CAMP: Coreset Accelerated Metacell Partitioning enables scalable analysis of single-cell data

Danrong Li[1], Young Kun Ko*[1], and Stefan Canzar*[2]

[1]Department of Computer Science and Engineering, The Pennsylvania State University, University Park, PA 16802, United States
[2]Faculty of Informatics and Data Science, University of Regensburg, Regensburg, Germany

Abstract

Scaling metacell inference to atlas-level single-cell datasets demands algorithms that are both computationally efficient and geometrically faithful. We introduce CAMP (Coreset Accelerated Metacell Partitioning), a metacell framework that preserves the intrinsic structure of the cellular manifold while enabling scalable analysis of millions of cells. CAMP leverages coreset-based sampling to construct a small, weighted subset of representative cells that approximates the full dataset with provable geometric guarantees. This formulation transforms metacell construction into a coreset inference problem, reducing runtime and memory complexity by up to an order of magnitude without loss of accuracy.

Through extensive experiments, we show that CAMP produces metacells that are compact, well-separated, and biologically coherent, achieving performance on par with or exceeding existing methods including MetaCell, SuperCell, SEACells, and MetaQ. By combining theoretical efficiency with empirical robustness, CAMP establishes coreset acceleration as a principled foundation for scalable, high-fidelity metacell inference in single-cell transcriptomics.

*Corresponding authors

1 Introduction

Single-cell RNA sequencing (scRNA-seq) technologies now profile hundreds of thousands of cells in a single experiment, offering an unprecedented view of cellular heterogeneity during development, immunity, and disease [1][2][3][4]. Yet this level of resolution introduces substantial redundancy and noise [5]: many cells exhibit near-identical expression profiles with high sparsity [6][7], while technical variability obscures underlying biological structure. Analyses that operate directly on the single-cell matrix therefore face severe statistical and computational challenges [6][8].

To address these challenges without introducing excessive smoothing [9][10] or generating false positives [11][7], which are known issues of imputation based approaches, a common strategy is to use metacell frameworks [10]. These methods summarize transcriptionally similar cells into coherent groups that represent stable and biologically meaningful expression states, typically obtained by averaging their expression profiles. Over the past few years, several computational strategies have emerged to implement this idea. MetaCell [10] partitions single-cell data into metacells by constructing a balanced k-nearest neighbor (kNN) graph and iteratively refining densely connected subgraphs through resampling, outlier detection, and rebalancing to ensure homogeneous clusters. MetaCell2 [12] extends this approach with a divide-and-conquer strategy that partitions large datasets into manageable subsets and replaces stochastic resampling with a deterministic stability score, improving scalability and sensitivity to rare cell types. SuperCell [5] is an unsupervised framework that constructs metacells by iteratively merging highly similar cells in a kNN graph built using Euclidean distance. It models the data as a graph in which each cell is represented as a singleton node, and then applies the walktrap community detection algorithm to progressively merge densely connected nodes until the desired number of metacells is reached. SEACells [13] begins by constructing a kNN graph using Euclidean distance and converting it into an adaptive Gaussian kernel, which serves as a similarity matrix in a high-dimensional feature space. Archetypal analysis is then performed on this kernel to identify representative "extreme" transcriptional states, where the number of archetypes corresponds to the desired number of metacells. MetaQ [8] formulates metacell construction as a vector quantization problem over single-cell expression profiles, using a deep autoencoder with an encoder, a learnable codebook, and a decoder that reconstructs gene expression to guide the quantization.

Despite their success, existing frameworks share fundamental limitations. Most rely on graph or manifold formulations that require constructing large matrices and performing iterative optimization, leading to beyond superlinear computational cost. They are also sensitive to hyperparameter choices and often depend heavily on GPU resources to remain practical at scale. As single-cell datasets continue to grow into the hundreds of thousands and millions of cells, there is an increasing need for methods that achieve both computational scalability and geometric fidelity without requiring specialized hardware. Even after we applied substantial implementation-level optimizations, SEACells continued to produce out-of-memory errors on large inputs, while MetaCell and MetaCell2 exceeded a 48-hour runtime limit when dataset sizes increased. Independent evaluations in [8] confirm similar behavior, with SEACells and MetaCell2 stalling or failing on datasets in the 100,000 to 200,000 cell range. These limitations collectively make existing approaches impractical for performing metacell partitioning on modern large-scale single-cell data.

In order to make metacell partition feasible for large scale datasets, we introduce CAMP (Coreset Accelerated Metacell Partitioning), a framework designed to meet this need. CAMP builds on the archetypal formulation introduced by SEACells but replaces its iterative Frank–Wolfe optimization with a lightweight geometric coreset [14], through which metacells are inferred according to positional proximity. It yields results that are comparable in metacell quality metrics (see Section 3.2) while drastically reducing computational cost. By anchoring analysis on representative subsets rather than the full data matrix, CAMP bridges theoretical ideas from geometric summarization [14] with practical demands in single-cell transcriptomics, providing an efficient and conceptually straightforward foundation for large-scale metacell inference.

2 Methods

CAMP builds on the principles of archetypal analysis [15][16], a convex-hull formulation that identifies a set of extremal points ("archetypes") capturing the boundary of the cellular state space. These archetypes delineate distinct transcriptional states within the manifold and therefore serve as natural centers for metacell formation (Supplementary Figure 1). A key theoretical insight enabling CAMP is Proposition 1 of [17], which shows that a coreset constructed for the k-means objective is also a valid coreset for archetypal analysis, establishing a direct connection between clustering and archetypal inference.

Classical archetypal analysis, including the kernel formulation used in SEACells [13], typically refines archetypes through iterative optimization over the full dataset or its kernel representation, a procedure that becomes computationally prohibitive at atlas scale. CAMP circumvents this bottleneck by selecting a lightweight k-means coreset [14] of size k when inferring k metacells and treating the resulting representative points as archetypes directly. Each cell is then assigned to the archetype that best represents it in the chosen geometric or kernel space in a one-shot setting. This bypasses iterative archetypal refinement entirely while retaining the desirable properties of k-means, which is to minimize within-cluster variance and maximize separation, aligning naturally with biological notions of metacell compactness and distinctness. Moreover, k-means has been extensively studied from a theoretical perspective, particularly through coreset constructions [18][19][20][21][22][23], providing a well-established foundation for scalable and geometry-preserving metacell inference. The specific coreset construction and sampling distribution used in CAMP follow the framework of [14], which we describe in the next section.

2.1 Weighted distribution for sampling CAMP coresets

We define the gene expression matrix as $\mathcal{X} \in \mathbb{R}^{n\times d}$, where n denotes the number of cells and d the number of features (e.g., genes). Each cell is represented as a row vector $\mathcal{X}_i \in \mathbb{R}^{1\times d}$ for $i \in [n]$. Let μ denote the global mean of all cell vectors, and let $d_i = \|\mathcal{X}_i - \mu\|_2$ be the Euclidean distance between the cell i and the mean. CAMP selects a subset of representative cells, termed coreset or archetypes, according to the following sampling distribution [14]:

$$q_i \;=\; \tfrac{1}{2}\cdot\frac{1}{n} \;+\; \tfrac{1}{2}\cdot\frac{d_i^2}{\sum_{r=1}^{n} d_r^2} \tag{1}$$

We now introduce the formal notion of a lightweight coreset, which is a small weighted subset approximating the full dataset for solving k-means with small relative error.

Definition 1 ([14, Definition 1. (ε, k)-lightweight coreset for k-means])**.** *Let $\varepsilon \in (0,1)$ and $k \in \mathbb{N}$. Let $\mathcal{X} \subset \mathbb{R}^d$ be a set of points with mean $\mu(\mathcal{X})$. A weighted set C is an (ε, k)-lightweight coreset of $\mathcal{X}$ if for every set $Q \subset \mathbb{R}^d$ of cardinality at most k, the k-means costs on $\mathcal{X}$ and C are within a relative error ε:*

$$(1-\varepsilon)\,\phi_{\mathcal{X}}(Q) \le \phi_C(Q) \le (1+\varepsilon)\,\phi_{\mathcal{X}}(Q) \tag{2}$$

The following theorem shows that sampling points according to (1) produces such a coreset as described in Definition 1 with high probability.

Theorem 1 ([14, Theorem 2.])**.** *Let $\varepsilon \in (0,1)$, $\delta \in (0,1)$ and $k \in \mathbb{N}$. Let $\mathcal{X}$ be a set of points in $\mathbb{R}^d$ and let coreset C contain cells sampled according to distribution (1) with a sample size m of at least*

$$m \;\ge\; c\,\frac{d\,k\log k + \log\!\left(\frac{1}{\delta}\right)}{\varepsilon^2} \tag{3}$$

where c is an absolute constant. Then, with probability all but δ, C is a (ε, k)-lightweight coreset (see Definition 1) of $\mathcal{X}$ for solving k-means problems.

Theoretical Guarantee Informally, Theorem 1 shows that if the coreset size m satisfies

$$m \;\ge\; O(dk) \tag{4}$$

then, with probability all but δ, the sampled set C forms a (ε, k)-lightweight coreset of $\mathcal{X}$ for the k-means objective as described in Definition 1. In other words, the clustering error computed on the coreset C approximates that of the full dataset $\mathcal{X}$ within a multiplicative factor of $1 \pm \varepsilon$ as stated in (2).

