scValue: value-based subsampling of large-scale single-cell transcriptomic data

Li Huang[1], Dongsheng Chen[1]*

[1]State Key Laboratory of Common Mechanism Research for Major Diseases, Suzhou Institute of Systems Medicine, Chinese Academy of Medical Sciences & Peking Union Medical College, Suzhou 215123, China.

*To whom correspondence should be addressed.

Abstract

Motivation: Large single-cell RNA sequencing (scRNA-seq) datasets have the potential to drive significant biological discoveries but present computational and memory challenges for visualisation and analysis. Existing subsampling methods aim to improve efficiency but may not guarantee performance in downstream machine/deep learning tasks.

Results: Here, we introduce scValue, a Python package designed for value-based subsampling of large scRNA-seq datasets. scValue prioritises cells of higher value (indicating greater utility for cell type identification) over cells of lower value, and allocates more representation in subsamples to cell types with greater value variability. Using publicly available datasets ranging from tens of thousands to millions of cells, we show that scValue provides fast computation, enhances cell type separation, and maintains balanced cell type proportions. These capabilities support effective downstream learning tasks, including automatic cell type annotation, label transfer, and label harmonisation across datasets.

Availability: scValue is an open-source Python package available for installation via pip from https://pypi.org/project/scvalue/, with the source code freely accessible at https://github.com/LHBCB/scvalue.

CONTACT: cds@ism.pumc.edu.cn

1 Introduction

Large-scale single-cell transcriptomic atlases from resources such as the Human Cell Atlas (Regev, et al., 2017), CELLxGENE (CZI Single-Cell Biology Program, et al., 2023), and SCAR (Deng, et al., 2024) have become essential references for individual smaller-scale studies (Johnston, et al., 2024). Visualising and analysing these atlases can pose computational and memory challenges. By creating representative subsets, or "sketches", of vast datasets, researchers can derive meaningful insights in a more efficient and cost-effective manner (Hie, et al., 2019).

Several sketching methods have been developed so far. As included in Seurat (Satija, et al., 2015) and Scanpy (Wolf, et al., 2018) pipelines, uniform subsampling is the fastest and simplest method, but may fail to preserve transcriptional diversity. GeoSketch (Hie, et al., 2019) addresses this by using equal-sized, non-overlapping boxes to approximate the gene expression space and selecting representative cells within each box to form a sketch. Sphetcher (Do, et al., 2020), in contrast, covers cells by small-radius spheres that can better capture the transcriptomic landscape. Both approaches allow for more evenly distributed sampling across diverse cell types. Hopper (DeMeo and Berger, 2020) generates a subsample with theoretical guarantees to mirror the original dataset; a partition-based variant, TreeHopper, was further proposed to speed up sketching without considerably compromising quality.

GeoSketch, Sphetcher, and (Tree)Hopper commonly utilises a minimax distance design (Johnson, et al., 1990) that minimises the Hausdorff distance, i.e., the maximum distance between any point in the original dataset and the nearest point in the sketch. scSampler (Song, et al., 2022) adopts a maximin distance design (Johnson, et al., 1990) that maximises an inverse distance measure to obtain a subset holding cells as distinct as possible. Unlike these methods, Kernel Herding (KH) constructs a “stand-in” sketch (Baskaran, et al., 2022) that prioritises maintaining the original cell type distribution, rather than optimising the distance metrics.

Machine/deep learning (ML/DL) is increasingly vital in single-cell analysis (Ma and Xu, 2022). Relatively limited empirical research has been conducted to evaluate the performance of sketching methods in downstream learning tasks. Moreover, assessing how effectively different methods capture rare cell types, as quantified by the Gini coefficient, is an important consideration (Song, et al., 2022). In this study, we present scValue, a Python package designed to subsample a single-cell RNA sequencing (scRNA-seq) dataset based on each cell’s value. Cells deemed to have higher value (i.e., more beneficial for identifying cell types) are preferentially chosen to build the sketch, whereas cells of lower value are less likely to be included. Experimental results show that, compared with other sketching methods, scValue achieves better ML/DL task performance and more balanced cell type proportions in subsamples. Additionally, the computation time of scValue exhibits superior scalability as the dataset size grows from tens of thousands to millions of cells (Supplementary Section S1).

2 Methods and Implementation

For a large scRNA-seq dataset containing expression profiles for N cells and M cell types, we denote the input feature matrix $\mathbf{X} \in \mathbb{R}^{N \times d}$ and the input label vector $\mathbf{Y} \in \{1, 2, ..., M\}^N$, respectively. By default, and same as in the other sketching methods, d is set as the top number of principal components (PCs) calculated from log-normalised gene counts. Fig. 1A depicts the schematic workflow of scValue using the inputs to create a value-based sketch consisting of S cells.

2.1 Data value computation

A random forest model (Breiman, 2001) comprising B decision trees is fitted with $(\mathbf{X}, \mathbf{Y})$. The b-th tree, denoted by f_b, is trained on a bootstrap sample drawn with replacement from $(\mathbf{X}, \mathbf{Y})$. Cell i is considered out-of-bag (OOB) if it is not selected in the b-th bootstrap, which we denote by $i \in \mathrm{OOB}_b$. Using only OOB trees to predict for cell i and averaging the prediction errors yield the cell's OOB estimate. This statistic is traditionally used for performance evaluation of random forests (Efron, 1992) and recently revised to facilitate data valuation in machine learning tasks (Kwon and Zou, 2023). In our study, each cell's data value (V) is computed as the OOB accuracy of cell type prediction

$$V_i = \frac{\sum_b^B T_b(i) \cdot \mathbf{1}(i \in \mathrm{OOB}_b)}{\sum_b^B \mathbf{1}(i \in \mathrm{OOB}_b)} \tag{1}$$

where the correctness function $T_b(i)$ is given by $\mathbf{1}(Y_i = f_b(\mathbf{X}_i))$, which evaluates to 1 if the b-th tree correctly predicts the type of cell i and 0 otherwise. Cells of higher value are expected to bring greater benefits when included in the sketch compared to those of lower value.

2.2 Value-weighted subsampling

Sketching is conducted at the cell type level in a value-weighted manner to improve the cell type balance in the sketch. The number of cells to be subsampled for cell type m is determined by considering both the cell type abundance and the standard error of data value

$$S_m = \frac{S \cdot N_m \cdot \sigma_{\bar{m}}}{\sum_j^M N_j \cdot \sigma_{\bar{j}}} \tag{2}$$

Here, N_m represents the number of cells belonging to type m in the original dataset; $\sigma_{\bar{m}}$ is given by $\sigma_m / \sqrt{N_m}$, in which σ_m denotes the standard deviation of data value for cells of type m. The denominator normalises the value-weighted subsample size across M cell types. In addition, the largest remainder method (Balinski and Young, 2010) is employed to ensure that the total allocation for all cell types sums up exactly to S.

Once S_m has been determined, three cell selection strategies can be performed: full binning (FB), mean-threshold binning (MTB), and top-pick (TP). They can achieve different levels of cell type separation in the sketch (to be demonstrated in the results). By default, scValue utilises FB since it balances top-valued cell selection and the similarity between the sketch and the original dataset. The rationales behind value-weighted subsampling and the three strategies are available in Supplementary Section S2.

2.3 Implementation

scValue is implemented in Python. The data value computation step is built upon the RandomForestClassifier class from the latest version (1.5.2) of scikit-learn. Parallelisation of tree-fitting is supported to enhance computational speed. By default, scValue accepts the top 50 PCs ($d = 50$) as the latent representation of cells; it can also take gene features or other low-dimensional representations as input. The number of decision trees B in a random forest defaults to 100. In the value-weighted subsampling step, FB is set as the default cell selection strategy. Furthermore, proportional subsampling can be carried out instead if the original cell types are highly balanced or if users wish to maintain the original cell type proportions. scValue has $O(BdN \log N)$ computational complexity, which is discussed in Supplementary Section S2.

3 Results

The experiments involve 20 large scRNA-seq datasets of sizes ranging from 31 thousand to four million cells, covering over 30 tissues across five species. Data descriptions are provided in Supplementary Section S1.

3.1 Demonstration: balanced cell type proportions and enhanced separation of cell types using scValue

The mouse T cell dataset (Qiu, et al., 2024) consisting of ~90k cells and four cell types is utilised to illustrate the efficacy of scValue in achieving cell type balancing and separation. From this dataset, 10% (~9k) cells were subsampled using scValue. As shown in Fig. 1B, the original dataset (refered to as mTC) and the respective sketches derived from the three subsampling strategies (FB, MTB, and TP) are visualised via uniform manifold approximation and projection (UMAP) plots. Notably, natural killer cells, the relatively rare cell type, were effectively preserved by the sketches, as evidenced by the denser clustering of this cell type. Moving from FB to MTB and finally to TP, the sketches contained a progressively greater proportion of high-value cells, thereby enhancing cell type separation. This indicates that scValue can adaptively emphasise top-valued cells, facilitating the creation of flexible sketches. In subsequent experiments, the default strategy, FB, is adopted in scValue to maintain a balance between the separation of cell types and similarity to the original dataset.

