Efficient Sketch-Based Analysis of Single-Cell RNA-Seq Datasets Using Geometric Sketching.

Arth Banka

abanka@andrew.cmu.edu

1 Introduction

The advent of single-cell RNA sequencing (scRNA-seq) has enabled transcriptomic profiling at unprecedented resolution, allowing researchers to study cellular heterogeneity across tissues, diseases, and developmental stages (3). However, the explosive growth in the size of scRNA-seq datasets—now often exceeding millions of cells—poses substantial computational challenges for downstream analyses such as clustering, visualization, and trajectory inference. These tasks typically scale poorly with the number of cells, both in memory and runtime, creating a need for efficient data reduction strategies that preserve key biological signals.

To address this issue, recent work has focused on sketching methods: algorithms that extract a small, representative subset of cells (a "sketch") from the full dataset. A good sketching method should maintain transcriptional diversity, capture rare cell populations, and enable downstream analyses with minimal loss of information. Several strategies have been proposed to this end. Geometric sketching, for instance, selects cells to evenly tile the data manifold in PCA space using farthest-point sampling, showing strong performance in capturing rare and diverse populations (1). Other approaches include k-means clustering to select medoids (5), kernel herding for distributional preservation (6), and random hashing-based selection as in the FiRE algorithm for rare cell enrichment (7).

Despite their strengths, existing methods often involve trade-offs between efficiency, complexity, and biological fidelity. Geometric sketching captures outliers well but can be computationally intensive; kernel herding aims to match data distributions but may miss small populations; and uniform or hash-based sampling are fast but risk under-representing rare types.

In this work, we introduce a novel **PCA-guided sketching** approach that leverages the variance structure of the data to inform sampling. Inspired by statistical leverage scores used in randomized linear algebra, our method computes each cell's importance based on its contribution to the top principal components. This allows us to prioritize high-variance, potentially rare cells without explicitly relying on clustering or distance calculations. We evaluate PCA-guided sketching against five existing techniques—uniform sampling, geometric sketching, k-means medoids, kernel herding, and hashing—on the 68k PBMC dataset (3), assessing their performance in terms of rare cell retention, cluster diversity, and computational efficiency.

By systematically benchmarking these methods, we aim to characterize the trade-offs between biological fidelity and runtime and to demonstrate that PCA-guided sampling offers a practical and principled solution for scalable single-cell data analysis. Dimensionality reduction and visualization are performed using UMAP (4), and comparative metrics are analyzed across multiple sketch sizes.

2 Methods

2.1 Sketching Strategies Overview

Let $X = \mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N$ be the full set of N cell expression profiles (after preprocessing), where each $\mathbf{x}_i \in \mathbb{R}^d$ is a d-dimensional gene expression vector. A *sketch* is a subset $S \subset X$ of size $n \ll N$ that is intended to "represent" the original dataset. We briefly describe the sketching strategies compared in this study, with formal definitions where applicable:

Uniform random sampling. The simplest approach selects n cells uniformly at random without replacement. Each cell has equal selection probability $P(i \in S) = \frac{n}{N}$. This method is unbiased in that it preserves population proportions in expectation. Uniform sampling provides a baseline against which we compare more informed strategies.

Geometric sketching. Geometric sketching (*GeoSketch*) aims to preserve the geometry of the data distribution in gene expression space. In practice, this is implemented by first embedding cells into a lower-dimensional space (e.g. top principal components) and then selecting a subset that covers this space evenly. Hie *et al.*(1) approximate the high-dimensional expression space by dividing it into equally sized grid cells (hypercubes) and then picking one cell from each grid region. In our implementation, we follow the approach of *geosketch*: we compute a lower-dimensional embedding (k PCs) and use a greedy farthest-point sampling to pick n points that are roughly evenly dispersed. Geometric sketching thus ensures that densely populated regions don't overwhelm the sample and that sparse regions (potentially containing rare cell types) are represented.

k-means clustering selection. Clustering-based sketching involves grouping cells into k clusters and selecting one (or a fixed number) from each cluster. We use k-means clustering to partition the data into $k = n$ clusters, aiming to minimize the within-cluster variance:

$$\min_{C_1,\ldots,C_n} \sum_{j=1}^{n} \sum_{\mathbf{x}_i \in C_j} \|\mathbf{x}_i - \boldsymbol{\mu}_j\|^2$$

where $\boldsymbol{\mu}_j$ is the centroid of cluster C_j. Once clusters are obtained, we choose either the centroid (if it corresponds to an actual cell) or the cell closest to the centroid (medoid) from each cluster to include in S. This approach guarantees that most major cluster or cell type in the data is represented in the sketch (assuming n is at least the true number of cell states). The method's performance depends on the clustering algorithm's parameters and assumptions (e.g. cluster number k).