In our setting, the feature dimension d corresponds to the number of highly variable genes (HVGs) selected during preprocessing, typically set to the top 2,000 most variable genes. Since $d \ll n$ in large single-cell datasets, the required coreset size scales approximately as $m = O(k)$. Therefore, to infer k metacells, it suffices to sample on the order of k representative cells.

2.2 Kernel-based algorithm variants

We introduce CAMP, a family of coreset-accelerated metacell algorithms that share a common lightweight sampling strategy (see Section 2.1) but differ in how distances or similarities are utilized during metacell assignment. CAMP1 serves as our default variant and performs assignment directly in the gene-expression space using Euclidean distances. Motivated by observations in [13] that archetypal formulations tend to emphasize boundary cells while under-representing dense interior regions, we develop CAMP2 and CAMP3 as kernel-based extensions that can capture richer geometric structures. CAMP4 further combines gene-space sampling with kernel-space assignment to improve robustness across heterogeneous transcriptomic landscapes. While all variants follow the same coreset-based sampling pipeline, they exhibit different performance tradeoffs, which form the basis for a practical selection guideline that we introduce in Section 4.

Algorithm 1 CAMP1-4

```
Require: Input matrix $\mathcal{X}$, where:
  CAMP1: $\mathcal{X} \in \mathbb{R}^{n \times d}$ (cell-by-gene expression matrix)
  CAMP2/3: $\mathcal{X} \in \mathbb{R}^{n \times n}$ (cell-by-cell similarity matrix)
  CAMP4: $\mathcal{X} \in \mathbb{R}^{n \times d}$ (cell-by-gene expression matrix), $\mathcal{X}_k \in \mathbb{R}^{n \times n}$ (cell-by-cell similarity matrix)
Require: Coreset $C = \{c_1, \ldots, c_m\} \in \mathcal{X}$ sampled using distribution (1)
Ensure: Membership map $z : \mathcal{X} \to \{1, \ldots, m\}$

    CAMP1–3: Euclidean assignment
    Define $d(x, c) \leftarrow \|x - c\|_2$
    Initialize centers $\mu_j \leftarrow c_j$
    for $x \in \mathcal{X}$ do
        $z(x) \leftarrow \arg\min_j d(x, \mu_j)$
    end for

    CAMP4: Kernel-based assignment
    Initialize $B \in \mathbb{R}^{n \times m}$ with zeros
    for $j = 1$ to $m$ do
        $(B)_{s_j, j} \leftarrow 1$
    end for
    Compute $A \leftarrow \mathcal{X}_k B$
    for $i = 1$ to $n$ do
        $z(i) \leftarrow \arg\max_j A[i, j]$
    end for

    return $z$
```

According to Algorithm 1, CAMP1 samples a lightweight geometric coreset and treats the selected points as archetypes. Each remaining cell is then assigned to its nearest archetype in a single step based on Euclidean distance. Optional refinement with a few Lloyd iterations can be applied to adjust cluster boundaries (Supplementary Algorithm 1). Supplementary Figure 1a shows that CAMP1's coreset provides broad and well-distributed coverage of the transcriptomic manifold, indicating that Euclidean geometry alone is often sufficient to represent the global structure of scRNA-seq data. Supplementary Figure 1b and Figure 1c further confirm that CAMP2 and CAMP3 generate coresets with comparable coverage despite operating in kernel-defined similarity spaces.

To capture higher-order structure that may not be directly reflected in Euclidean distances, CAMP2 replaces raw geometry with similarities derived from the linear kernel

$$\mathcal{X}_{\text{linear}}(x_i, x_j) = x_i^\top x_j \tag{5}$$

which highlights direction-based relationships between expression vectors and can better distinguish cell states that differ along correlated gene-expression patterns. This representation is particularly effective in high-dimensional spaces, where angular information frequently provides a more stable notion of similarity than pointwise Euclidean distances.

CAMP3 generalizes this idea by incorporating local density information through an adaptive Gaussian kernel [13]. Here, similarities decay with Euclidean distance but are additionally scaled by bandwidth parameters σ_i and σ_j that reflect local neighborhood variability:

$$\mathcal{X}_{\text{adGau}}(x_i, x_j) = \frac{1}{\sqrt{2\pi(\sigma_i + \sigma_j)}} \exp\left(-\frac{(x_i - x_j)^\top (x_i - x_j)}{2(\sigma_i + \sigma_j)}\right) \tag{6}$$

The adaptive term $(\sigma_i + \sigma_j)$ allows the kernel to expand in sparse regions and contract in dense ones, yielding a similarity landscape that is sensitive to varying transcriptional densities and more robust to dropout or sparsity. This design mirrors the adaptive kernels used in SEACells [13].

Finally, CAMP4 integrates geometric and kernel-based representations in a hybrid strategy. Coreset sampling is first performed in the original gene-expression space to ensure broad geometric coverage, leveraging the efficiency and stability of the Euclidean coreset. Each cell is then assigned to the coreset representative with which it has the highest similarity under the adaptive Gaussian kernel defined in (6), allowing the assignment step to capture nonlinear or density-dependent similarity patterns that Euclidean distance may miss. This two-stage design keeps CAMP4 computationally lightweight while enabling more flexible modeling of cell-cell similarity.

3 Results

We used two datasets to benchmark the performance of CAMP against state-of-the-art metacell partitioning approaches, evaluating both metacell quality and downstream performance in a cell-type classification task.

3.1 Datasets and benchmarking setup

The first is a single-cell RNA-seq dataset [24] of human peripheral blood mononuclear cells (PBMC) of seven hospitalized COVID-19 patients and six healthy controls, comprising approximately $44,721$ cells . The second dataset is a human single-cell atlas of fetal gene expression [1], containing $504,028$ cells from 77 main cell types, after applying the same balanced sampling strategy as in [8]. UMAP embeddings of the two datasets are shown in Supplementary Figures 2 and 3.

Standard preprocessing steps included normalization by total counts per cell, scaling to 10,000 counts per cell, and applying a $\log(1 + p)$ transformation to stabilize variance. We then selected the top 2,000 highly variable genes to preserve the most biologically informative features while reducing technical noise. Principal component analysis (PCA) was then performed using the default implementation in *scikit-learn* to maintain a consistent latent space across all methods.

For all state-of-the-art methods, we used the official GitHub implementations (see Supplementary Section 4) with their default configurations unless otherwise noted. Because the original SEACells implementation was infeasible to run on the human fetal atlas dataset, we applied a set of low-level, algorithm-neutral optimizations to its Python/SciPy codebase. These modifications removed redundant Python-level loops and refactored several core routines into vectorized, sparse-efficient operations. In the original version, constructing the kernel matrix alone required more than 22 hours on the human fetal atlas dataset. After optimization, the same step completed in approximately 8 minutes, representing a speedup of over 160×. For MetaQ on the same large dataset, we additionally reduced the number of training epochs from default 300 to 40, decreased the convergence threshold from 10 to 2, and increased the batch size from 512 to 1024 to ensure convergence within available computational resources. All methods were executed through the integrated benchmarking pipeline of [7] to ensure consistency and reproducibility.

All computational experiments in this study were performed on a high-performance computing cluster. Each job was executed on a single node with an Intel Xeon Gold 6226R CPU (2.90 GHz) and 120 GB of RAM for both the PBMC and the human fetal atlas datasets. No GPU acceleration was used in any experiment, and each run was limited to a maximum of 48 hours.

3.2 Metacell quality metrics

We evaluated intrinsic structural and biological coherence of metacells. Specifically, we implemented established quality metrics that are widely applied in metacell benchmarking studies [25][13] to evaluate the homogeneity of cells within metacells and the heterogeneity across metacells. As noted in [25], comparing compactness and separation across methods can be biased when each method operates in a different latent space. To mitigate this potential unfairness, all methods were provided identical PCA-transformed data, ensuring a consistent geometric basis for comparison. For a given metacell $m \in [k]$ where k represents the total number of metacells, we summarize the metrics in Table 1 below.

Metric	**Definition**	**Desirable Direction**
Compactness	$\text{compactness}(m) = -\frac{1}{N}\sum_{i=1}^{N}\text{Var}(\vec{x}_i^m)$	↑ (higher = more homogeneous)
Separation	$\text{separation}(m) = \min_{\ell \neq m} \text{dist}(m, \ell)$	↑ (higher = better separation)
SC Ratio	$\text{SC Ratio}(m) = \frac{\text{separation}(m)}{\sqrt{-\text{compactness}(m)}}$	↑ (higher = better balance)
ECDF	$\text{ECDF}(t) = \frac{1}{k}\sum_{i=1}^{k}\mathbf{1}_{\{x_i \leq t\}}$	→ (flatter / right-shifted = better)
Purity	$\text{purity}(m) = \max_j \frac{\lvert M \cap C_j\rvert}{\lvert M\rvert}$	↑ (higher = more coherent)
INV	$\text{INV}(m) = \text{P}_{95}\left(\frac{\text{Var}(\vec{x}_g^m)}{\text{Mean}(\vec{x}_g^m)}\right)$	↓ (lower = less variability)

Table 1: Metacell Quality Metrics

Compactness measures how tightly cells belonging to a metacell cluster together in the latent space, where $\vec{x}_i^m$ represents the vector of the i-th diffusion component across all cells assigned to metacell m, and N is the embedding dimensionality. Note that we negate the compactness defined in [25] without loss of generality. Separation quantifies how distinct each metacell is from its nearest neighbor in latent space, where $\text{dist}(m, \ell)$ is the Euclidean distance between the centroids of metacells m and ℓ in diffusion space.