3.2 Evaluation: computation time, Gini coefficient, and Hausdorff distance

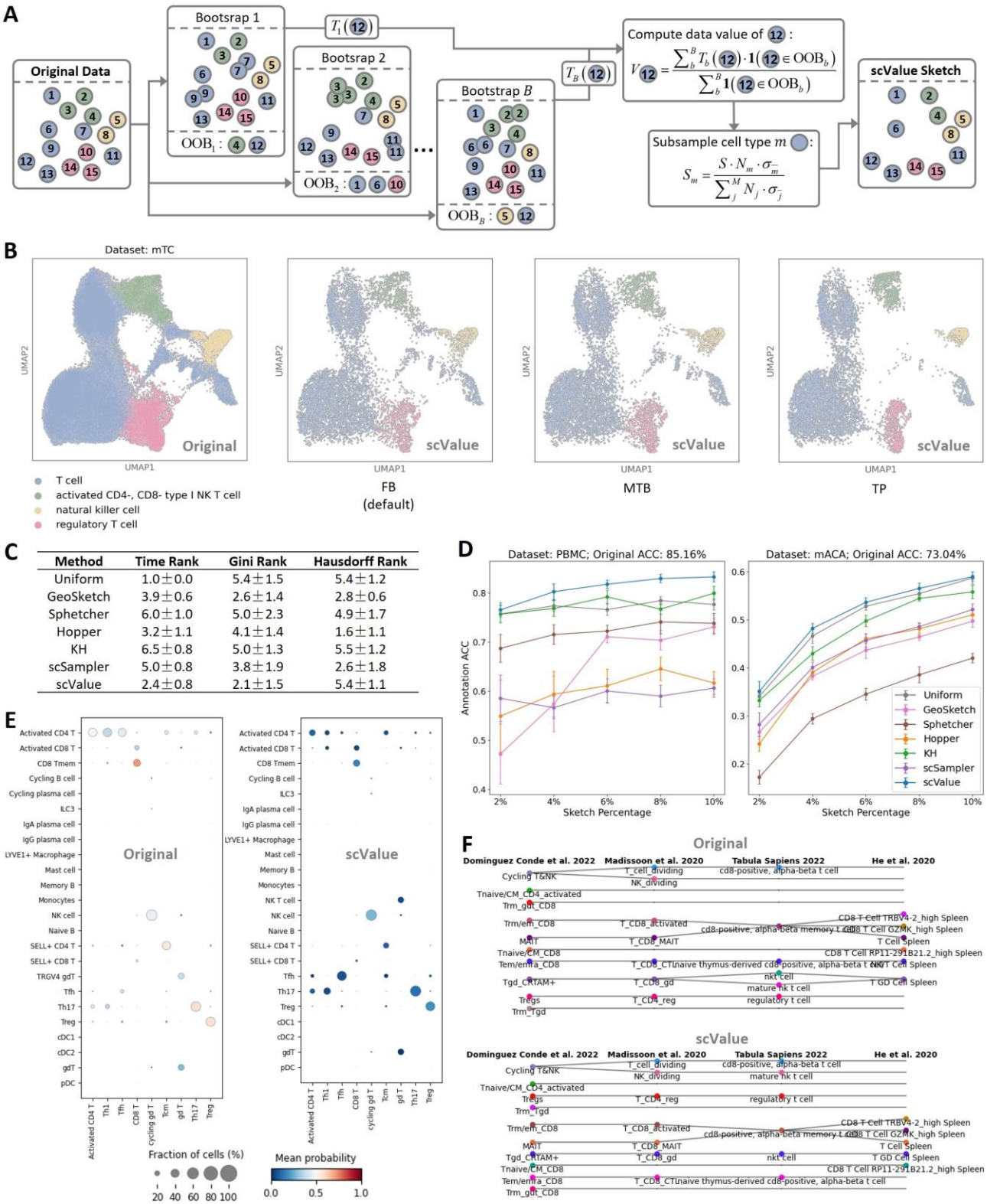
scValue was comprehensively benchmarked against the other six methods using the 20 datasets. For each dataset, a 10% sketch was created by each of the seven methods. Three metrics were calculated for each sketch: computation time, Gini coefficient, and Hausdorff distance. Fig. 1F summarises the overall rankings of each method regarding the metrics based on the 20 datasets. Specifically, scValue achieved the second lowest time rank of 2.4 ± 1.1 (only underperforming Uniform), the lowest Gini rank of 5.5 ± 1.2, and a Hausdorff rank of 5.4 ± 1.1 lower than Uniform (5.4 ± 1.2) and KH (5.5 ± 1.2). This suggests a trade-off introduced by scValue: it excels in computational efficiency and capturing rare cell types (balancing cell type proportions), at the expense of reduced similarity to the original dataset. Nonetheless, the similarity is still on par or better than that of Uniform. Detailed results are available in Supplementary Section S3.

3.3 Comparison: ML/DL performance of sketching methods

ML/DL models used for cell type annotation can be assessed based on accuracy, when reference and query data share the same cell type annotation styles. This experiment focuses on two distinct models, each applied to a distinct dataset:

- scArches (Lotfollahi, et al., 2022) with the 31k human peripheral blood mononuclear cells (PBMC) dataset (Abdelaal, et al., 2019)
- ACTINN (Ma and Pellegrini, 2020) with the 356k mouse aging cell atlas (mACA) dataset (Tabula Muris, 2020)

These model-dataset pairs have been investigated in (Du, et al., 2023) and (Chen, et al., 2023), respectively. In our study, for each pair, we respectively applied the seven subsampling methods (Uniform, GeoSketch, Sphetcher, Hopper, KH, scSampler, and scValue) to the reference partition of the dataset with sketch percentages ranging from 2% to 10%, while keeping the query partition unchanged. The model was trained on the reference sketches and then tested on the query partition. To ensure robustness, this operation was repeated ten times for each sketch, with the mean and standard deviation of accuracy being recorded. Fig. 1D summarises the results in line plots with error bars. Evidently, scValue consistently outperformed other methods for both model-dataset pairs at all sketch percentages. The detailed experiment procedures and results are presented in Supplementary Section S4.



Method	Time Rank	Gini Rank	Hausdorff Rank
Uniform	1.0±0.0	5.4±1.5	5.4±1.2
GeoSketch	3.9±0.6	2.6±1.4	2.8±0.6
Sphetcher	6.0±1.0	5.0±2.3	4.9±1.7
Hopper	3.2±1.1	4.1±1.4	1.6±1.1
KH	6.5±0.8	5.0±1.3	5.5±1.2
scSampler	5.0±0.8	3.8±1.9	2.6±1.8
scValue	2.4±0.8	2.1±1.5	5.4±1.1

Figure 1. The workflow and result demonstration of scValue. (A) scValue fits a random forest model, computes the (out-of-bag) OOB accuracy as the data value for each cell, and performs value-weighted subsampling for each cell type. (B) Example scValue sketches of the mTC dataset under three subsampling strategies: FB (full-binning, default), MTB (mean-threshold binning), and TP (top-pick). (C) Evaluation of subsampling methods with 20 large scRNA-seq datasets of varying size (31k~4m) by comparing computation time, Gini coefficient, and Hausdorff distance. (D) Evaluation of subsampling methods by comparing cell type annotation accuracy. Applications of scValue for maintaining (E) label transfer consistency and (F) label harmonisation consistency.

3.4 Application: label transfer learning by CellTypist

When reference and query data have different annotation styles, a label transfer learning model can be evaluated by examining the discrepancies between the annotations transferred from the reference to the query and the latter's original annotations. This experiment concerns the 428k intestinal cell (Gut) reference dataset (Elmentaite, et al., 2021) and the 42k immune cell query dataset (James, et al., 2020), which have been utilised in the CellTypist tutorial (Dominguez Conde, et al., 2022). In line with the tutorial, we reduced the Gut dataset to a cell type-balanced dataset of 55k cells. From it we applied scValue with proportional subsampling to generate a 10% sketch. A CellTypist model was then trained on the sketch to transfer its annotations to T-related cells in the query dataset. Dot plots in Fig. 1E compare the original annotation alignment in the tutorial (left) with the alignment derived from the scValue sketch (right). For comprehensive assessment, we also included dot plots illustrating alignments for sketches obtained by the other six methods, as detailed in Supplementary Section S5. scValue exhibited the closest alignment to the original one, whereas the other methods showed greater tendencies for misassignments.

3.5 Application: label harmonisation by CellHint

Prior to integrating multiple datasets with different annotation tyles to construct a standardised atlas, CellHint (Xu, et al., 2023) can be used to match and harmonise the diverse cell types across the datasets. In this experiment, following the harmonisation workflow in the CellHint tutorial, we utilised the 200k Spleen dataset (Dominguez Conde, et al., 2022) encompassing cells from four independent studies. From the dataset scValue was employed to sketch a 10% subset that was then fed into CellHint to learn the relationships between T cell-related types across the different studies. Fig. 1F (right) shows the output in the form of a tree plot, which is compared with the original tree plot (left) for the full dataset. Additionally, tree plots for sketches produced by the other six methods are presented in Supplementary Section S6. scValue demonstrated the best performance in preserving the various relationships depicted in the original tree plot, while the other methods displayed varying degrees of inconsistencies, including the introduction of incorrect and/or additional relationships.

Acknowledgements

We thank the high-throughput sequencing and high-performance computing platform of the Institute of Systems Medicine for technical support.

Funding

This work has been supported by the National Natural Science Foundation of China (32300560); Natural Science Foundation of Jiangsu Province [BK20220279]; CAMS Innovation Fund for Medical Sciences (CIFMS) [2021-I2M-1-061, 2022-I2M-2-004, 2023-I2M-2-005]; Non-profit Central Research Institute Fund of Chinese Academy of Medical Sciences [2022-RC416-01]; NCTIB Fund for R&D Platform for Cell and Gene Therapy, the Suzhou Municipal Key Laboratory [SZS2022005]; Gusu Innovation and Entrepreneurship Leading Talents Program [ZXL2022475]; the Special Research Fund for Central Universities, Peking Union Medical College (3332024089).

Conflict of Interest: none declared.

Data availability

The 20 real scRNA-seq datasets used in this study are described in Supplementary Section S1 with information on how to access them.