Kernel herding. Kernel herding is a deterministic sampling technique, designed to select representative "super-samples" that approximate the overall data distribution (often in reproducing kernel Hilbert space). The idea is to iteratively pick points such that their average is as close as possible to the true mean of the full dataset in a feature space defined by a kernel(6). Formally, suppose $\phi(\mathbf{x})$ is a (possibly nonlinear) feature map and let $\mu = \frac{1}{N}\sum_{i=1}^{N} \phi(\mathbf{x}_i)$ be the mean embedding of the full data. Kernel herding builds S iteratively: start with $S_0 = \emptyset$ and residual $r_0 = \mu$. At each step $t = 1, \ldots, n$, select

$$x_{j_t} = \arg\max_{x_j \in X} \langle \phi(x_j), r_{t-1} \rangle$$

and update $r_t = r_{t-1} - \frac{1}{t}\phi(\mathbf{x}_{j_t})$. Intuitively, it greedily adds the cell that most improves the approximation of the true mean. In the end, the selected set $S = \mathbf{x}_{j_1}, \ldots, \mathbf{x}_{j_n}$ has the property that the average feature vector $\frac{1}{n}\sum i \in S\phi(\mathbf{x}_i)$ is close to μ. In our case, we use an RBF kernel $K(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\gamma|\mathbf{x}_i - \mathbf{x}_j|^2)$ on the PCA space (with a chosen γ parameter) so that $\phi(\mathbf{x})$ lives in an implicit high-dimensional space. We include it as a method with strong theoretical grounding in sample approximation.

Hashing-based sampling. In this approach, each cell's gene expression vector is first converted into a compact numerical representation by computing a hash over its expression values. Specifically, for each cell, we concatenate the gene expression values into a single string (optionally after rounding for numerical stability), and compute an MD5 hash. The resulting hash is interpreted as an integer. We then sort all cells based on their hash values and select the top n cells with the smallest hashes to form the sketch.

This method provides a deterministic, data-driven sampling mechanism that maps transcriptional content while maintaining simplicity and efficiency. It is expected to perform very similar to random sampling because the hashing is agnostic to similar expression values. While it does not explicitly optimize for diversity or density, it offers a fast and reproducible baseline that indirectly captures variation in cell profiles through the hash-based ordering.

2.2 PCA-Based Sketching (Novel Contribution)

PCA-Guided Sampling (Proposed Method)

We propose a novel and simple sketching strategy that prioritizes cells based on their overall variance across principal components. The method operates by computing a variance score for each cell after PCA transformation, and then sampling cells in proportion to these scores.

Concretely, given a matrix $X \in \mathbb{R}^{N \times d}$ of gene expression values, we perform PCA to obtain a reduced representation $Z \in \mathbb{R}^{N \times k}$, where each row $\mathbf{z}_i$ corresponds to the PCA coordinates of cell i. Instead of computing leverage scores per se, we approximate each cell's contribution to total variance using the squared norm of its vector in PCA space:

$$w_i = \|\mathbf{z}_i\|^2$$

These scores are then normalized into sampling probabilities:

$$p_i = \frac{w_i}{\sum_{j=1}^{N} w_j},$$

and we sample n cells without replacement using p_i as selection probabilities.

