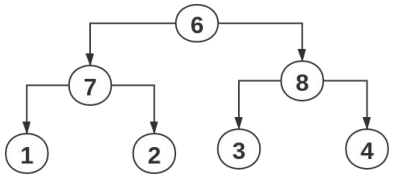
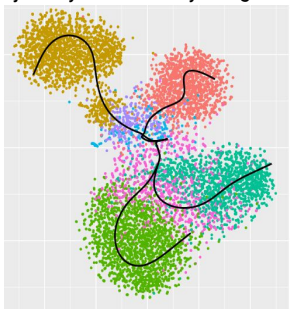


Table 2: Inferred lineages by Slingshot

Lineages	path of connected cell types		
Lineage 1	cell type "6"	cell type "7"	cell type "1"
Lineage 2	cell type "6"	cell type "7"	cell type "2"
Lineage 3	cell type "6"	cell type "8"	cell type "3"
Lineage 4	cell type "6"	cell type "8"	cell type "3"

trajectory inference by Slingshot



Do trajectory inference methods always find the underlying cell state tree?

- When the cells have relatively balanced composition of states, the TI methods can usually find this tree.
- Otherwise the inferred trajectories tend to not reflect the true state tree
- TedSim can help wet labs to better design experiments such that cells are collected at the right time (or combination of multiple time points) to obtain cells with best coverage of all intermediate cell states.

PAGA-tree, $p_a = 0.6$ Slingshot, $p_a = 1$ PAGA-tree, $p_a = 1$