

SCBOOLSEQ – scRNA-Seq data binarization and synthetic generation from Boolean dynamics

Gustavo Magaña López^{*1}, Laurence Calzone², Andrei Zinovyev², and Loïc Paulevé¹

^{*}gustavo.magana@labri.fr

¹ Univ. Bordeaux, CNRS, Bordeaux INP, LaBRI, UMR 5800, F-33400 Talence, France

² Institut Curie, INSERM, U900, MINES ParisTech, PSL Research University, CBIO-Centre for Computational Biology, F-75006 Paris, France



poster



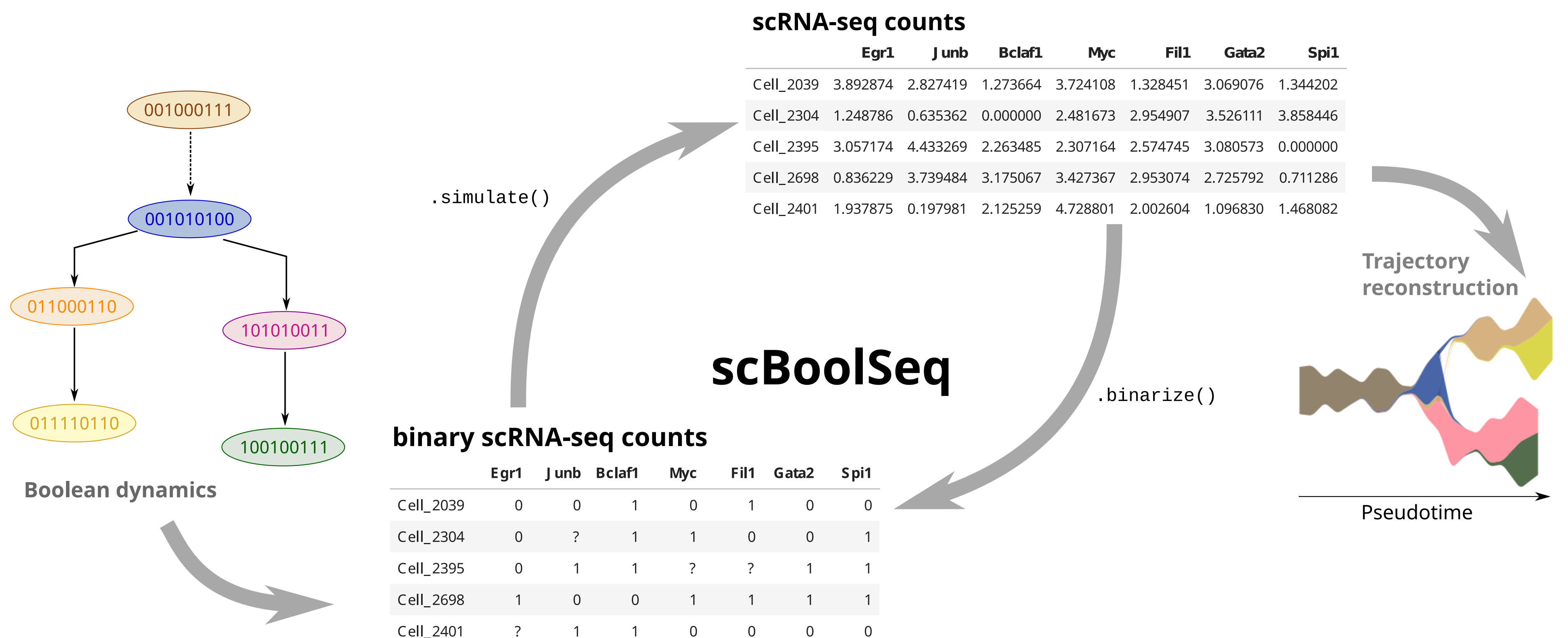
github

LaBRI



Overview

- SCBOOLSEQ is a Python package for **linking scRNA-Seq data and Boolean gene activation states**.
- It uses a reference dataset to:
 - **Binarize experimental data**
 - **Sample synthetic scRNA-Seq** from simulations of Boolean networks.
- Synthetic scRNA-Seq** can serve as a **ground-truth baseline** for validating inference methods.