The full algorithm is shown below:

Algorithm 1 PCA-Guided Sampling

Require: PCA matrix $Z \in \mathbb{R}^{N \times k}$, sketch size n
Ensure: Index set $S \subset \{1, \ldots, N\}$, with $|S| = n$
1: Compute $w_i = \|\mathbf{z}_i\|^2$ for each row $\mathbf{z}_i \in Z$
2: Normalize weights: $p_i = \frac{w_i}{\sum_j w_j}$
3: Sample n unique indices from $\{1, \ldots, N\}$ with probabilities $\{p_i\}$ **return** Sampled index set S

Interpretation: This method emphasizes cells with high variance across PCA dimensions, which often correspond to rare or transcriptionally distinct states. Unlike uniform sampling, which treats all cells equally, or geometric sketching, which seeks global spatial coverage, our PCA-guided strategy implicitly favors biologically informative or outlier-like cells that dominate variation. Empirically, this leads to strong rare cell capture and diversity preservation while remaining computationally lightweight: the method requires only PCA (which can be performed efficiently) and weighted sampling in $\mathcal{O}(N)$ time.

3 Implementation Details

We implemented all sketching methods in a Python environment and applied them to a real single-cell dataset to compare their performance. Below we describe key preprocessing steps, dataset characteristics, parameter choices, and any necessary implementation notes for each method:

Data preprocessing and clustering: We used the publicly available **10x Genomics 68k PBMC** dataset(3) as the testbed. This dataset contains $\sim 68{,}000$ peripheral blood mononuclear cells (PBMCs) from a healthy donor, with gene expression quantified by unique molecular counts. We applied standard quality control filters (removing low-quality cells with very few detected genes, and filtering out genes not expressed in any cell) and log-normalized the expression values. We then performed an initial PCA on the full dataset to reduce dimensionality, retaining $k = 50$ principal components for downstream analysis. For reference and evaluation (not for the sketch selection itself, except in PCA-based method), we also ran a clustering on the full 68k dataset: we constructed a k-nearest-neighbor graph in the 50-dimensional PCA space, and applied the Leiden community detection algorithm to identify clusters of cells. This yielded a set of clusters that correspond to distinct cell subtypes in the PBMC population (e.g., T cell subsets, B cells, NK cells, monocytes, dendritic cells). These clusters and their sizes serve as ground truth to evaluate how well each sketch captures the cellular heterogeneity.

Dataset and sketch sizes: The full dataset size is N = 68000 cells. We experimented with sketch sizes ranging from $n = 100$ up to $n = 2000$ for plotting how time taken changes. For the UMap visualization and quantifying the biological fidelity, sketch size of $n = 1000$ was used. The lower end ($n = 100$) is an extremely aggressive compression ($< 0.2\%$ of the data), essentially a stress test for the methods to see if they can pick up any rare populations at all with so few cells. The upper end ($n = 2000$) is about 3% of the data; at this size, we expect even uniform sampling to capture most clusters at least once, so it provides a sense of near-asymptotic performance.

Parameter choices: Unless otherwise noted, all methods were applied on the 50-dimensional PCA-transformed data for fairness and efficiency (except uniform which does not use any structure, and hashing which we applied on the raw gene space with random projections):

- **Uniform:** Used Python's random sampling without replacement to pick n cells uniformly.
- **Geometric sketching (GeoSketch):** We used the `geosketch` library implementation by Hie *et al.*(1) on the PCA space. We set the sketch size parameter to n for each run and used default settings otherwise. (Internally, `geosketch` divides the PCA space into grid bins as described, which is efficient for large N.)
- **k-means:** We ran Lloyd's k-means algorithm on the PCA coordinates with $k = n$ clusters. We used the implementation from `scikit-learn`, with a maximum of 100 iterations and 10 random initializations to avoid poor local optima. After clustering, we identified the cell closest to each centroid (Euclidean distance in PCA space) and took those n cells as the sketch. In cases where n exceeded the true number of distinct cell types, k-means naturally produces multiple clusters for one cell type, this this method can sample multiple representatives from large populations and at least one from each small population (as long as the small population forms its own cluster).
- **Kernel herding:** We implemented kernel herding in the 50-D PCA space. We chose an RBF kernel $K(\mathbf{x}, \mathbf{y}) = \exp(-\gamma|\mathbf{x} - \mathbf{y}|^2)$ with $\gamma = 0.5$ (chosen by rough scale of PCA variance so that distances of order a few units produce a moderate kernel value). Starting from the mean of all points' feature maps, we iteratively added cells. Due to the greedy nature, the order in which points are selected is deterministic given the initial μ. We computed inner products $\langle \phi(\mathbf{x}), r_{t-1} \rangle$ efficiently by vectorizing over all candidate points at each step. Herding $n = 1000$ points out of 68k required on the order of 1000×68000 kernel computations, which we mitigated by working in PCA space (50 dims). The resulting set S is returned.
- **Hashing:** In our implementation, each cell's expression profile was hashed to generate a deterministic identifier based on its transcriptomic content. Specifically, we rounded each gene expression vector to a fixed precision and concatenated its values into a string. We then computed an MD5 hash for this string and interpreted the result as an integer. After hashing all cells, we sorted them by their hash values and selected the top n cells as the sketch. This method offers a fast, reproducible way to sample cells based on their content without explicitly computing distances or densities. While it does not guarantee diversity or rare cell capture, it serves as a lightweight and deterministic alternative to random sampling.
- **PCA-based sampling:**After computing the 50-dimensional PCA representation for all cells, we calculated a variance score for each cell as $w_i = \|\mathbf{z}_i\|^2$, where $\mathbf{z}_i$ is the PCA vector for cell i. These scores were normalized into probabilities $p_i \propto w_i$, which we used to sample n cells without replacement. At each step, we selected one cell according to p, removed it, renormalized the probabilities, and repeated until the sketch was complete. Since $n \ll N$, the renormalization effect was minimal and the procedure remained efficient. In practice, sampling with precomputed cumulative probabilities made the method fast and scalable even for large datasets.

4 Results

We evaluated the runtime performance and biological fidelity of the sketching algorithms on the 68k PBMC dataset for sketch size = 1000. The results demonstrate clear differences in the subsets produced by each method and highlight the strengths of each approach.

4.1 Visualization of Sketches in UMAP Space

To understand how well each sketch captures the overall structure of the data, we embedded the full dataset and the sketch points in a 2D UMAP plot (Uniform Manifold Approximation and Projection). Figure 1 shows the UMAP visualization with the full 68k data in the background (gray points) and the sketch points from each method overlaid in color (each subplot corresponds to one method, and sketch points are colored by their full-data cluster identity). We see that all methods broadly cover the main clusters visible in the UMAP: large clusters of cells, are all represented to some extent by colored points in the same areas as the gray background.

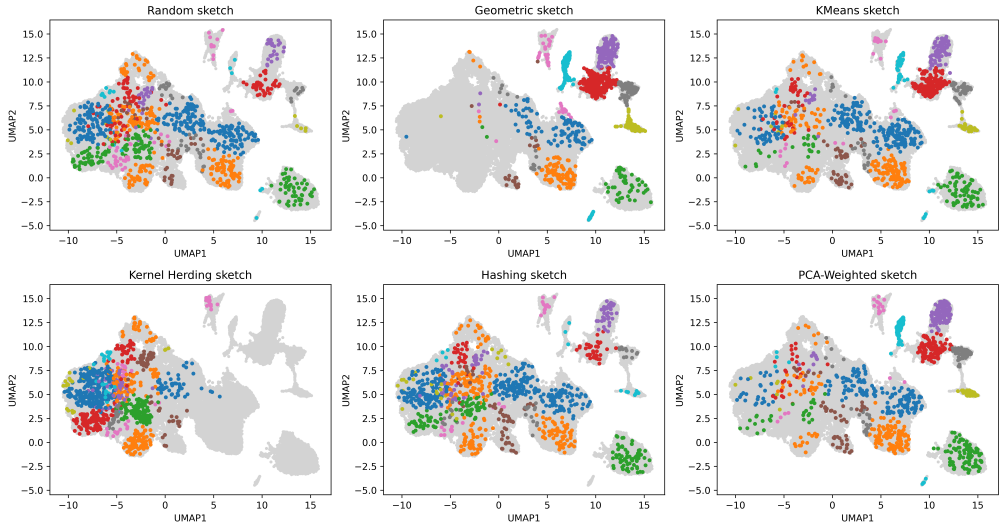


Figure 1: **UMAP visualization of the 68k PBMC data with different sketching methods.** Gray points show all 68k cells in 2D UMAP space. Colored points overlay the cells selected by each sketching method, with colors indicating their cell type cluster.

