nature communications

Article

https://doi.org/10.1038/s41467-025-56424-6

MetaQ: fast, scalable and accurate metacell inference via single-cell quantization

Received: 15 July 2024

Accepted: 14 January 2025

Published online: 31 January 2025

Check for updates

Yunfan Li[1], Hancong Li[2,3], Yijie Lin[1], Dan Zhang[4], Dezhong Peng[1], Xiting Liu[5], Jie Xie[6], Peng Hu[1], Lu Chen[4], Han Luo[2,3] & Xi Peng[1,7] ✉

To overcome the computational barriers of analyzing large-scale single-cell sequencing data, we introduce MetaQ, a metacell algorithm that scales to arbitrarily large datasets with linear runtime and constant memory usage. Inspired by cellular development, MetaQ conceptualizes each metacell as a collective ancestor of biologically similar cells. By quantizing cells into a discrete codebook, where each entry represents a metacell capable of reconstructing the original cells it quantizes, MetaQ identifies homogeneous cell subsets for efficient and accurate metacell inference. This approach reduces computational complexity from exponential to linear while maintaining or surpassing the performance of existing metacell algorithms. Extensive experiments demonstrate that MetaQ excels in downstream tasks such as cell type annotation, developmental trajectory inference, batch integration, and differential expression analysis. Thanks to its superior efficiency and effectiveness, MetaQ makes analyzing datasets with millions of cells practical, offering a powerful solution for single-cell studies in the era of high-throughput profiling.

The rapid advancements in single-cell capture and sequencing technologies give rise to a continuously increasing number of profiled cells[1,2], exhibiting advantages in revealing cell heterogeneity[3] and reconstructing developmental trajectories[4]. On the flip side, this surge in large-scale sequencing data poses a significant computational hurdle for downstream analyses. For instance, a typical single-cell data analysis pipeline[5]—encompassing data integration, clustering, visualization, and differential expression analysis—requires about 16 h to process half a million cells on a standard desktop[6]. When the cell number slightly increases to 600,000, the above pipeline can crash due to memory exceeding, even on a professional computing platform with 512 GB RAM[6]. To handle large-scale data, several scalable and efficient single-cell analysis tools have been developed for downstream tasks such as imputation[7,8], integration[9–11], clustering[12–14], and cell type annotation[15,16]. Nonetheless, these methods are commonly tailored for specific tasks and cannot be easily integrated into well-established frameworks[5,17], leading to additional learning and deployment challenges.

Instead of exhaustively scaling up various analysis tools, a more direct and general solution is to compress the sequencing data, thereby energizing all commonly used methods to handle arbitrarily large datasets once and for all. As a specific implementation, metacell algorithms[18] propose merging homogeneous cell subsets into metacells to reduce redundancy, given that biologically similar cells are often repeatedly sampled during high-throughput profiling. The inferred metacells act as proxies of the original cells, which could be analyzed using existing tools without any modification while enjoying the following two merits. On the one hand, metacells decrease the

[1]School of Computer Science, Sichuan University, Chengdu, Sichuan, China. [2]Department of Thyroid and Parathyroid Surgery, