*Bioinformatics*, 38(11), 2022, 3126–3127
https://doi.org/10.1093/bioinformatics/btac271
Advance Access Publication Date: 15 April 2022
Applications Note

Gene expression

# scSampler: fast diversity-preserving subsampling of large-scale single-cell transcriptomic data

**Dongyuan Song**[1,†], **Nan Miles Xi**[2,†], **Jingyi Jessica Li**[3,*] **and Lin Wang**[4,*]

[1]Bioinformatics Interdepartmental Ph.D. Program, University of California, Los Angeles, CA 90095-7246, USA, [2]Department of Mathematics and Statistics, Loyola University Chicago, Chicago, IL 60660-1601, USA, [3]Department of Statistics, University of California, Los Angeles, CA 90095-1554, USA and [4]Department of Statistics, The George Washington University, Washington, DC 20052-0086, USA

*To whom correspondence should be addressed.

†The authors wish it to be known that, in their opinion, the first two authors should be regarded as Joint First Authors.

Associate Editor: Olga Vitek

Received on January 3, 2022; revised on March 2, 2022; editorial decision on March 21, 2022; accepted on April 12, 2022

## Abstract

**Summary:** The number of cells measured in single-cell transcriptomic data has grown fast in recent years. For such large-scale data, subsampling is a powerful and often necessary tool for exploratory data analysis. However, the easiest random subsampling is not ideal from the perspective of preserving rare cell types. Therefore, diversity-preserving subsampling is required for fast exploration of cell types in a large-scale dataset. Here, we propose scSampler, an algorithm for fast diversity-preserving subsampling of single-cell transcriptomic data.

**Availability and implementation:** scSampler is implemented in Python and is published under the MIT source license. It can be installed by "pip install scsampler" and used with the Scanpy pipline. The code is available on GitHub: https://github.com/SONGDONGYUAN1994/scsampler. An R interface is available at: https://github.com/SONGDONGYUAN1994/rscsampler.

**Contact:** jli@stat.ucla.edu or linwang@gwu.edu

**Supplementary information:** Supplementary data are available at *Bioinformatics* online.

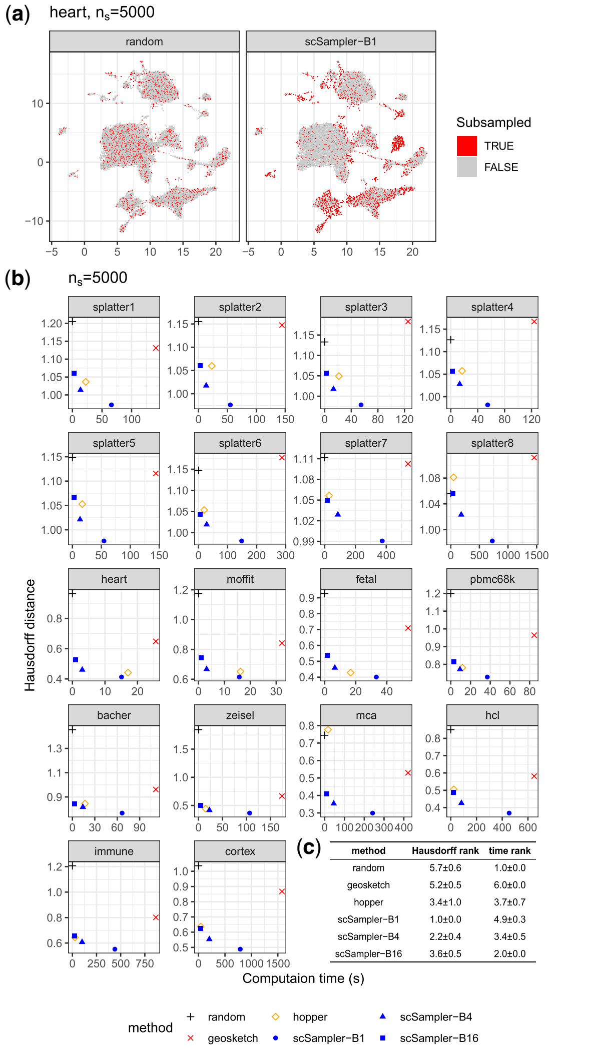
## 1 Introduction

Single-cell RNA sequencing (scRNA-seq) technologies have undergone rapid development in recent years. A remarkable achievement is the generation of large-scale datasets, and there are even datasets containing a million cells (see Supplementary Table S1). Such massive scRNA-seq datasets have impeded exploratory data analysis (e.g., visualization) on standard computers.