While the overall structures are preserved, there are notable differences: The uniform and hashing-based samples (Figure 1a and 1e) scatter points roughly uniformly across all gray regions, reflecting the unbiased nature of random selection. This results in a good coverage of large clusters, but some very small clusters have only a handful of colored points or potentially none if unlucky. In our sample, the uniform sketch did happen to include a couple of points from the rare cluster (which appears as a tiny gray island in the UMAP), but not many.

Geometric sketching (Figure 1b) shows a very broad coverage: it places points even at the far extremes of the distribution. The sketch points include alot of the points in the remote gray islands while missing out on large parts of the bigger central continent of data. Visually, geometric sketching "hugs" the perimeter of the data cloud, ensuring even the farthest outliers are captured. This confirms its design goal of preserving global geometry.

The k-means medoids sketch (Figure 1c) also captures all main clusters, but one can observe that within very large clusters, the colored points tend to lie centrally. This is because medoids of large clusters will be near the cluster centroid. k-means did include the rare cluster (one medoid landed in that cluster, since we used $N = 1000$ clusters, more than the number of true clusters), thus it did not completely miss any group. However, it may not sample multiple points from a small cluster, whereas geometric might if that cluster has internal heterogeneity.

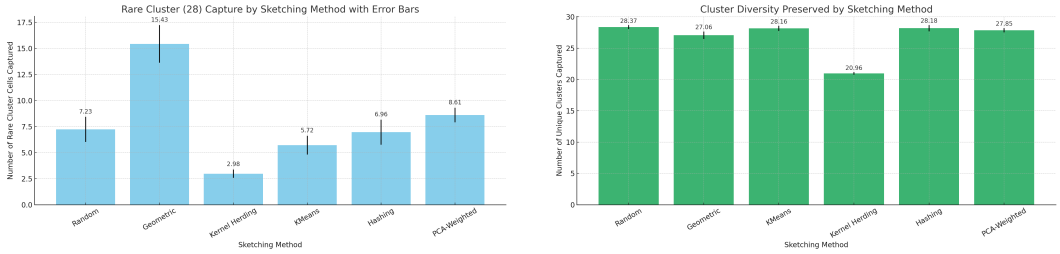
Kernel herding (Figure 1d) yields a sketch that visibly under-represents the rare cluster. In the figure, the rare cluster region (indicated by an arrow) has no colored points from the kernel herding subset, meaning none of the cells from some of the rare clusters were selected. The kernel herding points concentrate in the densest areas, reflecting its aim to match the overall distribution (the densest cluster contributes the most to the RKHS mean, so those points are heavily selected). Consequently, some peripheral populations are entirely missed.

The PCA-guided sketch (Figure 1f) shows a distribution of points somewhat intermediate between uniform and geometric. The PCA-weighted subset includes the rare cluster (several colored points are present in that tiny cluster) and other outlying clusters, demonstrating success in capturing those high-variance extremes. At the same time, it also includes many points from the large central clusters (though slightly fewer proportionally compared to uniform). This indicates that PCA-guided sampling enriched for peripheral clusters but still kept a fair representation of major groups.

Overall, the UMAP visualization qualitatively confirms that geometric and PCA-guided methods are effective at including rare/extreme cells, whereas kernel herding can fail to include those, and uniform/k-means lie in between. Next, we quantify these observations.

4.2 Retention of Rare Populations and Cluster Diversity

We quantified how well each sketch retained the rarest cell population and the overall diversity of clusters, as defined by the full-data clustering. Figure 2a plots the number of cells from the rare cluster that were captured in each method's sketch. The total size of the rarest cluster in the full data was 193. Figure 2b shows the number of distinct full-data clusters (out of 29 total) that have at least one representative in the sketch. This was simulated over a 100 times and the results are shown below.



(a) Rare cluster capture (193 cells total).

(b) Cluster diversity (12 total clusters).

Figure 2: **Biological fidelity:** Comparison of sketching methods in terms of rare cluster recovery and cluster representation. This was simulated 100 times and the error bars are shown as well.