Due to the inherent tradeoff between compactness and separation discussed in [25], we propose a combined metric, the SC Ratio, to jointly evaluate the balance between intra-metacell homogeneity and inter-metacell distinctness. To visualize the distributional characteristics of SC Ratio values, we use the empirical cumulative distribution function (ECDF), defined for a set of k SC Ratio values $\{x_1, x_2, \ldots, x_k\}$ for k metacells, where $\mathbf{1}_{\{x_i \leq t\}}$ is the indicator function that equals 1 if $x_i \leq t$ and 0 otherwise.

Purity evaluates the consistency of celltype composition within each metacell. For a metacell m and ground-truth celltype labels $\{C_j\}$, denote M as the set of cells belonging to metacell m, and C_j is the set of cells annotated as label j. The INV metric captures the internal variability of gene expression within each metacell relative to its mean level, focusing on highly variable genes, where $\vec{x}_g^m$ denotes the vector of expression values for gene g across cells in metacell m, and P_{95} denotes the 95th percentile computed over all genes.

Furthermore, we evaluated how well metacell representations preserve biologically meaningful signal in a downstream analysis task, namely cell-type classification. We computed balanced accuracy using the CellTypist majority voting framework [26]. Balanced accuracy accounts for potential class imbalance by averaging the recall (true positive rate) across all T cell types:

$$\text{Balanced Accuracy} = \frac{1}{T}\sum_{t=1}^{T}\frac{TP_t}{TP_t + FN_t}, \tag{7}$$

where TP_t and FN_t denote the number of true positive and false negative predictions for cell type t, respectively. This metric provides an unbiased estimate of classification performance even when the cell-type distribution is highly imbalanced.

3.3 Coresets produce geometrically and biologically coherent metacells

From the top row of Figure 1, we observe that, for most methods, except MetaCell2, compactness increases monotonically with the number of metacells, indicating that finer partitions capture increasingly homogeneous transcriptional neighborhoods in the PBMC dataset. Conversely, separation generally decreases as expected, as metacell centroids move closer in higher-resolution partitions. The irregular pattern of MetaCell2 likely arises from the need to generate smaller γ values to match the same number of metacells shown on the x-axis, effectively altering its resolution and destabilizing graph partitions.

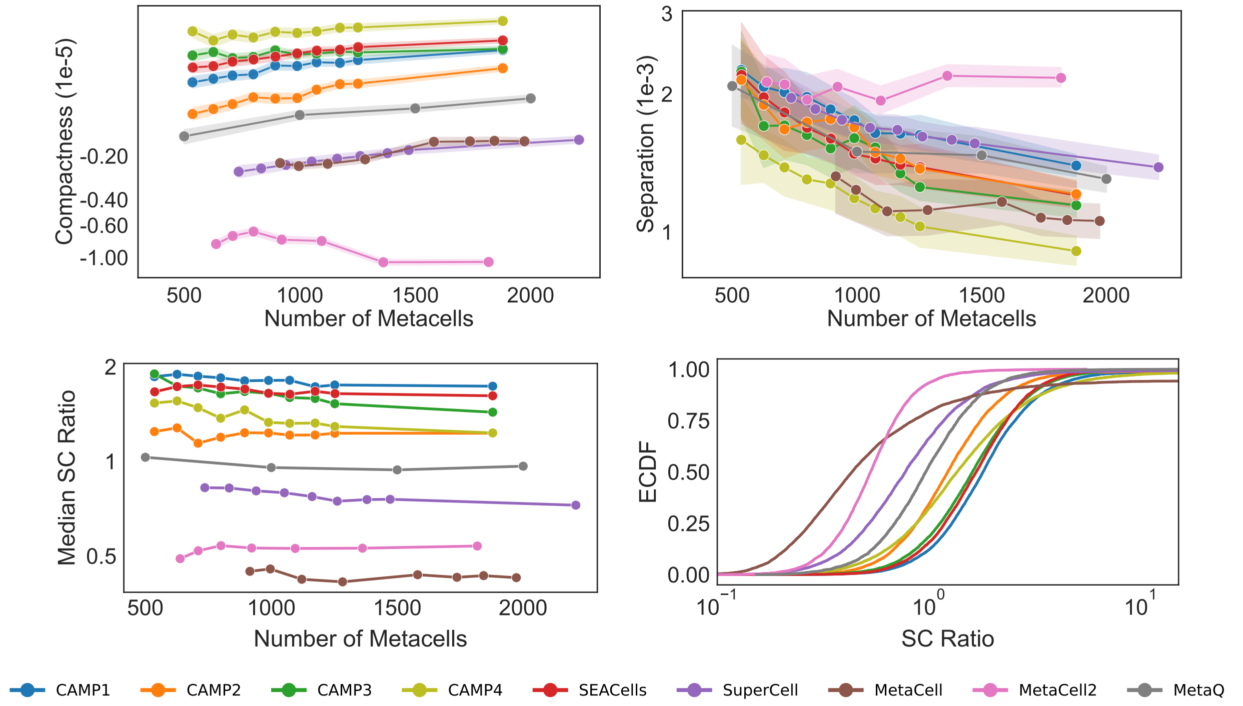


Figure 1: Compactness and separation of metacells produced by competing methods on the PBMC dataset. Because MetaCell and MetaCell2 do not support explicitly specifying the number of metacells, their results appear slightly shifted along the x-axis. One SuperCell point was omitted to maintain consistent resolution levels across methods.

In contrast, all CAMP variants, especially CAMP1, achieve strong compactness and separation profiles, ranking consistently among the top methods. CAMP2's slightly lower SC ratio reflects using $O(k)$ instead of its kernel-driven $O(nk)$ coreset requirement derived from (4), yet it remains highly competitive (see Section 4). The ECDF plot (Figure 1, bottom right) further underscores CAMP's stability: CAMP1 exhibits the most right-shifted distribution and CAMP4 the flattest, whereas the ECDF curve for MetaCell2 is both left-shifted and steep despite its higher separation.

For the human fetal atlas dataset, we observe the same overall trends (Figure 2). MetaQ, in contrast, suffers from unstable vector-quantization behavior on large datasets. Among all methods, CAMP1 and CAMP4 consistently yield the highest compactness and SC ratio values. SEACells achieves slightly higher separation values than CAMP1 and CAMP2, but this comes at the cost of substantially lower compactness. From the ECDF plot, all CAMP variants exhibit right-shifted and flatter distributions, while competing methods such as MetaQ and SuperCell show steeper or left-skewed curves. For the human fetal atlas dataset, the optimal γ settings [25] for MetaQ may fall outside the range evaluated here, potentially below $1,000$ metacells or above the highest resolution of $25,000$ examined in this study.

The purity trend closely parallels that of compactness, which is expected since cells with similar transcriptional profiles are more likely to belong to the same cell type. Consistent with their performances in terms of compactness, MetaCell2 in the PBMC and MetaQ in the human fetal atlas data from Figure 3 exhibit poor performance due to their smaller γ and unstable quantization, respectively. In contrast, our CAMP framework demonstrates consistent and biologically coherent performance. Among all methods, CAMP4 achieves the highest purity and lowest INV, indicating the most homogeneous and well-defined metacell compositions for both PBMC and the human fetal atlas dataset. The default variant, CAMP1, is competitive in terms of Purity across both datasets, outperforming MetaQ and MetaCell2 on the PBMC data, and MetaQ and SEACells on the human fetal atlas data.

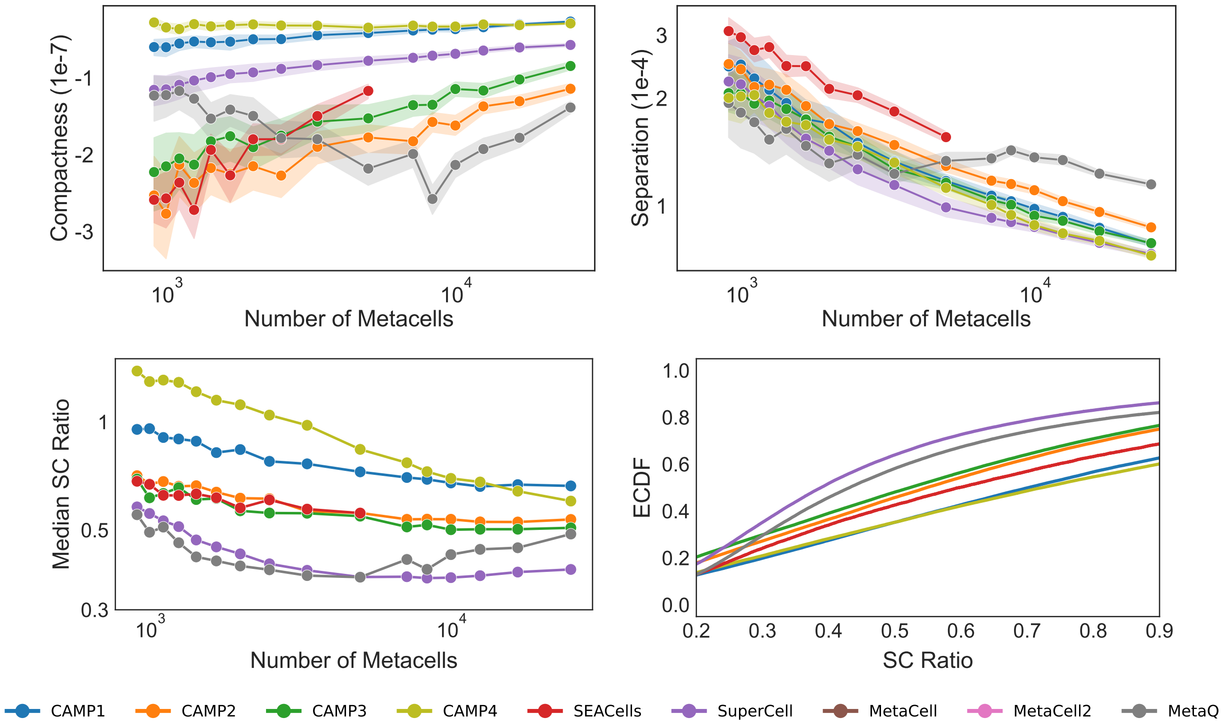


Figure 2: Compactness and separation of metacells returned by competing methods on the human fetal atlas dataset.

