dyngen: simulating developing single cells

The *dynverse* guys

September 5, 2019

Abstract:

1 Introduction

Continuous technological advancements to high-throughput profiling of single cells are having profound effects on how researchers can validate biological hypotheses. For example, single-cell RNA sequencing (scRNA-seq) directly resulted in the development of a new type of computational method called trajectory inference (TI). By profiling the transcriptomics profiles of developing cells, TI methods attempt to reconstruct and characterise the underlying dynamic processes [1]. While early experimental technologies allowed to profile one single modality (e.g. DNA sequence, RNA or protein expression), recent developments permit profiling multiple modalities simultaneously.

An ideal experiment would be able to observe all aspects of a cell, including a full history of its molecular states, spatial positions and environmental interactions [2]. While this falls outside the reach of current experimental technologies, *in silico* simulations of single cells would allow developing the next wave of computational techniques in anticipation of new experimental technologies.

A few simulators of scRNA-seq profiles have already been developed (e.g. splatter [3], PROSSTT [4] and SymSim [5]). These can be used to evaluate the performance of computational tools, and to explore their strengths and weaknesses. A limitation of directly simulating a scRNA-seq profile (instead of a single cell) is that extending the simulation to other aspects of the cell – such as tracking the full history of molecular states – becomes difficult.

We introduce dyngen, a multi-modality simulator of single cells (Figure 1). dyngen was initially developed as part of a comprehensive benchmark of TI methods [6] but has since been extended to be applicable in a much broader context. We demonstrate its flexibility by simulating three different types of biological experiments, and using these simulations to develop new benchmarking techniques for computational tools.

2 Results

dyngen simulates the transcriptomic changes of a cell over time using a model of gene regulation. Throughout this section, a simple simulation of a cell undergoing a cyclic process

is used to illustrate key strengths of dyngen (Figure 2). This example only comprises of a single cell containing 5 genes, but dyngen can easily scale up to thousands of simulations containing thousands of genes.

In dyngen, a cell consists of a set of molecules, the abundance of which are affected by a set of reactions: transcription, splicing, translation, and degradation (Figure 2A). These reactions are determined from a predefined set of gene regulatory interactions (Figure 2B), henceforth referred to as a gene regulatory network (GRN). The likelihood of a reaction occurring at any given point in time is defined by the GRN and by the abundance of molecules involved each reaction.

One of dyngen's main advantages is that through careful engineering of the GRN, different cellular developmental processes can be obtained. Different GRNs can result in branching, converging, cyclic, or even disconnected developmental topologies. Multiple simulations with slightly different GRNs can emulate rewiring events in disease or perturbation experiments. Multiple simulations with different initial molecule abundance levels can be used to replicate batch effects.

Another advantage is that dyngen returns many modalities throughout the whole simulation: molecular abundance, cellular state, number of reaction firings, reaction likelihoods, and regulation activations (Figure 2C–F). These modalities can serve both as input data and ground truth for benchmarking many types of computational approaches. For example, a network inference method could use mRNA abundance and cellular states as inputs, and its output could be benchmarked against the gold standard GRN.

The final main advantage is that by making alterations to the simulation pipeline, multiple types of experiments (sampling technique or profiling technique) can be simulated. By default, dyngen supports snapshot experiments (uniformly sampling from an asynchronous dynamic process) and time-series experiments (sampling cells from different intervals in the simulation). It is possible to implement other experimental protocols (which perhaps do not exist in real life), such as sampling the same cell at regular intervals.

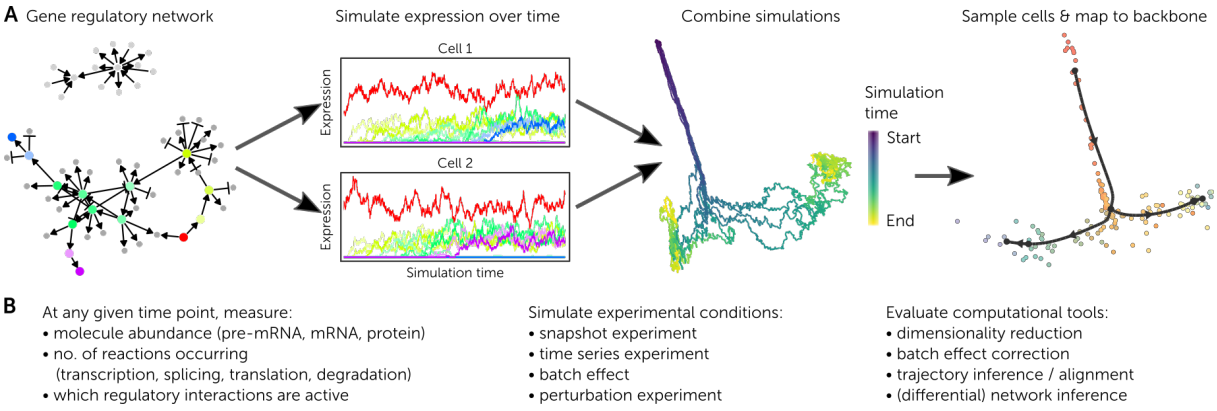

Figure 1: Showcase of dyngen functionality. **TODO: change to pdf.** Remove B? *Yes please*