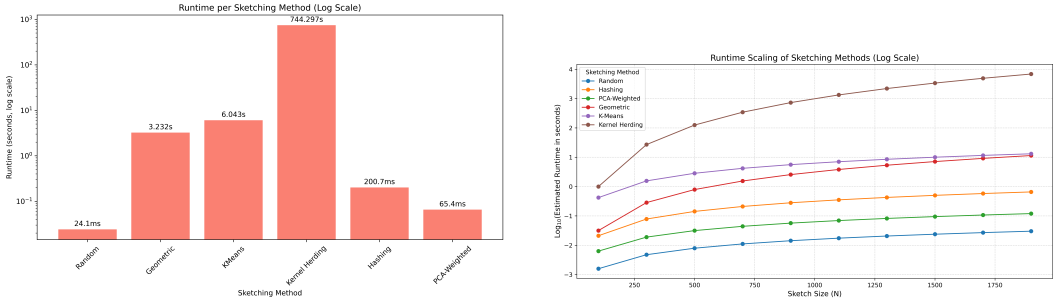
As shown in Figure 2a, **geometric sketching** was the top performer in recovering the rare cluster. This is expected, since the rare cluster is an extreme outlier in expression space and geometric sketching will aggressively pick those outliers. The **PCA-guided** method was the second best but not as superior to the other methods as geometric sketching. This demonstrates that weighting by variance successfully boosts the inclusion of rare, highly distinct cells too. **Uniform** and **hash-based** sampling picked a comparable number of cells from the rare cluster which is expected since the methods are very similar in principle. The **k-means** medoid approach was also consistently worse than uniform sampling when picking cells from the rare cluster as it is possible that the rare cluster was either underrepresented or net well captured in the K-means representation. Finally, as foreshadowed, **kernel herding** picked very few rare cluster cells. This starkly highlights the weakness of a purely distribution-matching approach when faced with an extremely small but outlying group.

Figure 2b shows the average number of unique clusters captured by each method across multiple runs, with error bars denoting standard deviation. Most methods—including PCA-weighted, K-Means, Hashing, and Random sampling—consistently captured over 28 of the 29 clusters, indicating strong coverage of transcriptional diversity. Geometric sketching slightly underperformed in this regard, averaging 27.06 clusters. Kernel Herding performed the worst, recovering only 21 clusters on average, highlighting its tendency to concentrate on dense regions and miss peripheral or rare populations. These results underscore that PCA-based and geometric methods strike a favorable balance between rare cell capture and diversity preservation.

In summary, the PCA-guided method performed the second best in terms of retaining rare cells and preserving all cell type clusters after geometric sketching. Both outperformed uniform sampling in rare cell capture (even though random did surprisingly capture the cluster at least once, it got far fewer of its cells). Kernel herding, while preserving proportions, clearly underperformed in capturing the full diversity of the dataset.

4.3 Computational Efficiency and Scaling

We recorded the runtime of each method on the 68k dataset and also explored how runtime scales with sketch size. Note that $N = 68,000$ here refers to the size of the entire dataset. Figure 3a compares the wall-clock time for each method to produce a 1000-cell sketch from 68k cells. Figure 3b presents a plot of the log of runtime versus number of cells in sketch of methods to illustrate scaling.



(a) Runtime (log scale) for sketch size of $n = 1000$ cells.

(b) Log-log runtime scaling as a function of $n =$ sketch size.

Figure 3: **Computational efficiency:** Execution time and scalability of each sketching method.

Figure 3a shows the runtime (on a log scale) for each method for sketch size of $n = 1000$. As expected, **Random sampling** was the fastest, taking under 30 milliseconds. **Hashing-based** and **PCA-guided sampling** were also highly efficient, with runtimes below 1 second. The additional overhead for PCA is offset by its high impact on biological fidelity.

Geometric sketching and k**-Means** were slower, taking a few seconds each due to iterative selection or clustering steps. The most computationally expensive method was **Kernel Herding**, which required over 12 minutes (744 seconds), making it impractical for large-scale datasets despite its theoretical appeal.