An intuitive solution to this 'big data' challenge is to subsample (downsample) a large-scale dataset, i.e. to select a subset of representative cells. Random subsampling is fast and has been implemented in popular pipelines such as Seurat (Satija *et al.*, 2015) and Scanpy (Wolf *et al.*, 2018). However, random subsampling may miss rare cell types and is thus not ideal for preserving the transcriptome diversity. To overcome this drawback, Hie *et al.* (2019) proposed the algorithm Geosketch for 'intelligently selecting' a subset (called 'sketching') of single cells. Geosketch aims to evenly sample cells across the transcriptome space by approximately minimizing the Hausdorff distance between the subsample and the original sample. In the follow-up algorithm Hopper (DeMeo and Berger, 2020), the authors improved the performance of Geosketch in terms of reducing the Hausdorff distance. In fact, prior to Geosketch and Hopper and outside of the single-cell field, this 'intelligent subsampling' problem has been well studied in the field of computer experiment design, in which the 'space-filling design' implements the idea of even subsampling (Joseph, 2016). Popular space-filling designs include the minimax and maximin distance designs (Johnson *et al.*, 1990). Geosketch and Hopper conceptually belong to the minimax distance design, which can be approximated by, but is much more computationally intensive than, the maximin distance design (Johnson *et al.*, 1990). Here, we propose scSampler, a Python package for fast diversity-preserving subsampling of large-scale single-cell transcriptomic data. By 'diversity-preserving sampling', scSampler implements the maximin distance design to make cells in the subsample as separative as possible. We show that scSampler outperforms existing subsampling methods in minimizing the Hausdorff distance between the subsample and the original sample. Moreover, scSampler is fast and scalable for million-level data.

## 2 Implementation

The input is a matrix $\mathbf{X} \in \mathbb{R}^{n \times p}$. The columns of $\mathbf{X}$ correspond to $p$ features (by default, top $p$ PCs from a cell-by-gene log(count + 1) matrix and scaled to $[0, 1]$), and rows correspond to $n$ cells. Then we have a set of cells $\mathcal{X} = \{x_1, \ldots, x_n\}$, where $x_i \in \mathbb{R}^p$ is a row vector of $\mathbf{X}$. Our goal is to find a size $n_s$ subset $\mathcal{X}_s \subset \mathcal{X}$ that minimizes the Hausdorff distance



| method | Hausdorff rank | time rank |
|---|---|---|
| random | 5.7±0.6 | 1.0±0.0 |
| geosketch | 5.2±0.5 | 6.0±0.0 |
| hopper | 3.4±1.0 | 3.7±0.7 |
| scSampler−B1 | 1.0±0.0 | 4.9±0.3 |
| scSampler−B4 | 2.2±0.4 | 3.4±0.5 |
| scSampler−B16 | 3.6±0.5 | 2.0±0.0 |

**Fig. 1.** Benchmarking scSampler against other subsampling methods. (**a**) UMAP + scubi visualization (Hou and Ji, 2022) of selected cells in the original data by random subsampling and scSampler-*B*1, respectively. (**b**) Scatter plots of Hausdorff distance against computation time. (**c**) Summary of the performance of each method. The table shows the mean ranks and standard deviations across all datasets and subsample sizes

$$d_{\mathrm{H}}(\mathcal{X}_s, \mathcal{X}) = \max_{x_i \in \mathcal{X}} \min_{x_j \in \mathcal{X}_s} d(x_i, x_j), \qquad (1)$$

where $d(\cdot,\cdot)$ is the Euclidean distance. The Hausdorff distance in (1) measures the distance from $\mathcal{X}_s$ to $\mathcal{X}$. A small Hausdorff distance means that all data points in $\mathcal{X}$ are represented well by at least one data point in $\mathcal{X}_s$. Since the direct optimization of (1) is computationally expensive, we propose to approximate the optimality and search for $\mathcal{X}_s$ via the following optimization problem:

$$\min_{x_i, x_j \in \mathcal{X}_s} \sum_{i=1}^{n_s - 1} \sum_{j=i+1}^{n_s} \frac{1}{[d(x_i, x_j)]^{\alpha}}, \qquad (2)$$

for a sufficiently large $\alpha$ (Joseph, 2016). Cells in $\mathcal{X}_s$ obtained from (2) have maximized distance and minimized similarity between each other and therefore can represent the diversity of $\mathcal{X}$. Joseph (2016) and our numerical results show that $\alpha = 4p$ is big enough and keeps the algorithm numerically stable, so we set $\alpha = 4p$ by default. For computational efficiency, scSampler can randomly split the original sample into *B* subsets and perform subsampling on each subset. Such a splitting procedure, although does not improve the asymptotic scalability, dramatically reduces the running time (roughly a *B*-fold decrease) in practice. The detailed optimization algorithm and the discussion on running time are provided in Supplementary Sections S1 and S3.

## 3 Results

To comprehensively benchmark scSampler—including three variants: scSampler-*B*1 (no sample splitting; the slowest), scSampler-*B*4 (splitting the sample into four subsets), and scSampler-*B*16 (splitting the sample into 16 subsets; the fastest)—against random sampling and two state-of-the-art subsampling methods, Geosketch and Hopper, we use the scRNA-seq simulator Splatter (Zappia *et al.*, 2017) to generate eight simulated datasets, and we collect 10 real datasets (see Supplementary Table S1). Figure 1a shows an example that illustrates the difference between random subsampling and scSampler: compared to random sampling, scSampler selects more cells from small cell clusters. Quantitatively, we compare subsampling methods by two measures: (i) the Hausdorff distance between the subsample and the original sample, (ii) computation time, both of which are better if smaller (Fig. 1b, Supplementary Tables S3 and S4). Figure 1b summarizes the performance of subsampling methods in the two measures. Notably, scSampler-*B*1 consistently yields the smallest Hausdorff distances across all datasets and all subsample sizes. Moreover, scSampler is fast: on the largest cortex dataset (more than 1 million cells), scSampler-*B*1 finishes within 15 min, and scSampler-*B*16 takes only 1 min and still outperforms Geosketch and Hopper by achieving a lower Hausdorff distance. Figure 1c shows that scSampler is consistently ranked the top (smaller ranks are better) across the 18 datasets.

To verify if rare cell types are better captured by scSampler than by other methods, we calculate the Gini coefficient of cell type proportions in each subsample; a smaller Gini coefficient indicates more balanced cell types (Supplementary Section S3.3). In more than 60% of the combinations of 18 datasets and four subsample sizes, the fastest scSampler-*B*16 leads to the smallest Gini coefficient (Supplementary Table S5). Considering that the real datasets may not have accurately annotated cell types, we examine the simulated datasets and find that scSampler-*B*16 leads to the smallest Gini coefficient in 90% of the combinations of eight simulated datasets and four sample sizes, confirming that scSampler preserves rare cell types well.

## Funding

This work was supported by National Science Foundation DBI-1846216 and DMS-2113754, NIH/NIGMS R01GM120507 and R35GM140888, Johnson and Johnson WiSTEM2D Award, Sloan Research Fellowship, and UCLA David Geffen School of Medicine W.M. Keck Foundation Junior Faculty Award (to J.J.L.).

*Conflict of Interest*: The authors declare no conflict of interest.

## Data availability

The real datasets are described in Supplementary Materials S6 with access and preprocessing information. The code to reproduce the analyses is available at https://doi.org/10.5281/zenodo.5811787.

## References

DeMeo,B. and Berger,B. (2020) Hopper: a mathematically optimal algorithm for sketching biological data. *Bioinformatics*, **36**, i236–i241.

Hie,B. *et al.* (2019) Geometric sketching compactly summarizes the single-cell transcriptomic landscape. *Cell Syst.*, **8**, 483–493.

Hou,W. and Ji,Z. (2022) Unbiased visualization of single-cell genomic data with scubi. *Cell Rep. Methods*, **2**, 100135.

Johnson,M.E. *et al.* (1990) Minimax and maximin distance designs. *J. Stat. Plan. Inference*, **26**, 131–148.

Joseph,V.R. (2016) Space-filling designs for computer experiments: a review. *Qual. Eng.*, **28**, 28–35.

Satija,R. *et al.* (2015) Spatial reconstruction of single-cell gene expression data. *Nat. Biotechnol.*, **33**, 495–502.

Wolf,F.A. *et al.* (2018) Scanpy: large-scale single-cell gene expression data analysis. *Genome Biol.*, **19**, 1–5.

Zappia,L. *et al.* (2017) Splatter: simulation of single-cell RNA sequencing data. *Genome Biol.*, **18**, 1–15.