Bioinformatics
doi.10.1093/bioinformatics/xxxxxx
Advance Access Publication Date: Day Month Year
Application Note

single-cell transcriptomics

PROSSTT: probabilistic simulation of single-cell RNA-seq data for complex differentiation processes

Nikolaos Papadopoulos, R. Gonzalo Parra, Johannes Söding

Quantitative and Computational Biology, Max Planck Institute for Biophysical Chemistry, Göttingen, 37077, Germany

Associate Editor: XXXXX

Received on XXXXX; revised on XXXXX; accepted on XXXXX

Abstract

Summary: Cellular lineage trees can be derived from single-cell RNA sequencing snapshots of differentiating cells. Currently, only datasets with simple topologies are available. To test and further develop tools for lineage tree reconstruction, we need test datasets with known complex topologies. PROSSTT can simulate scRNA-seq datasets for differentiation processes with lineage trees of any desired complexity, noise level, noise model, and size. PROSSTT also provides scripts to quantify the quality of predicted lineage trees.
Availability: https://github.com/soedinglab/prosstt
Contact: soeding@mpibpc.mpg.de
Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

Recent advances in single-cell RNA sequencing (scRNA-seq) (Macosko, 2015; Klein *et al.*, 2015) make it possible to generate expression profiles for thousands of cells. Clustering the transcriptomic snapshot of a cell population reveals cell types (Trapnell, 2015), and ordering the cells according to their progress through differentiation reconstructs cellular lineage trees, offering insights into complex processes such as organogenesis (Camp *et al.*, 2017). The change in gene expression along the reconstructed trees gives us unprecedented, time-resolved data to quantitatively investigate the gene regulatory processes underlying cellular development.

As more and more complex processes are investigated, there will be a need to derive lineage trees of topologies more complex than linear or singly-branched ones. Also, with various methods already published (Rostom *et al.*, 2017) and more being developed, the need to quantify method performance is becoming more pressing. With the available data, assessing method performance is challenging as there are no datasets with known ground truth, i.e. data with known intrinsic developmental time and cell identity. These needs can be addressed by simulating realistic scRNA-seq datasets of complex dynamic processes.

Tools like Splatter (Zappia *et al.*, 2017) and dyngen (Saelens *et al.*, 2018) can simulate scRNA-seq data from lineage trees, however both have limitations. In particular, Splatter does not explicitly model coordinated change in gene expression, which results in tree segments that are in truth non-adjacent being placed close to each other. This happens in gene expression space as well as after dimensionality reduction (section 5, Supplementary Material). Additionally, Splatter doesn't provide a global pseudotime for the simulated cells, reducing its usefulness in the context of the evaluation of tree inference methods. Dyngen is built around a gene regulatory network that gives rise to a certain network topology. This requires users to design the regulatory network or use one of the pre-generated modules, which limits the complexity of the topologies that can be simulated.

Here we present PROSSTT (PRObabilistic Simulation of Single-cell RNA-seq Tree-like Topologies), a python package for simulating UMI counts from scRNA-seq experiements of complex differentiation pathways.

2 Model

PROSSTT generates simulated scRNA-seq datasets in four steps:

1. Generate tree: The topology of the lineage tree (number of branches, connectivity) and the length of each branch are read in or, alternatively, sampled. The integer branch lengths give the number of steps of the random walk (see next point) and correspond to the pseudotime duration (Fig. 1A (inset)). The topology can also be linear.

2. Simulate average gene expression along tree: Gene expression levels are linear mixtures of a small number K (default: scales with number of bifurcations) functional expression programs w_k. For each tree segment, we simulate the time evolution of expression programs by random walks with momentum term (see Fig. 1A and Supplementary Material). The mean

© The Author(s) 2019. Published by Oxford University Press.
This is an Open Access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/4.0/), which permits unrestricted reuse, distribution, and reproduction in any medium, provided the original work is properly cited.

Downloaded from https://academic.oup.com/bioinformatics/advance-article-abstract/doi/10.1093/bioinformatics/btz078/5305637 by MPI Biophysical Chemistry user on 07 February 2019