References

Abdelaal, T., *et al.* A comparison of automatic cell identification methods for single-cell RNA sequencing data. *Genome Biol* 2019;20(1):194.
Balinski, M.L. and Young, H.P. Fair representation: meeting the ideal of one man, one vote. Rowman & Littlefield; 2010.
Baskaran, V.A., *et al.* Distribution-based sketching of single-cell samples. In, *Proceedings of the 13th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics*. 2022. p. 1-10.
Breiman, L. Random forests. *Machine learning* 2001;45:5-32.
Chen, J., *et al.* Transformer for one stop interpretable cell type annotation. *Nat Commun* 2023;14(1):223.
CZI Single-Cell Biology Program, *et al.* CZ CELL× GENE Discover: A single-cell data platform for scalable exploration, analysis and modeling of aggregated data. *BioRxiv* 2023:2023.2010. 2030.563174.
DeMeo, B. and Berger, B. Hopper: a mathematically optimal algorithm for sketching biological data. *Bioinformatics* 2020;36(Suppl_1):i236-i241.
Deng, Y., *et al.* SCAR: single-cell and Spatially-resolved Cancer Resources. *Nucleic Acids Research* 2024;52(D1):D1407-D1417.
Do, V.H., Elbassioni, K. and Canzar, S. Sphetcher: Spherical Thresholding Improves Sketching of Single-Cell Transcriptomic Heterogeneity. *iScience* 2020;23(6):101126.
Dominguez Conde, C., *et al.* Cross-tissue immune cell analysis reveals tissue-specific features in humans. *Science* 2022;376(6594):eabl5197.
Du, Z.H., *et al.* scPML: pathway-based multi-view learning for cell type annotation from single-cell RNA-seq data. *Commun Biol* 2023;6(1):1268.
Efron, B. Jackknife-after-bootstrap standard errors and influence functions. *Journal of the Royal Statistical Society Series B: Statistical Methodology* 1992;54(1):83-111.
Elmentaite, R., *et al.* Cells of the human intestinal tract mapped across space and time. *Nature* 2021;597(7875):250-255.
Hie, B., *et al.* Geometric Sketching Compactly Summarizes the Single-Cell Transcriptomic Landscape. *Cell Syst* 2019;8(6):483-493 e487.
James, K.R., *et al.* Distinct microbial and immune niches of the human colon. *Nat Immunol* 2020;21(3):343-353.
Johnson, M.E., Moore, L.M. and Ylvisaker, D. Minimax and maximin distance designs. *Journal of statistical planning and inference* 1990;26(2):131-148.
Johnston, K.G., *et al.* Small data methods in omics: the power of one. *Nature Methods* 2024:1-6.
Kwon, Y. and Zou, J. Data-oob: Out-of-bag estimate as a simple and efficient data value. In, *International Conference on Machine Learning*. PMLR; 2023. p. 18135-18152.
Lotfollahi, M., *et al.* Mapping single-cell data to reference atlases by transfer learning. *Nat Biotechnol* 2022;40(1):121-130.
Ma, F. and Pellegrini, M. ACTINN: automated identification of cell types in single cell RNA sequencing. *Bioinformatics* 2020;36(2):533-538.
Ma, Q. and Xu, D. Deep learning shapes single-cell data analysis. *Nat Rev Mol Cell Biol* 2022;23(5):303-304.
Qiu, C., *et al.* A single-cell time-lapse of mouse prenatal development from gastrula to birth. *Nature* 2024;626(8001):1084-1093.
Regev, A., *et al.* The human cell atlas. *elife* 2017;6:e27041.
Satija, R., *et al.* Spatial reconstruction of single-cell gene expression data. *Nat Biotechnol* 2015;33(5):495-502.
Song, D., *et al.* scSampler: fast diversity-preserving subsampling of large-scale single-cell transcriptomic data. *Bioinformatics* 2022;38(11):3126-3127.
Tabula Muris, C. A single-cell transcriptomic atlas characterizes ageing tissues in the mouse. *Nature* 2020;583(7817):590-595.
Wolf, F.A., Angerer, P. and Theis, F.J. SCANPY: large-scale single-cell gene expression data analysis. *Genome Biol* 2018;19(1):15.
Xu, C., *et al.* Automatic cell-type harmonization and integration across Human Cell Atlas datasets. *Cell* 2023;186(26):5876-5891 e5820.

Supplementary Materials

scValue: value-based subsampling of large-scale single-cell transcriptomic data

Li Huang, Dongsheng Chen

S1 Large scRNA-seq datasets used in this paper

S1.1 Data description

As detailed in Table S1, this study utilised 20 large scRNA-seq datasets, which contain between 31 thousand and four million cells, covering four to 197 cell types and spanning over 30 tissues across five species. One dataset, PBMC, was originally downloaded in the Matrix Market format comprising four separate files for barcodes, genes, expression values, and meta data; the remaining 19 datasets were originally downloaded in the h5ad format. All datasets were stored and processed in the AnnData format using the Scanpy pipeline.

Table S1: Overview of the 20 scRNA-seq datasets

Dataset	# of cells	# of cell types	Tissue	Species	Literature
PBMC	31,021	10	blood	Homo sapiens	PMID31500660
mTC	89,429	4	embryo	Mus musculus	PMID38355799
cMTG	112,929	18	middle temporal gyrus	Pan troglodytes	PMID37824638
Lung	134,076	29	upper lobe of left lung	Homo sapiens	PMID38597954
PAC	139,054	18	primary auditory cortex	Homo sapiens	PMID37824655
gMTG	139,945	18	middle temporal gyrus	Gorilla gorilla	PMID37824638
Liver	167,598	14	liver	Homo sapiens	PMID35021063
Plage	184,623	11	carotid artery segment, coronary artery, femoral artery	Homo sapiens	bioRxiv DOI 10.1101/2024.09.11.612431
GSCL	191,230	9	gonad	Homo sapiens	PMID35794482
Spleen	200,664	102	spleen	Homo sapiens	PMID35549406
T&ILC	216,611	16	cross-tissue	Homo sapiens	PMID35549406
Retina	265,767	31	chorioretinal region, fovea centralis, macula lutea proper, peripheral region of retina, retinal neural layer	Homo sapiens	bioRxiv DOI 10.1101/2023.11.07.566105
mRetina	330,930	27	optic cup retina	Mus musculus	PMID38812536
mACA	356,213	197	cross-tissue	Mus musculus	PMID32669714
Tumour	391,989	11	blood, breast, colorectum, liver, lung, ovary, skin epidermis, uvea	Homo sapiens	PMID38972873
Gut	428,469	134	gut	Homo sapiens	PMID34497389
Monkey	1,500,000	14	cross-tissue	Macaca mulatta	PMID37824616
mCNS	2,106,206	15	embryo	Mus musculus	PMID38355799
mEmbryo	3,267,338	5	embryo	Mus musculus	PMID38355799
Fetal	4,062,980	72	cross-tissue	Homo sapiens	PMID33184181

S1.2 Data availability

Download links for the datasets are listed below. Specifically, PBMC, Spleen, mACA, and Gut were obtained from the links provided by their corresponding literature; the remaining 16 datasets were downloaded from CELLxGENE (https://cellxgene.cziscience.com/datasets).

- PBMC from https://portals.broadinstitute.org/single_cell/study/SCP424/single-cell-comparisonpbmc-data. We used the Matrix Market files, including barcodes.txt, genes.txt, gene_sorted-expression.txt, and meta.txt.
- mTC from https://cellxgene.cziscience.com/collections/45d5d2c3-bc28-4814-aed6-0bb6f0e11c82. We used the "Major cell cluster: T cells" h5ad file.
- cMTG from https://cellxgene.cziscience.com/collections/4dca242c-d302-4dba-a68f-4c61e7bad553. We used the "Chimpanzee: Great apes study" h5ad file.
- Lung from https://cellxgene.cziscience.com/collections/2a9a17c9-1f61-4877-b384-b8cd5ffa4085. We used the "All annotated cells from infection 2" h5ad file.
- PAC from https://cellxgene.cziscience.com/collections/d17249d2-0e6e-4500-abb8-e6c93fa1ac6f. We used the "Dissection: Primary auditory cortex(A1)" h5ad file.
- gMTG from https://cellxgene.cziscience.com/collections/4dca242c-d302-4dba-a68f-4c61e7bad553. We used the "Gorilla: Great apes study" h5ad file.
- Liver from https://cellxgene.cziscience.com/collections/74e10dc4-cbb2-4605-a189-8a1cd8e44d8c. We used the "All cells from human liver dataset" h5ad file.
- Plage from https://cellxgene.cziscience.com/collections/db70986c-7d91-49fe-a399-a4730be394ac. We used the "plaque atlas" h5ad file.
- GSCL from https://cellxgene.cziscience.com/collections/661a402a-2a5a-4c71-9b05-b346c57bc451. We used the "Human Somatic Cell Lineage" h5ad file.
- Spleen from https://celltypist.cog.sanger.ac.uk/Resources/Organ_atlas/Spleen/Spleen.h5ad, which was downloaded directly as Spleen.h5ad file.
- T&ILC from https://cellxgene.cziscience.com/collections/62ef75e4-cbea-454e-a0ce-998ec40223d3. We used the "T & innate lymphoid cells" h5ad file.
- Retina from https://cellxgene.cziscience.com/collections/4c6eaf5c-6d57-4c76-b1e9-60df8c655f1e. We used the "scRNA-seq of human retina - all cells" h5ad file.
- mRetina from https://cellxgene.cziscience.com/collections/a0c84e3f-a5ca-4481-b3a5-ccfda0a81ecc. We used the "MRCA: scRNA-seq of the mouse retina - all cells" h5ad file.
- mACA from https://figshare.com/articles/dataset/Processed_files_to_use_with_scanpy_/8273102/3. We used the tabula-muris-senis-bbknn-processed-official-annotations.h5ad file (the Version 3 atlas).
- Tumour from https://cellxgene.cziscience.com/collections/3f7c572c-cd73-4b51-a313-207c7f20f188. We used the "A multi-tissue single-cell tumor microenvironment atlas" h5ad file.
- Gut from https://cellgeni.cog.sanger.ac.uk/gutcellatlas/Full_obj_raw_counts_nosoupx.h5ad, which was downloaded as the human intestinal cell dataset.
- Monkey from https://cellxgene.cziscience.com/collections/8c4bcf0d-b4df-45c7-888c-74fb0013e9e7. We used the h5ad file named "A single-cell multi-omic atlas spanning the adult rhesus macaque brain (1.5 million cell subset)".
- mCNS from https://cellxgene.cziscience.com/collections/45d5d2c3-bc28-4814-aed6-0bb6f0e11c82. We used "Major cell cluster: CNS neurons" h5ad file.
- mEmbryo from https://cellxgene.cziscience.com/collections/45d5d2c3-bc28-4814-aed6-0bb6f0e11c82. We used the "Major cell cluster: Mesoderm" h5ad file.
- Fetal from https://cellxgene.cziscience.com/collections/c114c20f-1ef4-49a5-9c2e-d965787fb90c. We used the "Survey of human embryonic development" h5ad file.