To understand how runtime scales with sketch size, we ran all methods on varying sketch sizes from 100 to 2000. Figure 3b plots the log runtime against sketch size. **Random**, **Hashing**, and **PCA-guided** methods showed near-linear scaling, confirming their suitability for large N. **Geometric sketching** and k**-Means** also scaled well, although with slightly higher slopes. **Kernel Herding**, however, scaled poorly due to its quadratic complexity, with runtime increasing steeply with n. Detailed time complexity analysis of each method has been further performed in the supplementary section which can be found at the end of the report before references.

These results underscore the practical trade-offs between fidelity and efficiency. While geometric and PCA-guided methods achieve high biological performance, PCA-guided sampling achieves this with significantly lower computational cost, making it a compelling alternative to slower but accurate methods.

5 Discussion and Conclusion

In this study, we presented a comparative analysis of sketching methods for large-scale single-cell RNA-seq data, with a particular focus on a novel PCA-guided sketching approach. Using the 68k PBMC dataset, we demonstrated that sketching can significantly reduce data size while retaining critical biological features such as rare cell types and global diversity. However, the choice of sketching strategy directly affects the fidelity and efficiency of downstream analyses.

The proposed **PCA-guided method** performed comparably to geometric sketching in preserving rare cell populations and overall diversity, while being significantly faster and easier to implement. By sampling in proportion to the variance explained in the top principal components, this method naturally emphasizes both rare and diverse cells, balancing between overrepresentation of outliers and inclusion of common populations. Importantly, its implementation is lightweight — requiring only a PCA transformation and weighted sampling — and scalable to large datasets.

Method Comparisons

- **Random / Hashing-based:** Fastest and simplest to implement. While they maintain population proportions well, they risk under-sampling rare populations, as seen in our results. They are most suitable for bootstrapping or exploratory visualization.
- **Geometric sketching:** Strong at capturing outliers and peripheral cells, with excellent rare cell recovery. However, it may distort abundance information and is slower for very large datasets.
- **k-Means medoids:** Guarantees inclusion of distinct clusters (assuming enough clusters are chosen), but may include only one cell from rare types. Performance depends on the choice of k and may not scale well.
- **Kernel Herding:** Aims to match the data distribution but suffers from high runtime and can miss rare clusters entirely. Its theoretical value is offset by poor scalability and practical limitations.
- **Hashing-based:** Offers reproducible random sampling, behaving similarly to uniform selection. Useful for fast downsampling but limited in its ability to capture rare or informative cells.
- **PCA-guided (ours):** Combines the strengths of variance-based prioritization with practical speed. It achieves high rare cell recovery and good cluster representation, at a much lower cost than geometric or kernel-based methods.

Practical Takeaways

All sketching methods dramatically reduced runtime for clustering and visualization, enabling efficient analysis of large datasets. PCA-guided and geometric sketches enriched rare cell types, which can help in discovery-focused workflows. In contrast, kernel herding preserved frequency distributions but failed to represent rare types. The choice of method should therefore depend on the downstream goal: enrichment versus estimation.

Future Directions

There are several avenues to explore building on this PCA-guided approach. One idea is to make it iterative: for instance, select some cells, then recompute PCA on the remaining or on the selected set to find if additional variance can be captured, akin to a "multi-round" selection. This starts to resemble algorithms like Hopper (which iteratively add points and allow refinement). Another extension is to incorporate nonlinear structure by using kernel PCA or autoencoder embeddings instead of linear PCA, then apply a similar variance-weighted sampling. We could also consider hybrid methods, such as first ensuring each known cluster is represented (like k-means or clustering ensures that), and then using PCA-weights within each cluster to pick additional points proportional to internal variance. Additionally, sketching strategies can be applied not just to single-cell profiling but to other domains with large high-dimensional data (e.g., choosing representative images or trajectories); adapting PCA-guided sketching there would be interesting.

Finally, as dataset sizes continue to grow (with millions of cells now common in atlas-scale projects), scalability will remain a concern. Geometric methods might need approximation to cope with millions of points. Our results underscore that any method with worse than linear complexity (like naive kernel herding) will become infeasible. Therefore, focusing on efficient approximate algorithms will be key to keep sketching tools useful in practice.