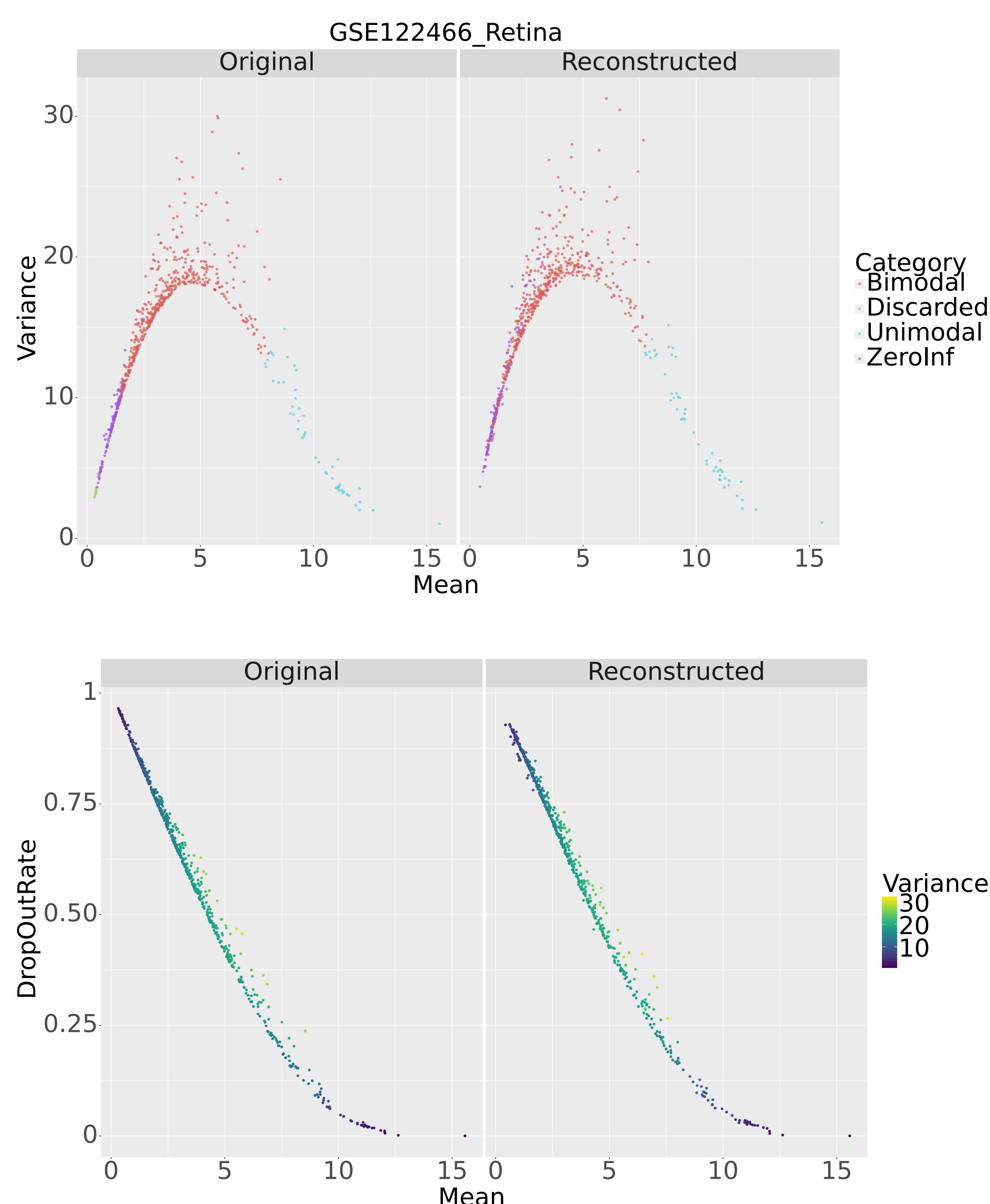
Binarization

Binarization is performed by comparing each count to the thresholds determined from the learned distributions on the reference dataset. It results in either a Boolean or undetermined value.

Minimal binarization example:

```
from scboolseq import scBoolSeq
reference = pd.read_csv("reference_scRNA_counts.csv")
scbool = scBoolSeq(data=reference)
scbool.fit() # compute binarization criteria
binarized = scbool.binarize(reference) # or other dataset
```

Sampling from the learned distributions with our algorithm reproduces the reference's profiles:



Total run-time (distribution learning + reconstruction) on specified dataset (688 genes over 5347 cells) is less than a minute.

Synthetic scRNA-Seq Generation

The synthetic RNA-Seq data **from Boolean states** is generated by **biased sampling** from the estimated gene count distributions.

- Synthetic scRNA-Seq counts use biased sampling to **reflect the underlying Boolean value**.
- Drop-outs are simulated** with probabilities that decay exponentially with the sampled expression value.

Minimal synthesis example:

```
from scboolseq import scBoolSeq
reference = pd.read_csv("reference_scRNA_counts.csv")
scbool = scBoolSeq(data=reference)
scbool.simulation_fit() # compute simulation criteria
syndata = scbool.simulate(boolean_trace)
```