S2 Additional discussion of methods and computational complexity

S2.1 Rationales of value-weighted subsampling and the three subsampling strategies

The rationale behind value-weighted subsampling is as follows. Rare cell types are often more challenging to learn in a random forest due to their underrepresentation, while heterogeneous cell types that contain diverse or complex data also present learning difficulties. Both conditions are expected to result in larger variability in data values. To address this, Eq. (2) seeks to adequately capture rare and heterogenous cell types through the allocation of more cells in the sketch, thereby preserving crucial biological information.

Once the sketch size S_m for cell type m has been determined, three cell selection strategies can be performed:

1. Full binning (FB) involves dividing the values for cells of each type into bins with 0.1 intervals, ranging from 0.0 to 1.0. Within each bin, top-valued cells are included in the final sketch.
2. Mean-threshold binning (MTB) uses the same binning setting as FB; however, only those bins above the mean value are considered. Bins that fall below the mean are combined into a single bin with a lower bound of 0. MTB emphasises cells that exhibit above-average values over less-valuable cells.
3. Top-pick (TP) directly selects cells with the highest values from each type, focusing on value-maximisation in the final sketch without binning constraints. These three strategies can cater to different analytical needs.

By default, scValue utilises FB since it balances top-valued cell selection and the similarity between the sketch and the original dataset.

S2.2 Discussion on computational complexity

This study compared scValue with six existing subsampling methods, i.e., Uniform (implemented in Seurat and Scanpy pipelines), GeoSketch (Hie, et al., 2019), Sphetcher (Do, et al., 2020), Hopper (DeMeo and Berger, 2020), KH (Baskaran, et al., 2022), and scSampler (Song, et al., 2022). We reviewed their theoretical computational complexities as documented in the respective publications. A summary of these complexities, along with that of scValue, is provided in Table S2 for comparative analysis.

Given a large scRNA-seq dataset consisting of N cells and d features (PCs by default), the sketching objective is to extract a subset of cells that effectively represents the original dataset. Among the seven methods evaluated, five include additional parameters or variables in their computational complexity. Specifically, scValue has a parameter B, the number of trees in a random forest, which defaults to 100. TreeHopper, similarly, involves a parameter B, denoting the number of partitions of the original dataset, with a default value of 1000. scSampler also utilises a parameter B for the number of partitions, evaluated in its original publication with values of 1 (no partition), 4, and 16 (Song et al., 2022). In the experiments of our study, we set $B = 4$ to balance computational cost and sketch quality. Lastly, KH incorporates a parameter D, which denotes the number of random features drawn from the original d features.

Table S2: Theoretical computational complexity for the seven sketching methods

Method	**Theoretical complexity**
Uniform	$O(1)$
Geosketch	$O(dN\log N)$ with d features
Hopper	$O(NS)$
	$O\left(\frac{NS}{B}\right)$ for TreeHopper with B partitions
Sphetcher	$O\left(\sum_i \lvert L_i \rvert\right)$ for each iteration until L converges, where $\lvert L_i \rvert$ is the size of the i-th spherical subset
Kernel Herding	$O(dDNS)$ with d features and D random features
scSampler	$O(NS)$
	$O\left(\frac{NS}{B}\right)$ if split by B subsets
scValue	$O(BdN\log N)$ with B trees and d features

Since S is considerably larger than $\log(N)$ for typical sketching applications in single-cell analysis (Hie, et al., 2019), it seems from Table S2 that the ranking of theoretical computational complexity, in ascending order, is as follows: Uniform < Geosketch < scValue < (Tree)Hopper $\approx$ scSampler (with partitioning) < Sphetcher < KH. However, it is important to note that the actual computation time may deviate from this theoretical ranking due to several factors (Cormen, et al., 2009), such as the constant and lower-order terms disregarded in Big-O notation, dependencies on specific characteristics of the data, memory management overhead, among others. Moreover, as a random forest in scValue can be easily parallelised, scValue may in practice achieve fast runtimes than most of the sketching methods, as evidenced in Fig. 1C. Another advantage of scValue is independence from in the computational complexity, implying that the computation time does not increase substantially as the data size grows. This property is illustrated in Fig. S1 using the Spleen dataset as an example. While Sphetcher shares this property, the method generally requires a significantly longer computation time than scValue.

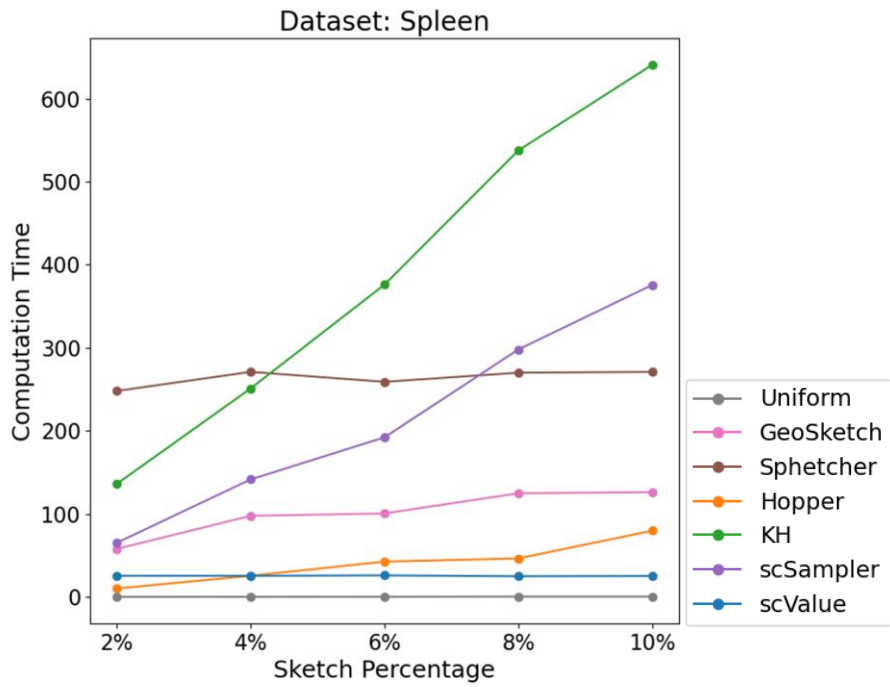


Figure S1: Computation time as sketch size increases.

S3 Evaluation: computation time, Gini coefficient, and Hausdorff distance

S3.1 Evaluation metrics

The seven sketching methods (scValue, Uniform, GeoSketch, Sphetcher, Hopper, KH, and scSampler) were evaluated across the 20 large scRNA-seq datasets (as detailed in Section S1) using three metrics.

The first metric, computation time, was measured in seconds, recording the time required for each method to complete the sketching process for each dataset.

The second metric, Gini coefficient, quantifies the cell type inequality in a subsample. A smaller Gini coefficient reflects a more balanced representation of cell types, whereas a large value indicates higher inequality. We denote this metric by G . As defined in the publication of scSampler (Song, et al., 2022)

G is calculated as

$$G = \frac{\sum_m^M (2m - M - 1) p_m}{M \sum_m^M p_m} \qquad \text{(S1)}$$

where M denotes the number of cell types; p_m denotes the proportion of cell type m .

Consistent with the scSampler literature, we used the implementation of Gini coefficient computation available at https://github.com/oliviaguest/gini.

Hausdorff distance, denoted by H , is defined as the maximum Euclidean distance ($dist$) between any point in the original dataset X and the nearest point in X_s

$$H(X, X_s) = \max_{x \in X} \min_{x_s \in X_s} dist(x, x_s) \qquad \text{(S2)}$$

A smaller value of H means that all cells in X are well-represented in X_s , implying a higher degree of similarity between the original dataset and the subsample. Conversely, a larger H suggests less similarity.

Computation time and Hausdorff distance have been commonly utilised in previous studies on sketching methods; Gini coefficient was introduced by scSampler as another important metric for sketch quality assessment. In our study, we evaluate the seven methods using these three metrics to provide a comprehensive comparison of subsampling efficiency and performance.

S3.2 Evaluation procedures and results

Each of the 20 scRNA-seq dataset (Table S1) was normalised with a scale factor of 10,000 and log-transformed $\log(x+1)$ if not already log-normalised. The top 3,000 highly variable genes (HVGs) were selected if HVGs were not provided or if a different count was not specified in the literature. The top 50 principal components (PCs) were then computed based on HVGs and used as input for the seven subsampling methods.

All methods were applied using their default parameters to create sketches containing 10% of the cells from each original dataset. For Hopper, we used the tree-partitioning variant, TreeHopper, in the experiments to balance computation time with sketch quality. scSampler was run with 4 partitions to achieve a similar balance.

For each dataset, we recorded the computation time, Gini coefficient, and Harsdorff distance of the sketches generated by each of the seven methods. The results for these metrics are illustrated in Fig. S2 and summarised in Table S3.