3.4 Fast and memory-efficient metacell construction

For the PBMC dataset (Figure 4, left), all CAMP variants consistently occupy the lower end of the runtime spectrum. At a resolution of 500 metacells, CAMP1 completes in approximately 1 second, whereas MetaQ requires 12,615 seconds, about four orders of magnitude slower. At finer resolutions near 2,000 metacells, CAMP1 finishes in 3 seconds while MetaQ takes 14,483 seconds, a difference of roughly 3.5 orders of magnitude. SEACells, despite achieving competitive compactness, requires 5,911 seconds at a comparable metacell resolution of 1,879, which is nearly three orders of magnitude slower than CAMP1. These discrepancies are consistent across the entire resolution range and highlight the substantial cost of iterative kernel-based updates in SEACells and the autoencoder training steps of MetaQ.

For the human fetal atlas dataset (Figure 4, right), all CAMP variants completed partitioning of the 504,028 cells in under 8 minutes across all metacell scales, with CAMP1-3 computing distances and similarities on-the-fly. In contrast, MetaCell and MetaCell2 each reached the 48-hour runtime ceiling, while SEACells required 10,189 seconds, which is more than an order of magnitude slower than CAMP, and ultimately failed with out-of-memory errors at higher resolutions approaching 7,000 metacells. MetaQ also becomes substantially slower at large scales, requiring 52,971 seconds at 25,000 metacells, approximately 113× slower than CAMP (nearly two orders of magnitude). SuperCell performs moderately faster than other non-CAMP methods due to its approximation parameter (see Supplementary Section 4), yet it still remains slower than all CAMP variants.

3.5 Accurate downstream classification from coreset-based metacells

In this section, we examine whether coreset-based partitions preserve sufficient biological signal to support accurate downstream classification. We assessed cell type prediction performance using CellTypist [26] with majority voting, reporting the balanced accuracy across the ten largest cell-type classes in each dataset. As no method reliably recovered the remaining low-frequency cell types, these minority populations were omitted from the quantitative analysis.

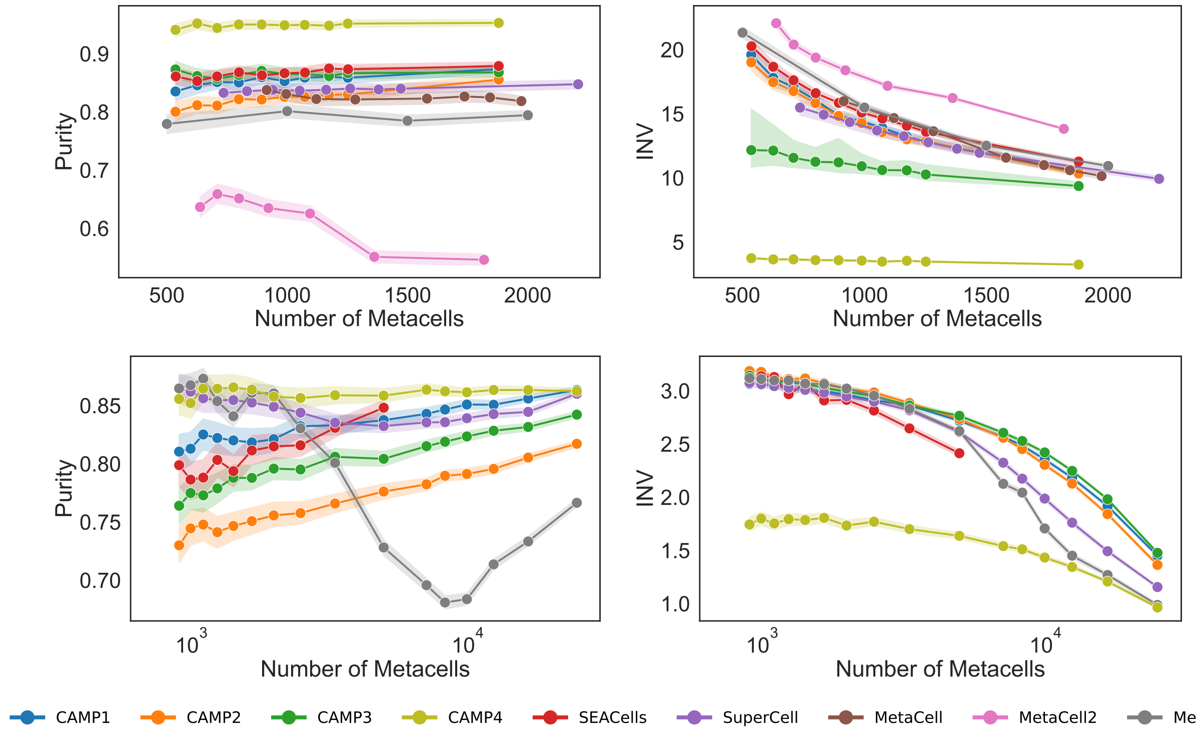


Figure 3: Purity and INV scores of metacells returned by competing methods on the PBMC (top) and the human fetal atlas datasets (bottom).

For the PBMC dataset, we assessed post hoc performance at a resolution of 1,000 metacells (Figure 5), as this is the common resolution at which all methods produce results. For the human fetal atlas data, results corresponding to 3,000 metacells are shown in Figure 6. Additional results with other metacell resolutions are provided in Supplementary Figures 4–9. The y-axis denotes the true labels and the x-axis the predicted labels, both sorted in descending order by class size (top to bottom and left to right). The corresponding cell types are listed in Supplementary Section 5.

Across both datasets, CAMP variants achieved the strongest overall performance. On the PBMC dataset (Figure 5), CAMP1 and CAMP4 obtained the highest balanced accuracies of 92.29% and 93.62%, respectively. In contrast, SEACells and MetaQ frequently misclassify Class-switched B cells as IgG plasmablasts, while MetaCell and MetaCell2 tend to confuse them with CD16 monocytes. SuperCell and MetaCell2 also show notable confusion between NK and CD8m T cells. On the human fetal atlas dataset (Figure 6), CAMP1, CAMP4, and MetaQ similarly achieved the top balanced accuracies (74.90%, 75.37%, and 74.95%), maintaining clean diagonal patterns in their confusion matrices. Most competing methods, except CAMP4 and MetaQ, show substantial confusion among neural cell types, especially assigning Inhibitory or Limbic neurons to the Excitatory neuron class. SEACells also performs poorly on Ganglion cells, often grouping them together with Excitatory neurons.

4 Conclusion

In this work, we introduced CAMP, a scalable and geometry-preserving framework for metacell construction that unifies coreset theory with archetypal analysis to address the computational and biological challenges of large-scale single-cell transcriptomics. By leveraging lightweight k-means coresets, CAMP identifies a small but highly representative subset of cells that captures the geometry of the transcriptomic manifold with high probability. This design provides strong theoretical guarantees while reducing both runtime and memory usage by orders of magnitude compared to existing metacell frameworks, including MetaCell, MetaCell2, SuperCell, SEACells, and MetaQ.

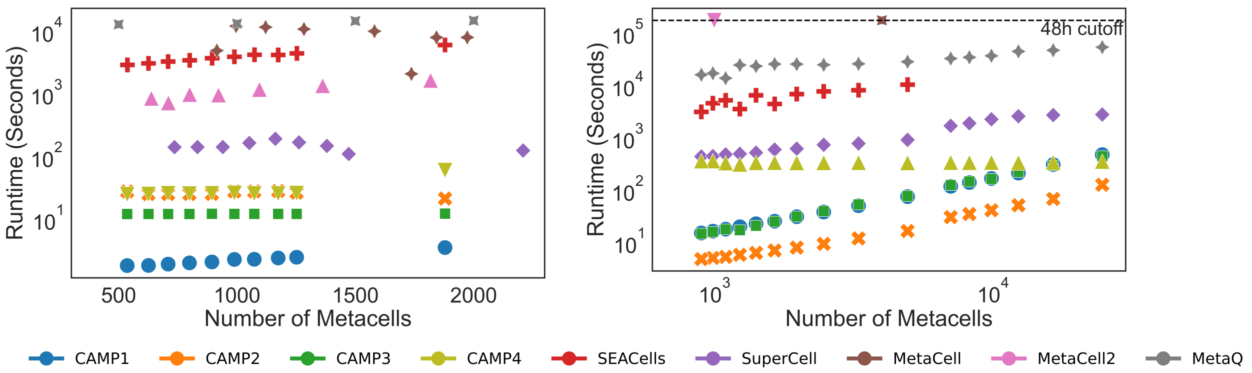


Figure 4: Runtime in seconds on the PBMC (left) and the human fetal atlas (right) datasets.

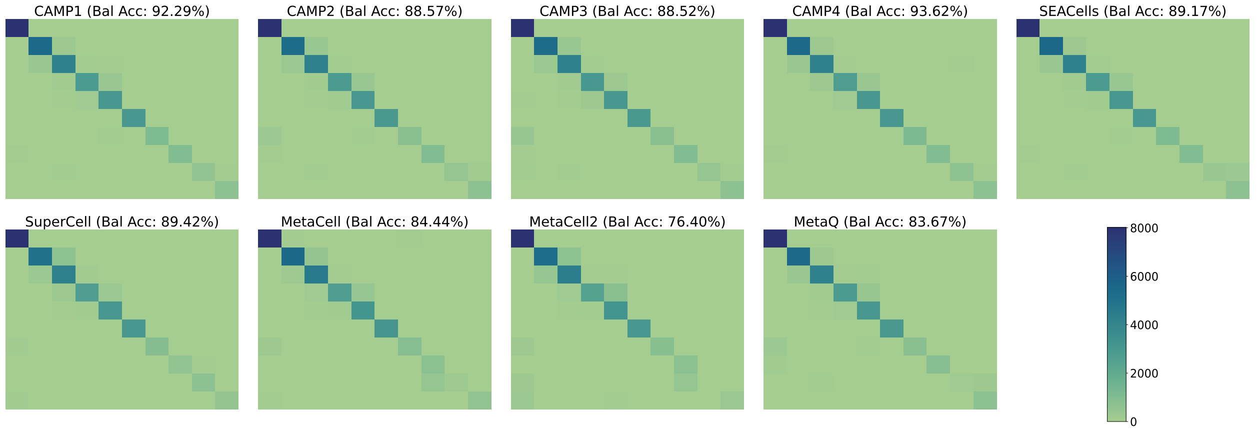


Figure 5: Cell-type confusion heatmaps for the ten largest cell-type classes based on 1,000 metacells computed by competing methods on the PBMC dataset. Balanced accuracy (Bal Acc) is reported on the top.

In our experiments, CAMP consistently produced compact, well-separated, and biologically coherent metacells. These geometric advantages translate into practical downstream benefits: CAMP1 and CAMP4 achieved the highest balanced accuracies in cell-type classification, demonstrating that CAMP preserves biologically meaningful variation even under aggressive compression. Importantly, CAMP attains this accuracy without incurring the heavy computational demands characteristic of prior methods. All CAMP variants completed metacell construction from half a million cells in under 8 minutes on CPU-only hardware, whereas several state-of-the-art approaches either exceed the 48-hour runtime limit or fail due to memory constraints. This establishes CAMP as a uniquely efficient and robust paradigm for atlas-scale metacell inference.

Collectively, its four variants allow CAMP to adapt to a wide range of biological settings. As summarized in Table 2 below, CAMP1 serves as a fast and reliable default; CAMP2 provides the most scalable option through fully vectorized linear-kernel updates; CAMP3 captures nonlinear and density-dependent structure when additional flexibility is required; and CAMP4 offers a balanced hybrid, trading a small amount of additional computation for greater robustness across heterogeneous transcriptomic landscapes. On the theoretical side, CAMP clarifies the relationship between coreset size and metacell complexity. The required sampling rate follows the bounds derived in Section 2.1: $m = O(k)$ for CAMP1, $m = O(nk)$ for CAMP2, and $m = O(k^2)$ for CAMP3. Although CAMP2 and CAMP3 have larger theoretical requirements due to their kernel constructions, our empirical findings demonstrate that sampling only $m = O(k)$ points is sufficient across diverse biological settings. This observation highlights an intriguing gap between worst-case analysis and practical behavior, and motivates future work on tightening theoretical guarantees for kernel-based CAMP variants.

[1] Although CAMP1–3 support on-the-fly (streaming) updates, only CAMP2 enables computationally efficient streaming. Its

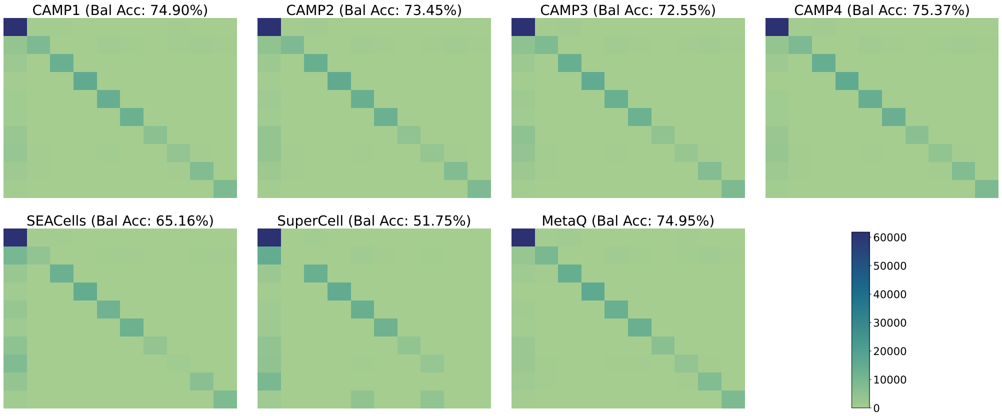


Figure 6: Cell-type confusion heatmaps for the ten largest cell-type classes based on 3,000 metacells computed by competing methods on the human fetal atlas dataset. Balanced accuracy (Bal Acc) is reported on the top.

Variant	Kernel	Compactness	Separation	Purity	INV	Accuracy	Streaming
CAMP1	Default Free	✓	✓	✓		✓	
CAMP2	Linear		✓				✓[1]
CAMP3	Non-linear	✓		✓			
CAMP4	Hybrid	✓		✓	✓	✓	

Table 2: CAMP variant selection guide across benchmark metrics. Streaming refers to efficient updates computed on-the-fly, i.e., only when needed, without constructing the full kernel or distance matrix in advance.

CAMP also raises several conceptual questions for methodological theory. Understanding how geometric summarization interacts with downstream tasks, including differential expression, clustering stability, and rare-cell detection. This could inspire metacell constructions tailored to specific biological objectives. Looking ahead, CAMP provides a modular foundation for several promising extensions. Geometry-preserving coresets could accelerate multi-omic metacell construction for ATAC-seq, CITE-seq, methylation, or spatial datasets, and incorporating trajectory-aware or pseudotime-informed constraints may allow CAMP to better capture continuous developmental processes.

All datasets, together with the complete CAMP implementation pipeline and reproducible scripts for all competing methods (including our SEACells optimizations), are publicly available in our GitHub repository: `https://github.com/danrongLi/CAMP`.

linear kernel allows fully vectorized operations in Python, enabling fast, loop-free batched dot-product updates.

References

1. Cao, J. *et al.* A human cell atlas of fetal gene expression. en. *Science* **370,** eaba7721. ISSN: 0036-8075, 1095-9203. (2025) (Nov. 2020).
2. Klein, A. M. *et al.* Droplet Barcoding for Single-Cell Transcriptomics Applied to Embryonic Stem Cells. en. *Cell* **161,** 1187–1201. ISSN: 00928674. (2025) (May 2015).
3. Macosko, E. Z. *et al.* Highly Parallel Genome-wide Expression Profiling of Individual Cells Using Nanoliter Droplets. en. *Cell* **161,** 1202–1214. ISSN: 00928674. (2025) (May 2015).
4. Zheng, G. X. Y. *et al.* Massively parallel digital transcriptional profiling of single cells. en. *Nature Communications* **8,** 14049. ISSN: 2041-1723. (2025) (Jan. 2017).
5. Bilous, M. *et al.* Metacells untangle large and complex single-cell transcriptome networks. en. *BMC Bioinformatics* **23,** 336. ISSN: 1471-2105. (2025) (Aug. 2022).
6. Lähnemann, D. *et al.* Eleven grand challenges in single-cell data science. en. *Genome Biology* **21,** 31. ISSN: 1474-760X. (2025) (Feb. 2020).
7. Liu, P. & Li, J. J. *mcRigor: a statistical method to enhance the rigor of metacell partitioning in single-cell data analysis* en. Oct. 2024. (2025).
8. Li, Y. *et al.* MetaQ: fast, scalable and accurate metacell inference via single-cell quantization. en. *Nature Communications* **16,** 1205. ISSN: 2041-1723. (2025) (Jan. 2025).
9. Huang, M. *et al.* SAVER: gene expression recovery for single-cell RNA sequencing. en. *Nature Methods* **15,** 539–542. ISSN: 1548-7091, 1548-7105. (2025) (July 2018).
10. Baran, Y. *et al.* MetaCell: analysis of single-cell RNA-seq data using K-nn graph partitions. en. *Genome Biology* **20,** 206. ISSN: 1474-760X. (2025) (Dec. 2019).
11. Andrews, T. S. & Hemberg, M. False signals induced by single-cell imputation. en. *F1000Research* **7,** 1740. ISSN: 2046-1402. (2025) (Mar. 2019).
12. Ben-Kiki, O., Bercovich, A., Lifshitz, A. & Tanay, A. Metacell-2: a divide-and-conquer metacell algorithm for scalable scRNA-seq analysis. en. *Genome Biology* **23,** 100. ISSN: 1474-760X. (2025) (Dec. 2022).
13. Persad, S. *et al.* SEACells infers transcriptional and epigenomic cellular states from single-cell genomics data. en. *Nature Biotechnology* **41,** 1746–1757. ISSN: 1087-0156, 1546-1696. (2025) (Dec. 2023).
14. Bachem, O., Lucic, M. & Krause, A. *Scalable k -Means Clustering via Lightweight Coresets* en. in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (ACM, London United Kingdom, July 2018), 1119–1127. ISBN: 978-1-4503-5552-0. (2025).
15. Hart, Y. *et al.* Inferring biological tasks using Pareto analysis of high-dimensional data. en. *Nature Methods* **12,** 233–235. ISSN: 1548-7091, 1548-7105. (2025) (Mar. 2015).
16. Hammer, B., Martinetz, T. & Villmann, T. *Workshop New Challenges in Neural Computation* Oct. 2015.
17. Mair, S. & Brefeld, U. *Coresets for Archetypal Analysis* in *Advances in Neural Information Processing Systems* (eds Wallach, H. *et al.*) **32** (Curran Associates, Inc., 2019).
18. Har-Peled, S. & Mazumdar, S. *On coresets for k-means and k-median clustering* en. in *Proceedings of the thirty-sixth annual ACM symposium on Theory of computing* (ACM, Chicago IL USA, June 2004), 291–300. ISBN: 978-1-58113-852-8. (2025).
19. Bandyapadhyay, S., Fomin, F. V. & Simonov, K. *On Coresets for Fair Clustering in Metric and Euclidean Spaces and Their Applications* Version Number: 1. 2020. (2025).
20. Braverman, V. *et al. The Power of Uniform Sampling for Coresets* Version Number: 2. 2022. (2025).
21. Cohen-Addad, V., Larsen, K. G., Saulpic, D. & Schwiegelshohn, C. *Towards Optimal Lower Bounds for k-median and k-means Coresets* Version Number: 1. 2022. (2025).
22. Feldman, D., Schmidt, M. & Sohler, C. *Turning Big data into tiny data: Constant-size coresets for k-means, PCA and projective clustering* Version Number: 1. 2018. (2025).

23. Lucic, M., Bachem, O. & Krause, A. *Strong Coresets for Hard and Soft Bregman Clustering with Applications to Exponential Family Mixtures* in *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics* (2016).

24. Wilk, A. J. *et al.* A single-cell atlas of the peripheral immune response in patients with severe COVID-19. en. *Nature Medicine* **26,** 1070–1076. ISSN: 1078-8956, 1546-170X. (2025) (July 2020).

25. Bilous, M., Hérault, L., Gabriel, A. A., Teleman, M. & Gfeller, D. Building and analyzing metacells in single-cell genomics data. en. *Molecular Systems Biology* **20,** 744–766. ISSN: 1744-4292. (2025) (May 2024).

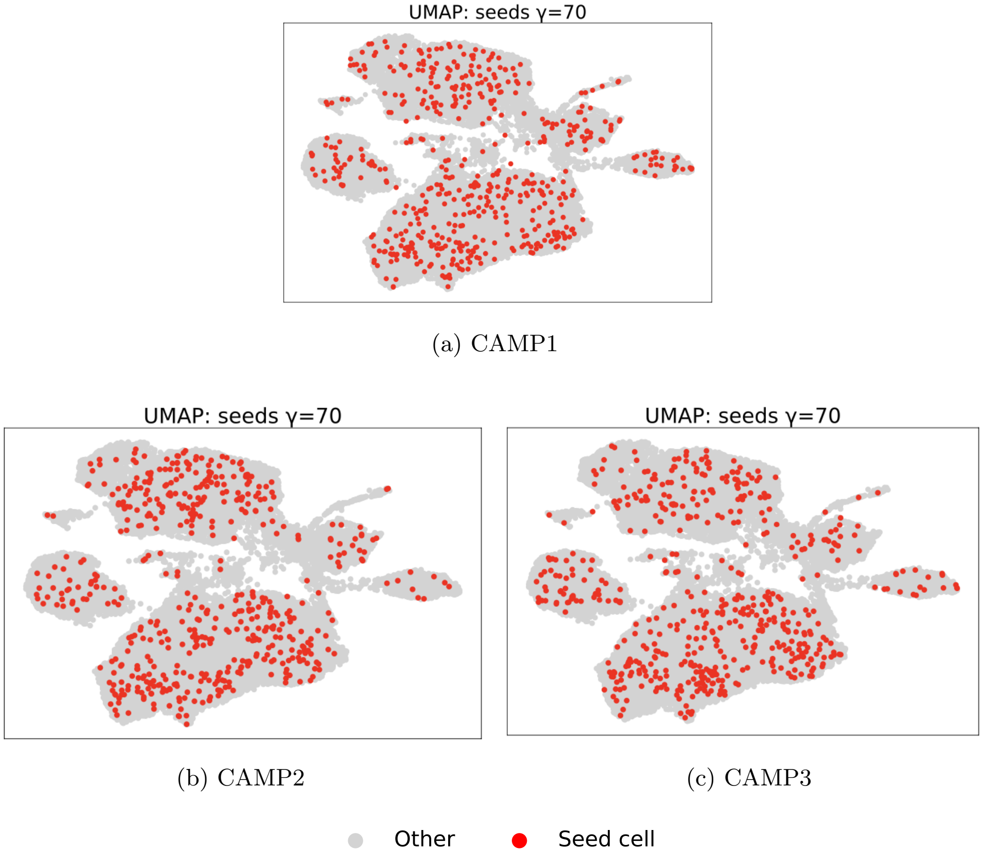
26. Domínguez Conde, C. *et al.* Cross-tissue immune cell analysis reveals tissue-specific features in humans. en. *Science* **376,** eabl5197. ISSN: 0036-8075, 1095-9203. (2024) (May 2022).

Supplementary Information
CAMP: Coreset Accelerated Metacell Partitioning enables scalable analysis of single-cell data

Danrong Li[1], Young Kun Ko*[1], and Stefan Canzar*[2]

[1]Department of Computer Science and Engineering, The Pennsylvania State University, University Park, PA 16802, United States
[2]Faculty of Informatics and Data Science, University of Regensburg, Germany

1 Seed distribution for CAMP1-3



(a) CAMP1

(b) CAMP2

(c) CAMP3

Figure 1: Seed distribution on the PBMC dataset.

Figure 1 shows that even when omitting a computationally expensive archetypal analysis, the lightweight coresest accurately covers the the inner region.

*Corresponding authors

2 Refinement Algorithm

Algorithm 1 CAMP Refinement (Lloyd-style updates for CAMP1-3)

```
Require: Dataset $\mathcal{X}$, initial assignments $z(x)$ and centers $\{\mu_j\}_{j=1}^m$ from initial CAMP step in Algorithm 1
Require: Maximum iterations $T \leftarrow 10$
Ensure: Refined membership map $z : \mathcal{X} \to \{1, \dots, m\}$
for $t = 1$ to $T$ do
    for $j = 1$ to $m$ do
        $S_j \leftarrow \{x \in \mathcal{X} : z(x) = j\}$
        if $|S_j| > 0$ then
            $\mu_j \leftarrow \frac{1}{|S_j|} \sum_{x \in S_j} x$        ▷ update center to cluster mean
        else
            $x^\star \leftarrow \arg\max_{x \in \mathcal{X}} \min_{\ell \in \{1,\dots,m\}} d(x, \mu_\ell)^2$        ▷ farthest unassigned point
            $\mu_j \leftarrow x^\star$        ▷ re-seed empty cluster
        end if
    end for
    for $x \in \mathcal{X}$ do
        $z(x) \leftarrow \arg\min_{j \in \{1,\dots,m\}} d(x, \mu_j)$        ▷ reassign to nearest updated center
    end for
end for
return $z$
```

3 UMAP embeddings of PBMC and human fetal atlas data

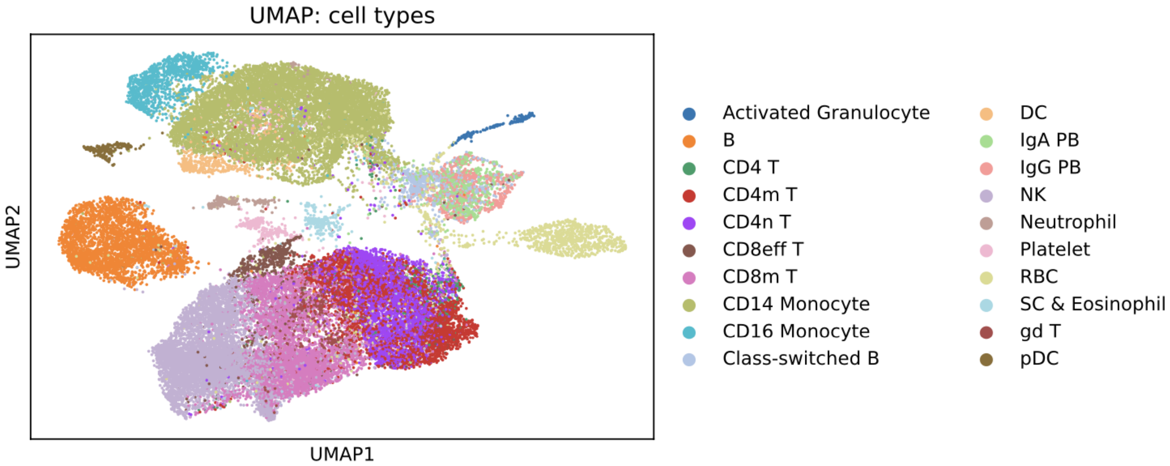


Figure 2: UMAP embedding of the PBMC dataset.

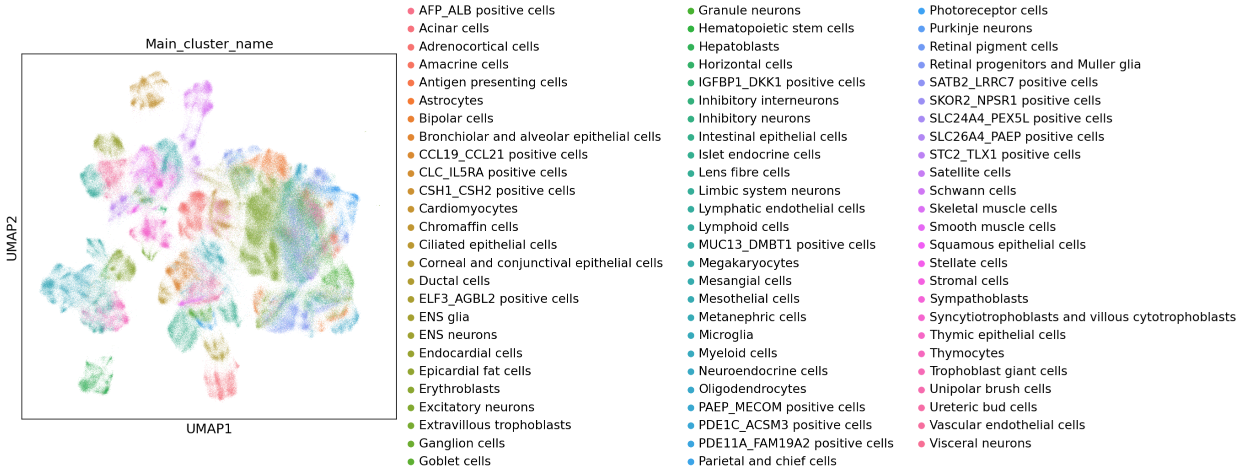


Figure 3: UMAP embedding of the human fetal atlas dataset.

4 Software packages used for state-of-the-art methods

For SEACells, we optimized it by modifying `cpu.py` and `build_graph.py` files in the original Python implementation (`https://github.com/dpeerlab/SEACells`) and used this optimized version with `use_sparse=True`, which is available only in CPU mode. SEACells produced out-of-memory errors on the human fetal atlas dataset whenever the target average number of cells per metacell, γ, was ≤ 70. For SuperCell, we used the R package (`https://github.com/GfellerLab/SuperCell`) with the option `do_approx=True`. For MetaQ, we used the Python implementation (`https://github.com/XLearning-SCU/MetaQ`) with default parameters on the PBMC dataset. On the human fetal atlas dataset, we changed the default parameters as discussed in main text. MetaCell and MetaCell2 were run using the R package (`https://tanaylab.github.io/metacell/`) and the Python package (`https://pypi.org/project/metacells/`), respectively. On the human fetal atlas dataset, MetaCell (configured with `knn = 50` and `amp = 1`) exceeded the 48-hour runtime limit, and MetaCell2 exceeded the limit for $\gamma = 900$ and $\gamma = 500$.

5 Ten largest cell-type classes

PBMC	**human fetal atlas**
CD14 Monocyte	Excitatory neurons
NK	Inhibitory neurons
CD8m T	Astrocytes
CD4m T	Adrenocortical cells
CD4n T	Granule neurons
B	Purkinje neurons
RBC	Metanephric cells
CD16 Monocyte	Ganglion cells
Class-switched B	Limbic system neurons
IgG PB	Intestinal epithelial cells

Table 1: Top 10 cell types in terms of class size in descending order from top to bottom for both the PBMC and the human fetal atlas datasets.

6 Cell type prediction on the human fetal atlas dataset

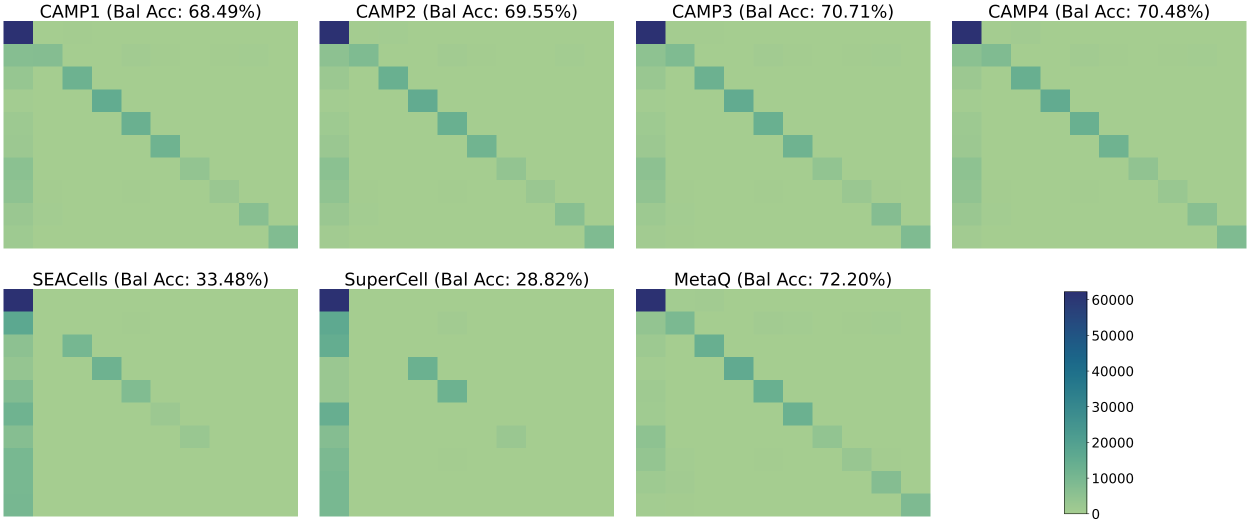


Figure 4: Cell-type confusion heatmaps for the ten largest cell-type classes based on 1,000 metacells computed by competing methods on the human fetal atlas dataset. Balanced accuracy (Bal Acc) is reported on the top.

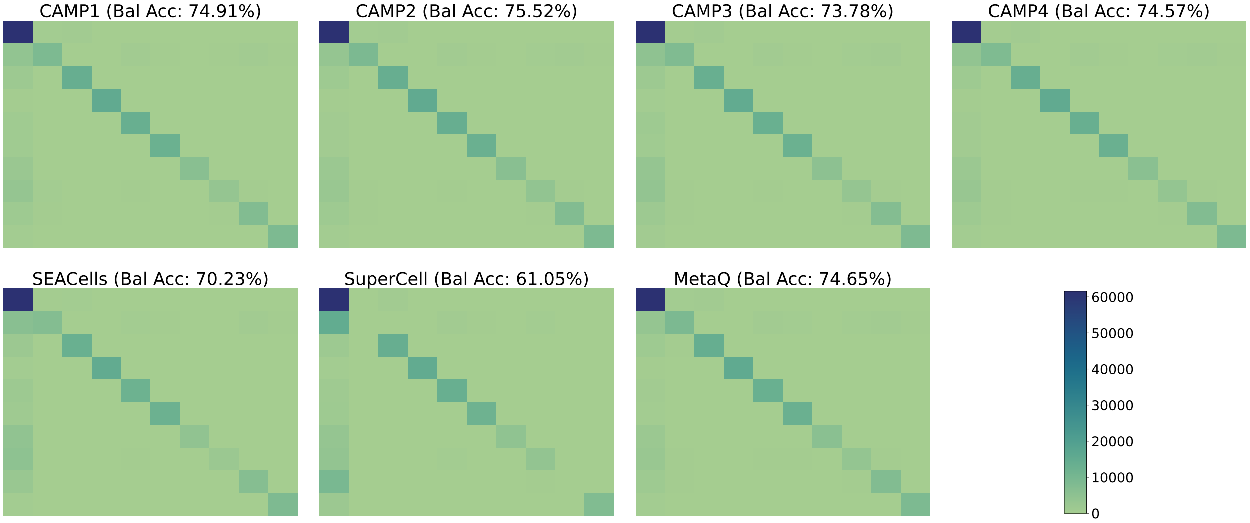


Figure 5: Cell-type confusion heatmaps for the ten largest cell-type classes based on 5,000 metacells computed by competing methods on the human fetal atlas dataset. Balanced accuracy (Bal Acc) is reported on the top.