3 Discussion

As is, dyngen's single cell simulations can be used to evaluate common single-cell omics computational methods such as clustering, batch correction, trajectory inference and network inference. However, the combined effect of these advantages results in a framework that is flexible enough to adapt to a broad range of applications, some of which may still be needed in the future. This may include methods that integrate clustering, network inference and trajectory inference. In this respect, dyngen may promote the development of new tools in the single-cell field similarly as other simulators have done in the past **10.1093/bioinformatics/btx631 10.1093/bioinformatics/btr373 10.1038/nmeth.3407**.

Adding batch effects to snapshot simulations of linear (or even branching) trajectories allows to evaluate trajectory alignment methods – which attempt to map two or more trajectories onto each other. Adding perturbations to the GRN allows to evaluate the performance of differential network inference methods – which predict differential regulatory interactions between two or more groups of profiles. Sampling a cell at a certain time point and once more at a later time point allows to evaluate the performance of RNA velocity approaches – which predict the future state of a cell by looking at differences in pre-mRNA and mRNA abundance levels.

dyngen ultimately also allows anticipating technological developments in single-cell multi-omics. In this way, it is possible to design and evaluate the performance and robustness of new types of computational analyses before experimental data becomes available. Similarly, it could also be used to compare which experimental technique will likely produce the most accurate result. For example, is it possible to infer directionality of regulatory interactions from snapshot experiments only, or are time series or knock down experiments a necessity in order to infer high quality regulatory networks?

Currently, dyngen focuses on simulating cells as standalone entities. Future developments include extending the framework to simulate multiple cells in a virtual environment. Allowing cells to receive and react to environmental and intercellular stimuli would enable simulating essential cellular processes such as cell division and migration.

4 Methods

4.1 Simulating a snapshot experiment with dyngen

4.1.1 Define the module network and the state network

4.1.2 Generate the transcription factor network

4.1.3 Generate targets and housekeeping genes

4.1.4 Convert gene regulatory network to a set of SSA reactions

4.1.5 Simulate backbone

4.1.6 Run SSA simulations

4.1.7 Map SSA simulations to backbone

4.1.8 Simulate snapshot experiment

4.2 Extensions

4.2.1 Predefined backbones

4.2.2 Backbone lego

4.2.3 Time series experiment

4.2.4 Perturbation experiment

4.2.5 Batch effects

4.3 Example use cases

4.3.1 Trajectory alignment

From discussion: Adding batch effects to snapshot simulations of linear (or even branching) trajectories allows to evaluate trajectory alignment methods – which attempt to map two or more trajectories onto each other.

4.3.2 Differential network inference

From discussion: Adding perturbations to the GRN allows to evaluate the performance of differential network inference methods – which predict differential regulatory interactions between two or more groups of profiles.

4.3.3 RNA velocity

From discussion: Sampling a cell at a certain time point and once more at a later time point allows to evaluate the performance of RNA velocity approaches – which predict the future state of a cell by looking at differences in pre-mRNA and mRNA abundance levels.

References

[1] Robrecht Cannoodt, Wouter Saelens, and Yvan Saeys. "Computational Methods for Trajectory Inference from Single-Cell Transcriptomics". en. In: *European Journal of Immunology* 46.11 (Nov. 2016), pp. 2496–2506. ISSN: 1521-4141. DOI: 10.1002/eji.201646347.

[2] Tim Stuart and Rahul Satija. "Integrative Single-Cell Analysis". en. In: *Nature Reviews Genetics* 20.5 (May 2019), pp. 257–272. ISSN: 1471-0064. DOI: 10.1038/s41576-019-0093-7.

[3] Luke Zappia, Belinda Phipson, and Alicia Oshlack. "Splatter: Simulation of Single-Cell {{RNA}} Sequencing Data". In: *Genome Biology* 18 (Sept. 2017), p. 174. ISSN: 1474-760X. DOI: 10.1186/s13059-017-1305-0.

[4] Nikolaos Papadopoulos, Rodrigo Gonzalo Parra, and Johannes Soeding. "{{PROSSTT}}: Probabilistic Simulation of Single-Cell {{RNA}}-Seq Data for Complex Differentiation Processes". In: *bioRxiv* (Jan. 2018), p. 256941. DOI: 10.1101/256941.