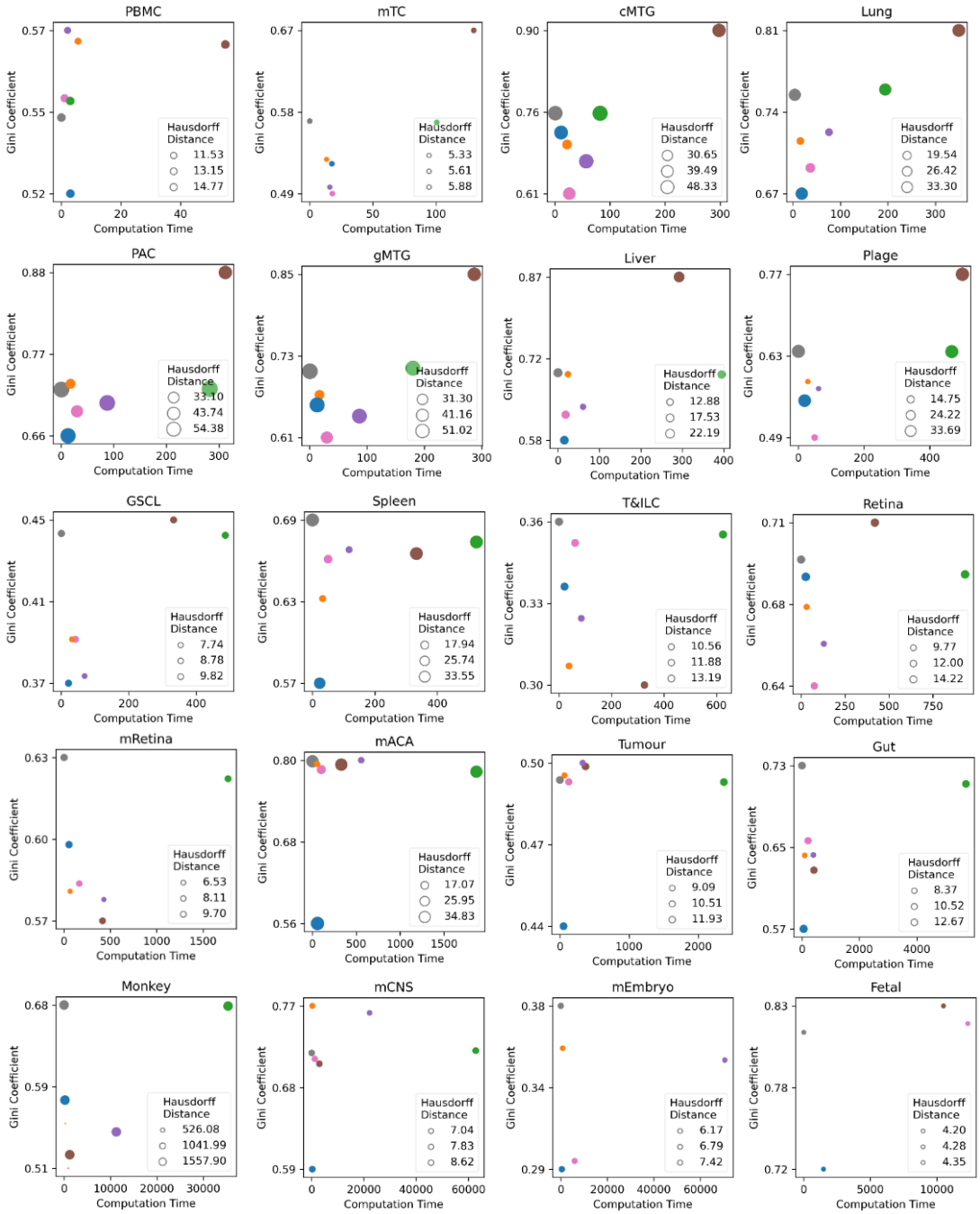


Figure S2: Comparison of the seven subsampling methods that respectively sketch 10% cells of the 20 scRNA-seq datasets in terms of computation time, Gini coefficient, and Hausdorff distance.

Table S3: Detailed statistics supporting the ranking table in Fig. 1C and the dot plots in Fig. S2.

Dataset	**Method**	**Computation time (seconds)**	**Time rank**	**Gini coefficient**	**Gini rank**	**Hausdorff distance**	**Hausdorff rank**
PBMC	Uniform	<1	1	0.5467	2	16.20	6
	GeoSketch	1	2	0.5525	4	14.34	3
	Sphetcher	55	7	0.5684	5	16.39	7
	Hopper	6	6	0.5694	6	9.91	1
	KH	3	4	0.5516	3	15.62	4
	scSampler	2	3	0.5726	7	10.27	2
	scValue	3	5	0.5242	1	16.02	5
mTC	Uniform	<1	1	0.5715	6	6.15	6
	GeoSketch	18	5	0.4920	1	6.00	3
	Sphetcher	130	7	0.6707	7	6.09	5
	Hopper	13	2	0.5295	4	5.09	2
	KH	101	6	0.5700	5	6.15	7
	scSampler	16	3	0.4991	2	5.06	1
	scValue	18	4	0.5247	3	6.01	4
cMTG	Uniform	<1	1	0.7530	6	51.97	5.5
	GeoSketch	26	4	0.6108	1	37.63	2
	Sphetcher	298	7	0.8993	7	44.12	3
	Hopper	22	3	0.6977	3	21.81	1
	KH	82	6	0.7528	5	57.17	7
	scSampler	57	5	0.6680	2	51.97	5.5
	scValue	11	2	0.7188	4	46.95	4
Lung	Uniform	4	1	0.7516	5	35.24	4
	GeoSketch	37	4	0.6898	2	21.63	3
	Sphetcher	349	7	0.8061	7	40.17	7
	Hopper	16	2	0.7125	3	12.66	1
	KH	194	6	0.7561	6	35.70	5
	scSampler	76	5	0.7200	4	12.91	2
	scValue	19	3	0.6680	1	36.57	6
PAC	Uniform	1	1	0.7200	4	62.19	5.5
	GeoSketch	31	4	0.6906	2	35.56	2
	Sphetcher	312	7	0.8781	7	44.09	3
	Hopper	18	3	0.7277	6	22.46	1
	KH	282	6	0.7209	5	65.03	7
	scSampler	88	5	0.7019	3	62.19	5.5
	scValue	13	2	0.6576	1	57.74	4
gMTG	Uniform	1	1	0.7061	5	60.89	7
	GeoSketch	30	4	0.6101	1	35.10	2
	Sphetcher	287	7	0.8468	7	44.00	3
	Hopper	17	3	0.6721	4	21.43	1
	KH	180	6	0.7106	6	55.50	5
	scSampler	87	5	0.6411	2	54.28	4
	scValue	13	2	0.6573	3	56.23	6
Liver	Uniform	<1	1	0.6968	6	18.56	6
	GeoSketch	19	3	0.6225	2	14.60	3
	Sphetcher	292	6	0.8663	7	26.84	7
	Hopper	24	4	0.6942	5	8.41	2
	KH	395	7	0.6937	4	15.35	4
	scSampler	60	5	0.6366	3	8.22	1
	scValue	16	2	0.5774	1	16.53	5

Plage	Uniform	<1	1	0.6343	6	41.61	6
	GeoSketch	50	4	0.4866	1	9.17	3
	Sphetcher	499	7	0.7665	7	43.16	7
	Hopper	29	3	0.5826	4	5.40	2
	KH	467	6	0.6339	5	41.28	5
	scSampler	62	5	0.5704	3	5.28	1
	scValue	19	2	0.5499	2	38.86	4
GSCL	Uniform	1	1	0.4475	6	10.86	7
	GeoSketch	43	4	0.3907	4	9.89	3
	Sphetcher	333	6	0.4548	7	10.08	4
	Hopper	32	3	0.3907	3	6.70	1
	KH	486	7	0.4465	5	10.38	5
	scSampler	70	5	0.3709	2	6.72	2
	scValue	22	2	0.3671	1	10.55	6
Spleen	Uniform	<1	1	0.6872	7	41.36	7
	GeoSketch	50	4	0.6594	3	16.51	3
	Sphetcher	333	6	0.6634	4	39.70	5
	Hopper	33	3	0.6316	2	10.13	1
	KH	525	7	0.6716	6	40.23	6
	scSampler	117	5	0.6661	5	10.24	2
	scValue	23	2	0.5717	1	31.34	4
T&ILC	Uniform	1	1	0.3574	7	14.33	6
	GeoSketch	62	4	0.3499	5	13.23	4
	Sphetcher	326	6	0.2998	1	12.00	3
	Hopper	39	3	0.3065	2	9.24	1
	KH	625	7	0.3529	6	14.51	7
	scSampler	85	5	0.3234	3	9.28	2
	scValue	21	2	0.3345	4	14.05	5
Retina	Uniform	1	1	0.6978	6	14.53	4
	GeoSketch	76	4	0.6389	1	11.40	3
	Sphetcher	420	6	0.7150	7	15.40	5
	Hopper	32	3	0.6757	3	7.55	1
	KH	932	7	0.6908	5	15.95	6
	scSampler	130	5	0.6585	2	7.55	2
	scValue	28	2	0.6897	4	16.45	7
mRetina	Uniform	2	1	0.6325	7	11.29	7
	GeoSketch	166	4	0.5873	4	8.16	3
	Sphetcher	417	5	0.5739	1	9.40	4
	Hopper	67	3	0.5845	3	4.94	1
	KH	1770	7	0.6249	6	9.81	5
	scSampler	432	6	0.5815	2	4.94	2
	scValue	55	2	0.6013	5	11.17	6
mACA	Uniform	1	1	0.7986	6	36.57	5
	GeoSketch	102	4	0.7864	3	18.41	3
	Sphetcher	330	5	0.7937	4	37.05	6
	Hopper	53	2	0.7947	5	8.19	1
	KH	1868	7	0.7833	2	36.05	4
	scSampler	556	6	0.8001	7	8.56	2
	scValue	59	3	0.5598	1	43.71	7
Tumour	Uniform	<1	1	0.4907	4	12.98	6
	GeoSketch	128	4	0.4901	3	10.89	3
	Sphetcher	370	6	0.4954	6	12.10	4
	Hopper	66	3	0.4923	5	7.72	2
	KH	2370	7	0.4900	2	12.81	5