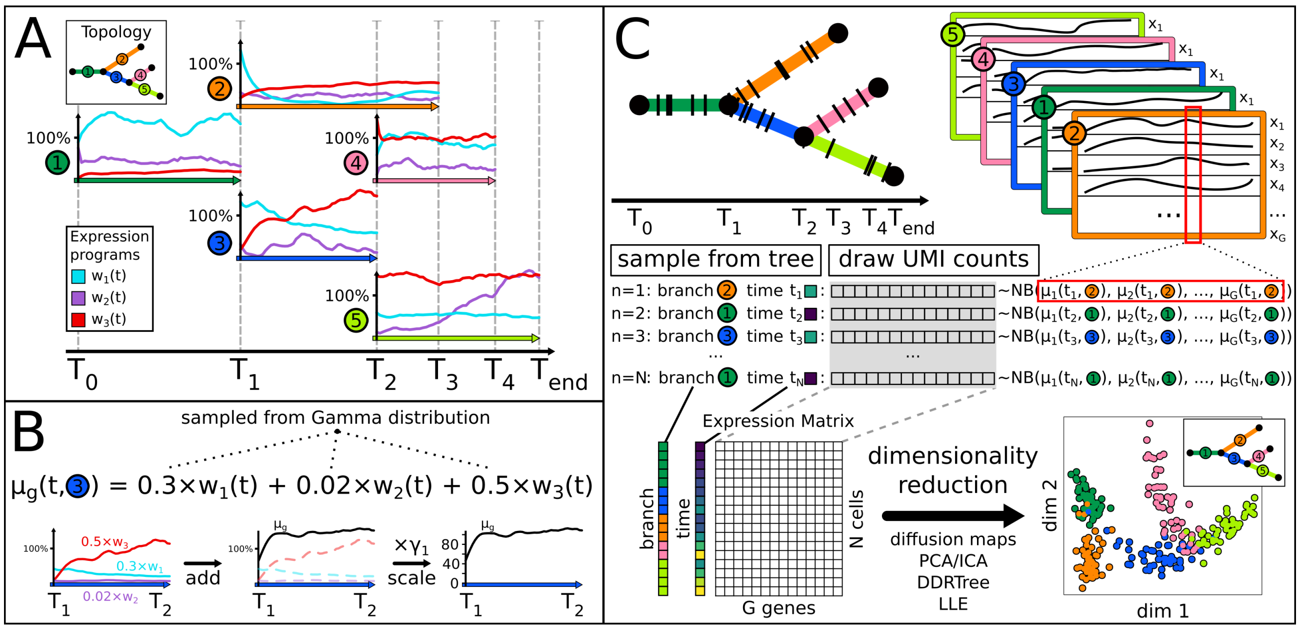
A
Topology
100%
Expression programs
w$_1$(t)
w$_2$(t)
w$_3$(t)
T$_0$ T$_1$ T$_2$ T$_3$ T$_4$ T$_{end}$
B
sampled from Gamma distribution
μ$_g$(t,3) = 0.3×w$_1$(t) + 0.02×w$_2$(t) + 0.5×w$_3$(t)
0.5×w$_3$
0.3×w$_1$
0.02×w$_2$
add
×γ$_1$
scale
μ$_g$
80
40
0
C
sample from tree
draw UMI counts
n=1: branch 2 time t$_1$: ~NB(μ$_1$(t$_1$,2), μ$_2$(t$_1$,2), ..., μ$_G$(t$_1$,2))
n=2: branch 1 time t$_2$: ~NB(μ$_1$(t$_2$,1), μ$_2$(t$_2$,1), ..., μ$_G$(t$_2$,1))
n=3: branch 3 time t$_3$: ~NB(μ$_1$(t$_3$,3), μ$_2$(t$_3$,3), ..., μ$_G$(t$_3$,3))
n=N: branch 1 time t$_N$: ~NB(μ$_1$(t$_N$,1), μ$_2$(t$_N$,1), ..., μ$_G$(t$_N$,1))
x$_1$ x$_2$ x$_3$ x$_4$ ... x$_G$
Expression Matrix
branch
time
N cells
G genes
dimensionality reduction
diffusion maps
PCA/ICA
DDRTree
LLE
dim 1
dim 2

Fig. 1. PROSSTT models the single-cell RNA-seq transcriptomes of cells differentiating along a (user supplied or sampled) lineage tree. (A) A small number of gene expression programs is simulated by random walk along each of the tree branches (number of steps = integer branch length). Here, a double bifurcation is regulated by thee expression programs . (B) Relative expected gene expression $\mu_g(t,b)$ is computed as weighted sum of the expression programs with randomly sampled weights (here: gene g in branch 3). Expected expression values are obtained by multiplying with a gene-dependent sampled scaling factor. (C) Cells are sampled from the tree as pairs of pseudotime t and branch b. For each pair, the corresponding average gene expression is retrieved and UMI counts sampled using a negative binomial distribution. Low-dimensional representations of the resulting gene expression matrix are similar to those of real data (section 1, Supplementary Material) and capture the lineage tree topology (diffusion map created with destiny, (Angerer et al., 2016)).

expression of gene g in tree branch b at pseudotime t is a weighted sum of the K different programs k: $\mu_g(t,b) = \sum_{k=1}^{K} w_k(t,b)h_{k,g}$ (Fig. 1B). The weights $h_{k,g}$ are drawn from a gamma distribution (Supplementary Material, section 2.2).

3. Sample cells from tree: We offer multiple ways of sampling cells from a lineage tree: (1) sampling cells homogeneously along the tree, (2) sampling centered diffusely around selected tree points, (3) sampling with user-supplied density, and (4) specifying the velocity with which the process progresses and sampling the resulting density. (Fig. 1C left, Supplementary Material, section 2.3).