[5] Xiuwei Zhang, Chenling Xu, and Nir Yosef. "Simulating Multiple Faceted Variability in Single Cell RNA Sequencing". en. In: *Nature Communications* 10.1 (June 2019), pp. 1–16. ISSN: 2041-1723. DOI: 10.1038/s41467-019-10500-w.

[6] Wouter Saelens et al. "A Comparison of Single-Cell Trajectory Inference Methods". In: *Nature Biotechnology* 37.May (2019). ISSN: 15461696. DOI: 10.1038/s41587-019-0071-9.

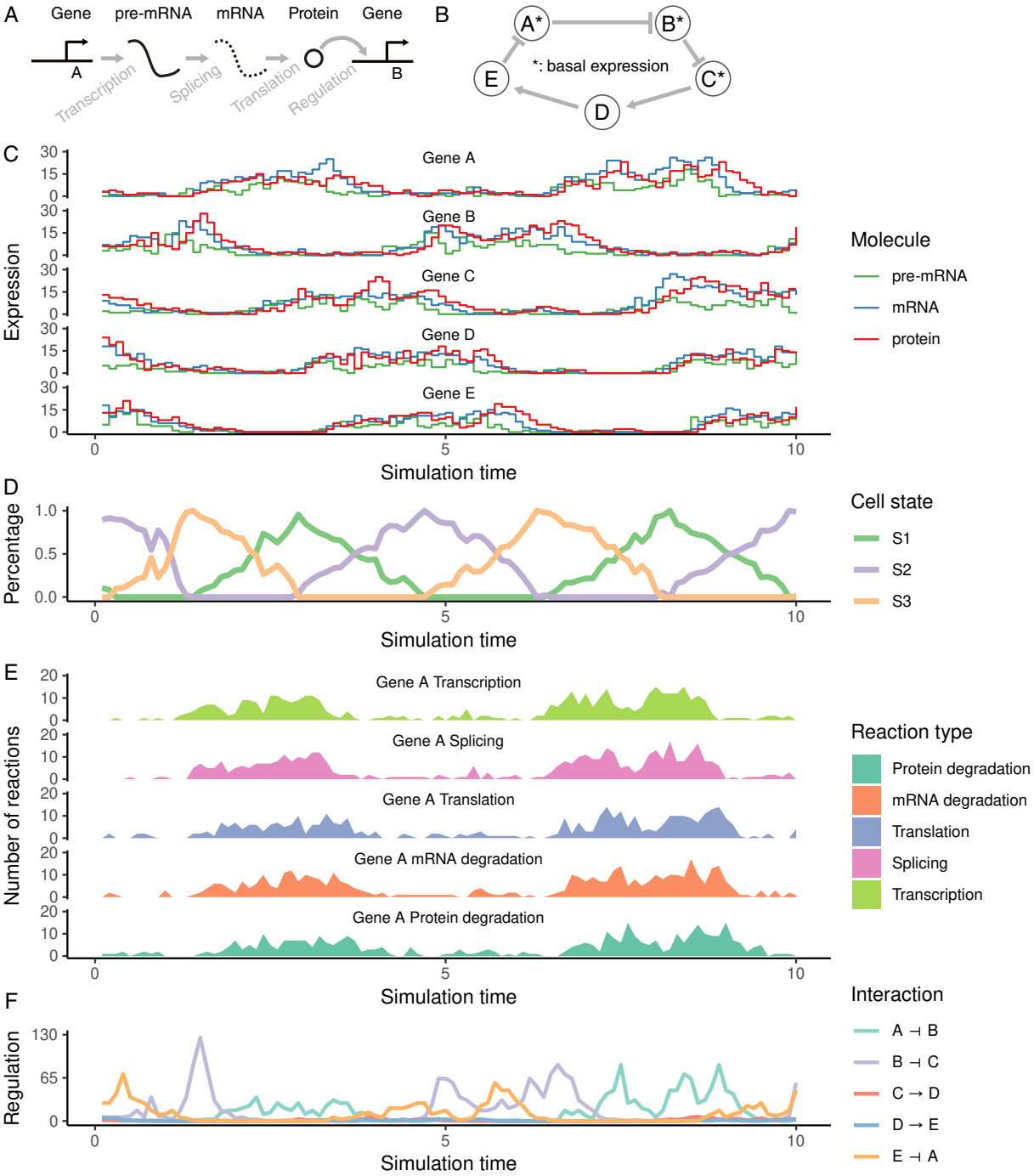

Figure 2: Showcase of dyngen functionality. A time resolution of 0.1 was used, but this can be increased or decreased without effect on performance of the execution of the simulation. **TODO: perhaps it's better to replace Figure 2 with one subfigure for each of the paragraphs in this text.**