	scSampler	329	5	0.4966	7	7.68	1
	scValue	53	2	0.4406	1	13.35	7
Gut	Uniform	<1	1	0.7281	7	12.94	6
	GeoSketch	212	4	0.6539	5	11.15	3
	Sphetcher	416	6	0.6248	2	11.61	4
	Hopper	96	3	0.6394	3	6.22	1
	KH	5708	7	0.7101	6	12.57	5
	scSampler	396	5	0.6398	4	6.35	2
	scValue	58	2	0.5668	1	14.81	7
Monkey	Uniform	3	1	0.6791	7	2073.07	4
	GeoSketch	903	4	0.5051	1	20.23	2
	Sphetcher	1218	5	0.5196	2	2073.81	7
	Hopper	234	3	0.5529	4	10.17	1
	KH	35300	7	0.6779	6	2072.98	3
	scSampler	11241	6	0.5439	3	2073.60	6
	scValue	202	2	0.5778	5	2073.27	5
mCNS	Uniform	11	1	0.7216	4	8.78	4
	GeoSketch	1243	4	0.7149	3	8.29	3
	Sphetcher	2995	5	0.7094	2	8.97	5
	Hopper	296	3	0.7745	7	6.95	2
	KH	62784	7	0.7242	5	9.40	7
	scSampler	22173	6	0.7667	6	6.25	1
	scValue	280	2	0.5909	1	9.14	6
mEmbryo	Uniform	13	1	0.3828	5	7.88	4
	GeoSketch	5946	4	0.2939	2	7.22	3
	Sphetcher	790	3	0.3585	4	5.94	2
	Hopper	—	—	—	—	—	—
	KH	—	—	—	—	—	—
	scSampler	70485	5	0.3517	3	5.54	1
	scValue	448	2	0.2892	1	8.05	5
Fetal	Uniform	8	1	0.8144	2	4.40	3
	GeoSketch	12324	4	0.8205	3	4.13	1
	Sphetcher	10501	3	0.8328	4	4.33	2
	Hopper	—	—	—	—	—	—
	KH	—	—	—	—	—	—
	scSampler	—	—	—	—	—	—
	scValue	1493	2	0.7198	1	**3.43**	4

S4 Comparison: machine/deep learning performance of sketching methods

S4.1 scArches evaluated on sketches of PBMC

The first machine/deep learning (ML/DL) experiment for automatic cell type annotation evaluation concerned with the scArches framework (Lotfollahi, et al., 2022) containing a collection of autoencoder-based DL architectures and the PBMC dataset (Abdelaal, et al., 2019) consisting of 31k cells. This combination of model and dataset has been investigated in a prior study (Du, et al., 2023). In alignment with this study, we split PBMC into

- a reference (training) set: comprising 20,303 cells obtained using the SeqWell, DropSeq, and Indrop sequencing protocols.
- a query (test) set: containing 16,941 cells obtained using the 10X V2 protocol.

The seven sketching methods were applied to generate sketches of the reference set at varying percentages of 2%, 4%, 6%, 8%, and 10%, respectively, while the query set remained unchanged. Each resulting sketch was used to train a SCANVI model (Xu, et al., 2021) within the scArches collection, following the official tutorial (https://docs.scarches.org/en/latest/scanvi_surgery_pipeline.html). The maximum number of epochs for vae.train(), scanvae.train(), and model.train() were set to 100, 10, and 10, respectively, to mitigate overfitting; all other parameters were left as default. To ensure robustness, this operation was repeated ten times for each sketch, with the mean and standard deviation of accuracy recorded. The experimental results are summarised in Table S4 and visualised in Fig. 1D (left).

Notably, scValue outperformed the other six methods across all sketching percentages. The average annotation accuracy (ACC) achieved by scValue was 0.8330±0.0103, which was close to the original ACC of the model trained on the full reference set. In contrast, none of the other methods reached an ACC above 0.8: Uniform (0.7769±0.0112), GeoSketch

(0.7312±0.0160), Sphetcher (0.7385±0.0205), Hopper (0.6170±0.0232), KH (0.7999±0.0140), and scSampler (0.6065±0.0190).

Moreover, a desirable characteristic of a stable sketching method is that an increase in the sketching percentage should correspond to better performance of the model trained on more data. This behaviour, however, was only observed for scValue and Sphetcher, indicating their stability across different sketching scales.

Table S4: the scArches model evaluated on sketches of the PBMC dataset; the original query annotation ACC of the model trained on the full reference data was 0.8516.

Method	**Sketch Percentage**				
	2%	**4%**	**6%**	**8%**	**10%**
Uniform	0.7575±0.0179	0.7742±0.0116	0.7664±0.0118	0.7846±0.0082	0.7769±0.0112
GeoSketch	0.4722±0.0613	0.5742±0.0566	0.7111±0.0125	0.7039±0.0197	0.7312±0.0160
Sphetcher	0.6874±0.0285	0.7156±0.0201	0.7228±0.0120	0.7413±0.0244	0.7385±0.0205
Hopper	0.5494±0.0732	0.5939±0.0466	0.6112±0.0337	0.6456±0.0240	0.6170±0.0232
KH	0.7572±0.0174	0.7686±0.0161	0.7923±0.0137	0.7674±0.0106	0.7999±0.0140
scSampler	0.5855±0.0475	0.5665±0.0225	0.6006±0.0250	0.5902±0.0222	0.6065±0.0190
scValue	**0.7658±0.0148**	**0.8029±0.0160**	**0.8182±0.0085**	**0.8297±0.0086**	**0.8330±0.0103**

S4.2 ACTINN evaluated on sketches of mACA

The second ML/DL experiment for automatic cell type annotation evaluation used the neural networks-based ACTINN model (Ma and Pellegrini, 2020) and the mACA dataset (Tabula Muris, 2020) consisting of 356k cells. This model-dataset pair has been explored in a previous study (Chen, et al., 2023). Following this study, we first filtered out cells with missing ("nan") cell type labels and then split the remaining cells based on timepoints into

- a reference (training) set: comprising 43,196 cells from mouse samples aged 18 months.
- a query (test) set: containing 140,743 cells obtained from mouse samples aged 1, 3, 21, 24, or 30 months.

Again, we applied the seven sketching methods to generate sketches of the reference set at percentages of 2%, 4%, 6%, 8%, and 10%, respectively, while keeping the query set unchanged. Each sketch was used to train an ACTINN model with the default parameters, i.e., 50 epochs, a learning rate of 0.0001, and a minibatch size of 128. The training and testing process was repeated ten times for each sketch, with the mean and standard deviation of accuracy being recorded. The experimental results are summarised in Table S5 and visualised in Fig. 1D (right).

Again, scValue outperformed the other six methods across all sketch percentages. At the 10% sketching percentage, scValue achieved an average ACC of 0.5901±0.0098, surpassing that of Uniform (0.5871±0.0129), GeoSketch (0.4973±0.0130), Sphetcher (0.4203±0.0100), Hopper (0.5107±0.0108), KH (0.5582±0.0140), and scSampler (0.5215±0.0117). Despite this performance, the highest ACC obtained with the scValue sketch remained significantly lower than the original ACC of 0.7304 attained with the full reference data. This gap (0.1403) was larger than that (0.0186) with PBMC in the first experiment.

We conjecture that this disparity may be due to the increase complexity of the mACA dataset, which may pose greater challenges in generating high quality sketches compared to the PBMC dataset in the first experiment. As shown in Table S1, mACA has the largest number of cell types (197), whereas PBMC has only 10 types. Moreover, as reflected in Fig. S2, sketches of mACA generally exhibit larger Gini coefficients and higher Hausdorff distances than those of PBMC, indicating a greater degree of imbalance and dissimilarity.

Nonetheless, it appears that mACA can facilitate more stable learning outcomes than PBMC, regardless of the sketching method used. The ACC for each method improved consistently as the sketch percentage increased.

Table S5: the ACTINN model evaluated on sketches of the mACA dataset; the original query annotation ACC of the model trained on the full reference data was 0.7304.

Methods	**Sketch Percentage**				
	2%	**4%**	**6%**	**8%**	**10%**
Uniform	0.3407±0.0103	0.4667±0.0159	0.5284±0.0084	0.5548±0.0102	0.5871±0.0129
GeoSketch	0.2664±0.0149	0.3827±0.0077	0.4369±0.0173	0.4645±0.0079	0.4973±0.0130
Sphetcher	0.1725±0.0142	0.2941±0.0109	0.3454±0.0125	0.3857±0.0169	0.4203±0.0100
Hopper	0.2416±0.0152	0.3904±0.0105	0.4608±0.0097	0.4813±0.0126	0.5107±0.0108
KH	0.3324±0.0135	0.4298±0.0142	0.4976±0.0118	0.5447±0.0058	0.5582±0.0140

scSampler	0.2821±0.0246	0.4011±0.0108	0.4572±0.0143	0.4859±0.0067	0.5215±0.0117
scValue	**0.3512±0.0205**	**0.4823±0.0102**	**0.5366±0.0091**	**0.5656±0.0108**	**0.5901±0.0098**

From these two experiments, it can be seen that scValue consistently outperformed other methods for both model-dataset pairs at all sketch percentages. Notably, at the 10% sketching level, scValue achieved accuracies that were closest to the original ACC obtained with the full reference dataset, demonstrating its superior ability to maintain performance with reduced training data.

S5 Application: label transfer learning by CellTypist

This experiment was conducted following the tutorial "Best practice in large-scale cross-dataset label transfer using CellTypist" available at:
https://colab.research.google.com/github/Teichlab/celltypist/blob/main/docs/notebook/celltypist_tutorial_cv.ipynb.