4. Simulate UMI counts: We simulate unique molecular identifier (UMI) counts using a negative binomial distribution. First, a scaling factor s_n for the library size is drawn randomly for each cell n (see section 2.4 in Supplementary Material). Following Grün *et al.* (2014) and Harris *et al.* (2017), we make the variance σ_g^2 depend on the expected expression $s_n\mu_g$ as $\sigma_{ng}^2 = \alpha_g(s_n\mu_g)^2 + \beta_g(s_n\mu_g)$. If $\mathbf{x}_n(t,b) = (x_1, x_2, ..., x_G)$ is a cell at pseudotime t and branch b, the transcript counts are $x_{ng}(t,b) \sim \mathrm{NegBin}\left(s_n\mu_g(t,b), \sigma_{ng}^2(t,b)\right)$ (Fig. 1C, right). For each of N cells and each of G genes we draw the number of UMIs from the negative binomial, resulting in an $N \times G$ expression matrix, which can serve as input for tree inference algorithms.

Users can specify the topology of the lineage tree (any connected acyclic graph is acceptable), assign branch pseudotime lengths, adjust parameters for the gene expression programs, and control the noise levels in the data. Default parameter values for α_g, β_g, and the base gene expression values were set in the range of parameters of real datasets (Supplementary Material, section 3). If provided with a real dataset, PROSSTT can learn hyperparameters that will generate simulated data with similar summary statistics.

3 Application

We generated 10 sets of 100 simulations each, for different degrees of topology complexity (from 1 up to 10 bifurcations). In another study, we used this dataset to assess the performance of our tool MERLoT and other methods (Parra *et al.*, 2018). We provide scripts with implementations of appropriate quality measures as well as the pipeline to generate the simulations and evaluate predictions by state-of-the-art software.

PROSSTT is capable of producing simulations with the summary statistics of true datasets, and can reproduce data faithfully in cases where the underlying lineage tree is available.

4 Conclusions

PROSSTT simulates scRNA-seq data for complex differentiation processes. Low-dimensional visualizations produced by tree reconstruction tools resemble those of real datasets. Increasingly complex datasets with uncertain biological ground truth are becoming available. PROSSTT can help the development of methods that can reconstruct such complex trees by facilitating their quantitative assessment. Furthermore, the modular nature of the software allows for easy extensions, for example PROSSTT could serve to test the influence of noise models and give biological insights into how to model and interpret scRNA-seq data.

Funding

RGP is a long term EMBO postdoctoral fellow (ALTF 212-2016).
Conflicts of interest: None declared.

References

Angerer, P. *et al.* (2016). destiny : diffusion maps for large-scale single- cell data in R. *Bioinformatics*, **32**(December 2015), 1241–1243.

Camp, J. G. *et al.* (2017). Multilineage communication regulates human liver bud development from pluripotency. *Nature Publishing Group*.

Grün, D. *et al.* (2014). Validation of noise models for single-cell transcriptomics. *Nature methods*, **11**(6), 637–640.

Downloaded from https://academic.oup.com/bioinformatics/advance-article-abstract/doi/10.1093/bioinformatics/btz078/5305637 by MPI Biophysical Chemistry user on 07 February 2019

Harris, K. D., Bengtsson Gonzales, C., Hochgerner, H., Skene, N. G., Magno, L., Katona, L., Somogyi, P., Kessaris, N., Linnarsson, S., and Hjerling-Leffler, J. (2017). Classes and continua of hippocampal CA1 inhibitory neurons revealed by single-cell transcriptomics. *bioRxiv*.

Klein, A. M. *et al.* (2015). Droplet barcoding for single-cell transcriptomics applied to embryonic stem cells. *Cell*, **161**(5), 1187–1201.

Macosko, E. Z. o. (2015). Highly parallel genome-wide expression profiling of individual cells using nanoliter droplets. *Cell*, **161**(5), 1202–1214.

Parra, R. G., Papadopoulos, N., Ahumada-Arranz, L., El Kholtei, J., Treutlein, B., and Soeding, J. (2018). Reconstructing complex lineage trees from scRNA-seq data using MERLoT. *bioRxiv*.

Rostom, R. *et al.* (2017). Computational approaches for interpreting scRNA-seq data.

Saelens, W., Cannoodt, R., Todorov, H., and Saeys, Y. (2018). A comparison of single-cell trajectory inference methods: towards more accurate and robust tools. *bioRxiv*.

Trapnell, C. (2015). Defining cell types and states with single-cell genomics. *Genome Research*, **25**(10), 1491–1498.

Zappia, L., Phipson, B., and Oshlack, A. (2017). Splatter: simulation of single-cell RNA sequencing data. *Genome Biology*, **18**(1), 174.

Downloaded from https://academic.oup.com/bioinformatics/advance-article-abstract/doi/10.1093/bioinformatics/btz078/5305637 by MPI Biophysical Chemistry user on 07 February 2019