Conclusion

We have introduced a fast, effective, and principled sketching method for large-scale single-cell RNA-seq data. PCA-guided sampling offers a compelling compromise between biological fidelity and computational efficiency, preserving key transcriptional diversity while remaining scalable. Our approach enriches the suite of tools available for single-cell analysis, making high-resolution exploration feasible on ordinary hardware without compromising insight.

Supplementary Section: Time Complexity of Sketching Methods

We analyze the time complexity of each sketching method in terms of the number of cells in the dataset N (e.g., 68,000), the sketch size n (e.g., 1,000–5,000), and the feature dimension d (e.g., PCA space, typically 50).

- **Uniform Sampling:** Randomly selects n cells from N without replacement. Time complexity: $\mathcal{O}(n)$. Efficient sampling algorithms make this method extremely fast.
- **Hash-Based Sampling:** Hashes cell identifiers or expression summaries, sorts them, and selects the top n. Time complexity: $\mathcal{O}(N \log N)$, dominated by sorting.
- **PCA-Weighted Sampling (Proposed):** Computes PCA ($\mathcal{O}(Nd^2)$) and uses squared norm of each row in PCA space as a sampling weight. Sampling n cells without replacement using precomputed weights gives a total time of $\mathcal{O}(Nd + n \log N)$.
- **Geometric Sketching:** After PCA, uses farthest-point sampling (FPS) to iteratively select diverse points. Time complexity: $\mathcal{O}(Nd + nN)$ due to n distance computations over all N points.
- **K-Means Medoids:** Uses k-means (with $k = n$) and selects medoids per cluster. With t iterations, the total time is $\mathcal{O}(Ndtn)$. MiniBatch variants reduce this cost.
- **Kernel Herding:** Greedy selection to match mean embedding in RKHS. Each of n steps involves N kernel evaluations, yielding $\mathcal{O}(nNd)$. This is the most computationally expensive method.

Table 1: Summary of Time Complexities for Sketching Methods

Method	**Time Complexity**	**Dominant Factor**
Uniform Sampling	$\mathcal{O}(n)$	Direct random sampling
Hash-Based Sampling	$\mathcal{O}(N \log N)$	Sorting hash values
PCA-Weighted Sampling	$\mathcal{O}(Nd + n \log N)$	PCA + weighted sampling
Geometric Sketching	$\mathcal{O}(Nd + nN)$	FPS distance updates
K-Means Medoids	$\mathcal{O}(Ndtn)$	MiniBatch clustering
Kernel Herding	$\mathcal{O}(nNd)$	Kernel evaluations

References

[1] Hie, B., Cho, H., DeMeo, B., Bryson, B., & Berger, B. (2019). *Geometric sketching compactly summarizes the single-cell transcriptomic landscape*. *Cell Systems*, 8(6): 483–493.e7. doi:10.1016/j.cels.2019.05.003.

[2] Hao, Y., Stuart, T., Kowalski, M. H., Choudhary, S., Hoffman, P., Hartman, A., *et al.* (2023). *Dictionary learning for integrative, multimodal and scalable single-cell analysis*. *Nature Biotechnology*, 41(3): 359–369. doi:10.1038/s41587-022-01432-8. (Seurat v5 leverage score sketching)

[3] Zheng, G. X. Y., Terry, J. M., Belgrader, P., *et al.* (2017). *Massively parallel digital transcriptional profiling of single cells*. *Nature Communications*, 8: 14049. doi:10.1038/ncomms14049. (Fresh 68k PBMCs dataset)

[4] McInnes, L., Healy, J., & Melville, J. (2018). *UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction*. *arXiv*:1802.03426.

[5] Arthur, D. & Vassilvitskii, S. (2007). *k-means++: The advantages of careful seeding*. In *Proc. of the 18th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pp. 1027–1035. (K-means clustering initialization)

[6] Chen, Y., Welling, M., & Smola, A. (2012). *Super-samples from kernel herding*. *arXiv*:1203.3472. (Kernel herding algorithm for selecting representative samples)

[7] Jindal, A., Gupta, P., Jayadeva, & Sengupta, D. (2018). *Discovery of rare cells from voluminous single cell expression data*. *Nature Communications*, 9(1): 4719. doi:10.1038/s41467-018-07234-6. (FiRE algorithm for rare cell detection)