Consistent with the tutorial, we utilised the 428k Gut reference dataset (Elmentaite, et al., 2021) and the 42k immune cell query dataset (James, et al., 2020) to perform label transfer learning using the CellTypist model (Dominguez Conde, et al., 2022). The Gut dataset was reduced to a cell type-balanced dataset of 55k cells, from which scValue was applied to generate a 10% sketch. Here, we adopted proportional subsampling instead of value-weighted subsampling since the input dataset was already well-balanced. The query dataset remained unchanged.

A CellTypist model was then trained on the sketch to transfer its cell types to the T-related cells in the query dataset, which included Activated CD4 T, Th1, Tfh, CD8 T, cycling gd T, Tcm, gd T, Th 17, and Treg. For comprehensive assessment, the other six methods were employed to produce sketches and perform label transfer using the same pipeline. The performance of transfer learning was evaluated by examining the discrepancies between the transferred annotations and the initial annotations in the query dataset. Dot plots in Fig. S3 and S4 illustrate the comparison between the annotation alignment obtained using the full reference data from the tutorial and those derived from the seven sketches.

From these plots, it is evident that scValue exhibited the closest alignment to the original one, whereas the other methods showed greater tendencies for misassignment:

- Uniform displayed a higher proportion of CD8 T cells being incorrectly predicted as SELL+ CD4 T, and did not transfer any cells labelled gd T from the reference sketch to the query dataset.
- GeoSketch had a larger proportion of CD8 T cells misclassified as Activated CD4 T, and Treg cells were also more frequently misclassified as Activated CD4 T.
- Hopper contained incorrect predictions for the majority of Activated CD4 T cells.
- Sphetcher failed to predict any Activated CD4 T cells.
- KH misclassified a noticeable proportion of Activated CD4 T as Activated CD8 T.
- scSampler misclassified a large proportion of Activated CD4 T as CD8 Tmem and a fraction of CD8 T as SELL+ CD4 T.

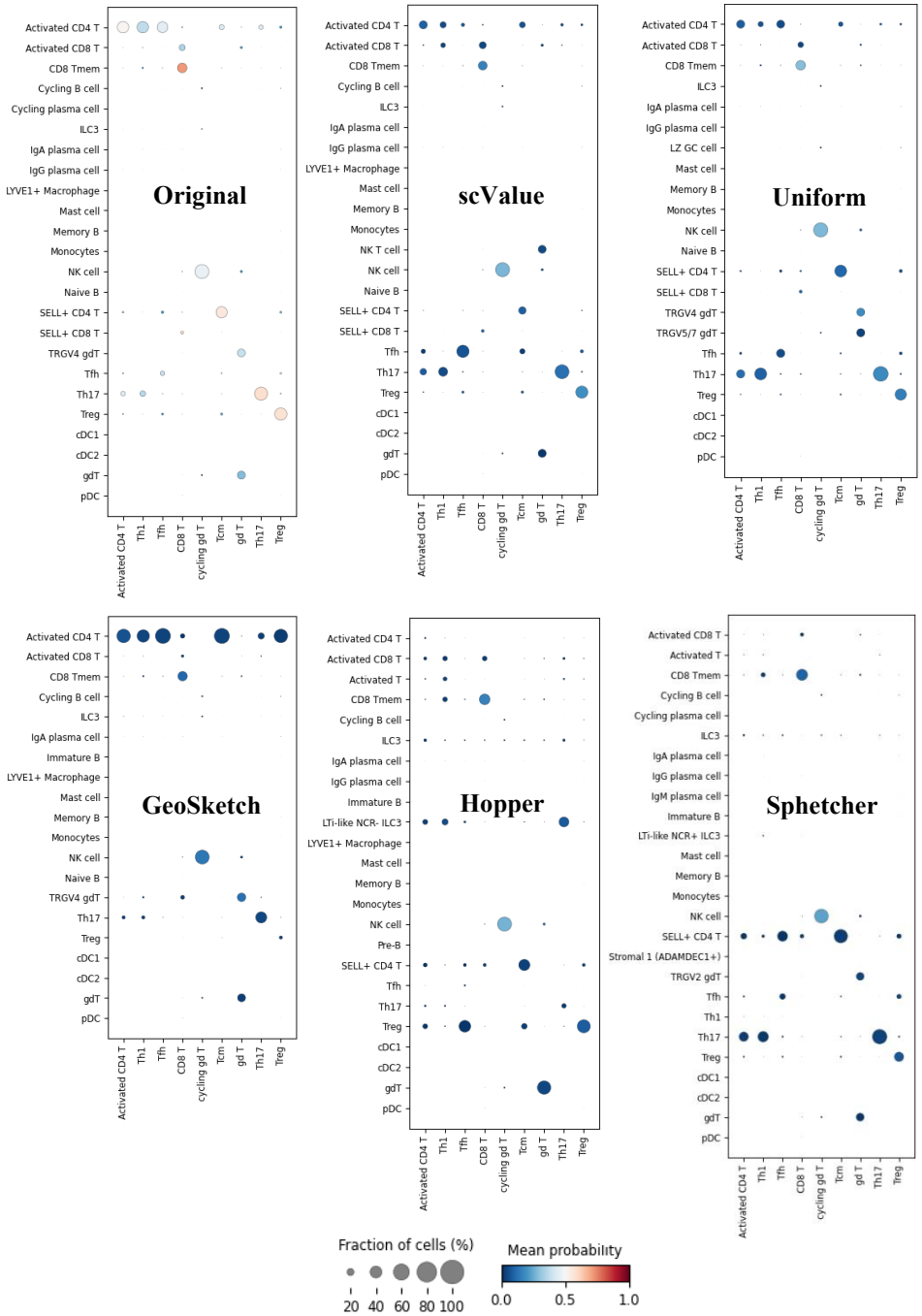


Figure S3: The original dot plot and the dot plots for sketches obtained using scValue, Uniform, GeoSktech, Hopper and Sphetcher in the CellTypist label transfer learning task to transfer cell types in the reference data to T-related cells in the query data.

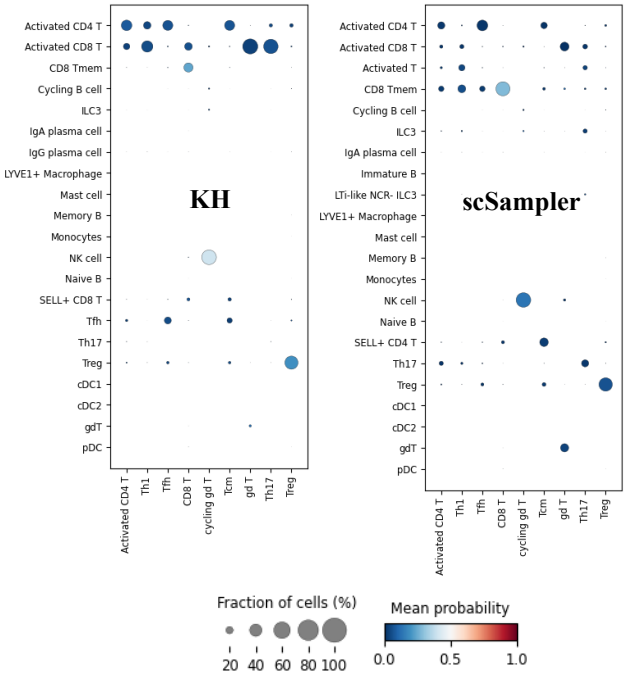


Figure S4: Dot plots for sketches obtained using KH and scSampler in the CellTypist label transfer learning task to transfer cell types in the reference data to T-related cells in the query data.

S6 Application: label harmonisation by CellHint

This experiment was carried out according to the tutorial "Using CellHint for cell type harmonisation" available at: https://colab.research.google.com/github/Teichlab/cellhint/blob/main/docs/notebook/cellhint_tutorial_harmonisation.ipynb.

Following instructions in the tutorial, we utilised the 200k Spleen dataset (Dominguez Conde, et al., 2022) encompassing cells from four independent studies. From the dataset, scValue was employed to sketch a 10% subset that was subsequently fed into CellHint to learn the relationships between T cell-related types across the different studies. Again, the other six methods were also applied to create sketches and perform label harmonisation using the same pipeline. The performance of label harmonisation was evaluated by investigating the learned relationships between diverse cell types across the four studies. Fig. S5 and S6 depict the relationships for sketches produced by the seven methods in the form of tree plots. They are compared with the original tree plot constructed from the full dataset.

The results indicate that scValue achieved the best performance in preserving the various relationships depicted in the original tree plot, while the other methods exhibited varying degrees of inconsistencies, including the introduction of incorrect and/or additional relationships:

- Uniform resulted in a reordering of datasets within the tree structure and led to a loss of relationships involving the CD8-positive, alpha-beta memory T cell and T Cell Spleen.
- GeoSketch incorrectly related the Tnaive/CM CD8 to T_CD4 naive.
- Hopper merged Trm_Tgd and Tgd_CRTAM+ into T_CD8_gd.
- Sphetcher altered the dataset order within the tree structure and also falsely linked Tnaive/CM CD8 to T_CD4 naive.
- KH similarly led to a change of dataset ordering within the tree structure and wrongly related Tnaive/CM CD8 to T_CD4 naive
- scSampler introduced several additional relationships: Tnaive/CM_CD4_activated and T_CD4_conv were incorrectly linked; Trm_Tgd and Tgd_CRTAM+ were combined into T_CD8_gd; Trm_gut_CD8 and Trm/em_CD8 were merged into T_CD8_activated.