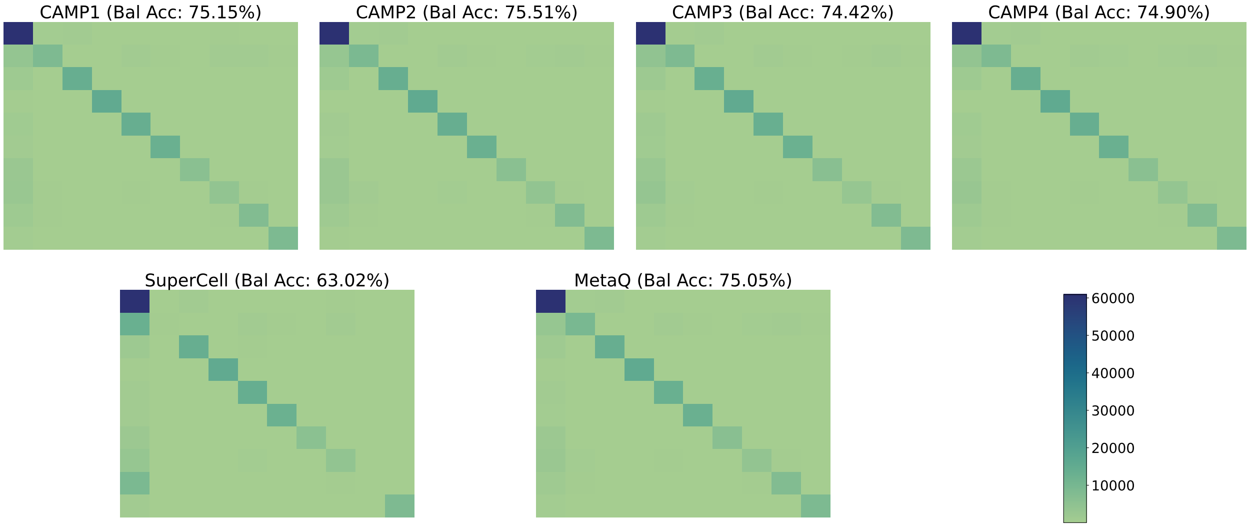


Figure 6: Cell-type confusion heatmaps for the ten largest cell-type classes based on 7,000 metacells computed by competing methods on the human fetal atlas dataset. Balanced accuracy (Bal Acc) is reported on the top.

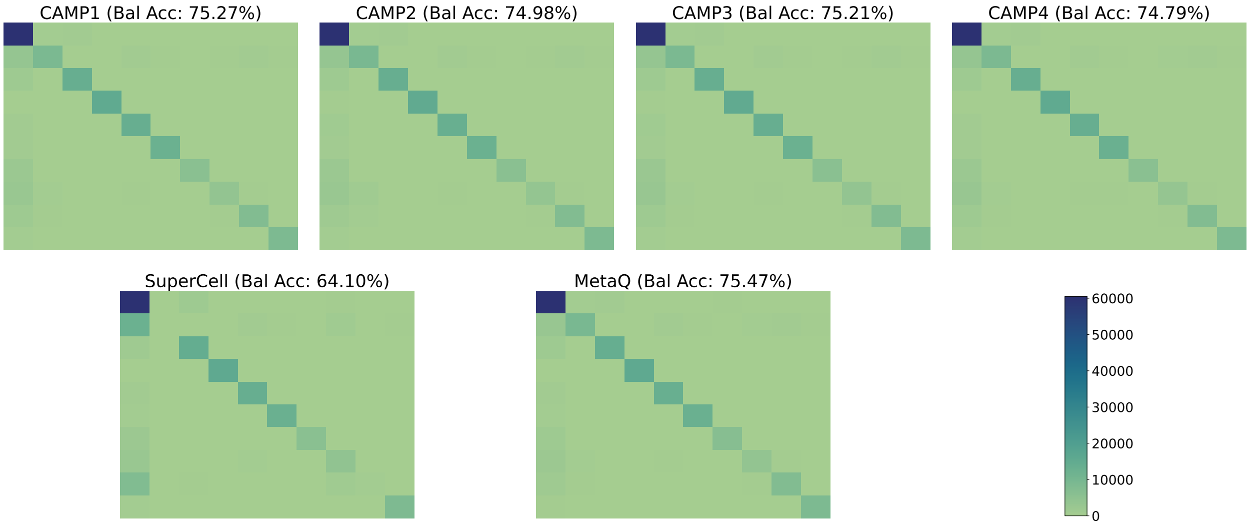


Figure 7: Cell-type confusion heatmaps for the ten largest cell-type classes based on 10,000 metacells computed by competing methods on the human fetal atlas dataset. Balanced accuracy (Bal Acc) is reported on the top.

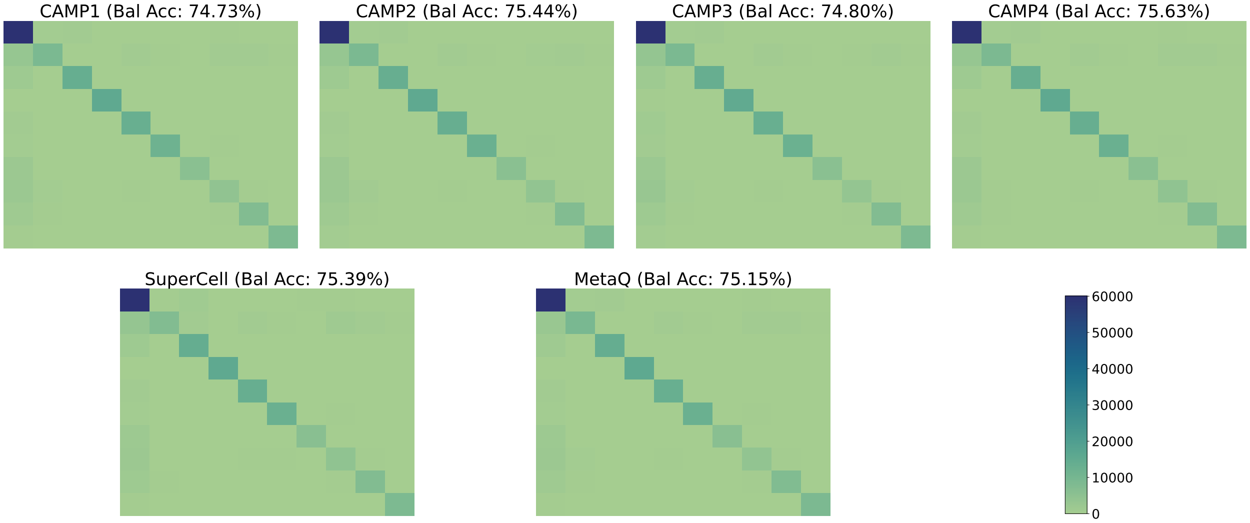


Figure 8: Cell-type confusion heatmaps for the ten largest cell-type classes based on 15,000 metacells computed by competing methods on the human fetal atlas dataset. Balanced accuracy (Bal Acc) is reported on the top.

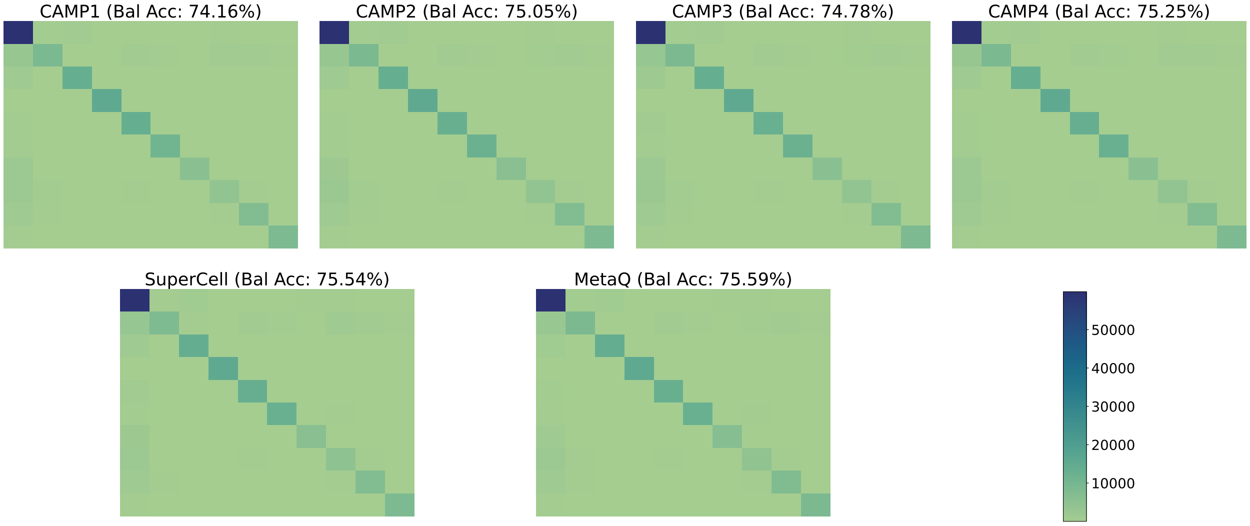


Figure 9: Cell-type confusion heatmaps for the ten largest cell-type classes based on 25,000 metacells computed by competing methods on the human fetal atlas dataset. Balanced accuracy (Bal Acc) is reported on the top.