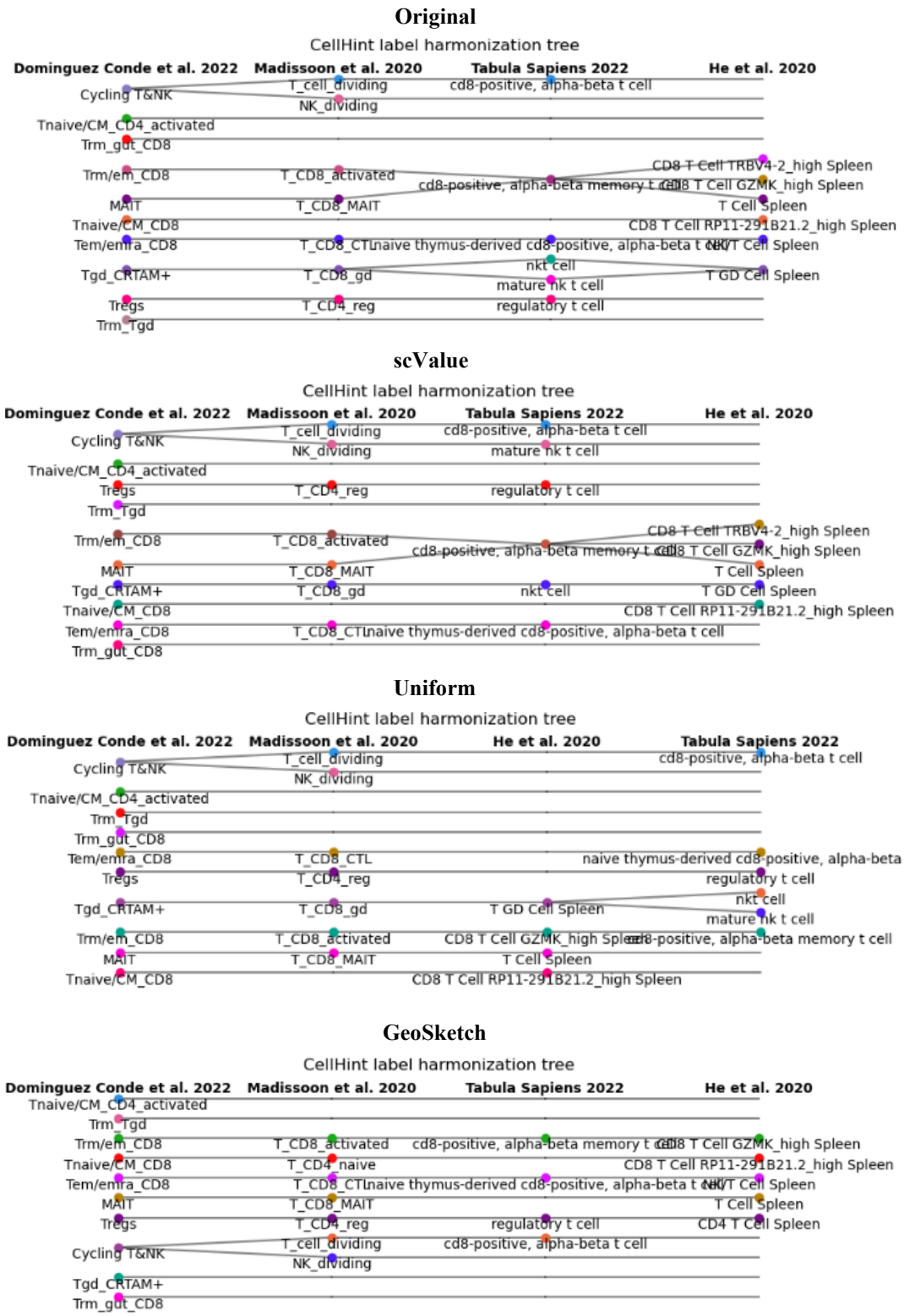


Figure S5: The original tree plot and the tree plots for sketches obtained using scValue, Uniform, and GeoSketch in the CellHint label harmonisation task to learn the relationships between various T-related cell types across four studies.

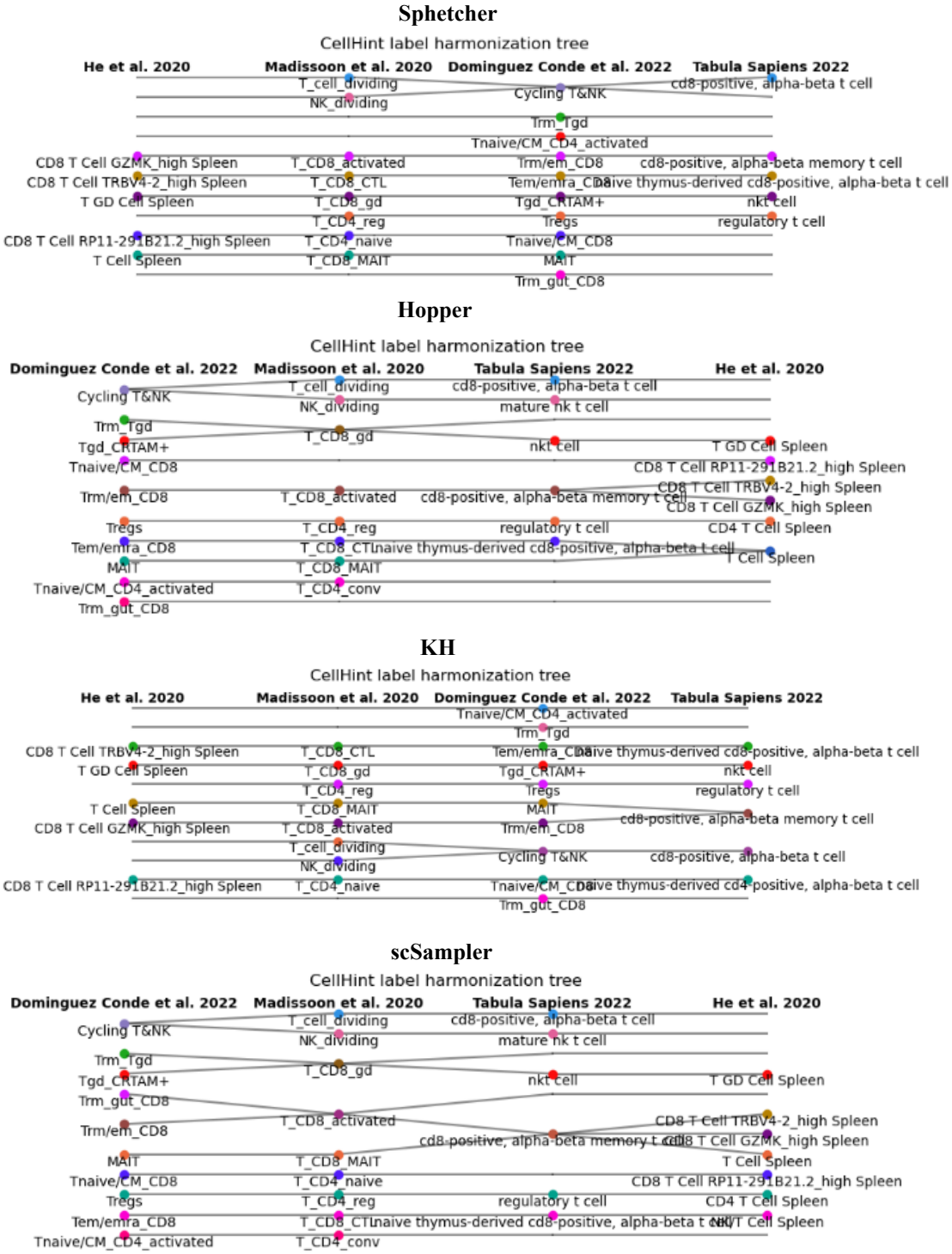


Figure S6: The tree plots for sketches obtained using Sphetcher, Hopper, KH, and scSampler in the CellHint label harmonisation task to learn the relationships between various T-related cell types across four studies.

References

Abdelaal, T., et al. A comparison of automatic cell identification methods for single-cell RNA sequencing data. Genome Biol 2019;20(1):194.

Baskaran, V.A., et al. Distribution-based sketching of single-cell samples. In, Proceedings of the 13th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics. 2022. p. 1-10.

Chen, J., et al. Transformer for one stop interpretable cell type annotation. Nat Commun 2023;14(1):223.

Cormen, T.H., et al. Introduction to algorithms. Cambridge, Massachusetts: MIT press; 2009.
DeMeo, B. and Berger, B. Hopper: a mathematically optimal algorithm for sketching biological data. Bioinformatics 2020;36(Suppl_1):i236-i241.
Do, V.H., Elbassioni, K. and Canzar, S. Sphetcher: Spherical Thresholding Improves Sketching of Single-Cell Transcriptomic Heterogeneity. iScience 2020;23(6):101126.
Dominguez Conde, C., et al. Cross-tissue immune cell analysis reveals tissue-specific features in humans. Science 2022;376(6594):eabl5197.
Du, Z.H., et al. scPML: pathway-based multi-view learning for cell type annotation from single-cell RNA-seq data. Commun Biol 2023;6(1):1268.
Elmentaite, R., et al. Cells of the human intestinal tract mapped across space and time. Nature 2021;597(7875):250-255.
Hie, B., et al. Geometric Sketching Compactly Summarizes the Single-Cell Transcriptomic Landscape. Cell Syst 2019;8(6):483-493 e487.
James, K.R., et al. Distinct microbial and immune niches of the human colon. Nat Immunol 2020;21(3):343-353.
Lotfollahi, M., et al. Mapping single-cell data to reference atlases by transfer learning. Nat Biotechnol 2022;40(1):121-130.
Ma, F. and Pellegrini, M. ACTINN: automated identification of cell types in single cell RNA sequencing. Bioinformatics 2020;36(2):533-538.
Song, D., et al. scSampler: fast diversity-preserving subsampling of large-scale single-cell transcriptomic data. Bioinformatics 2022;38(11):3126-3127.
Tabula Muris, C. A single-cell transcriptomic atlas characterizes ageing tissues in the mouse. Nature 2020;583(7817):590-595.
Xu, C., et al. Probabilistic harmonization and annotation of single-cell transcriptomics data with deep generative models. Mol Syst Biol 2021;17(1):e9620.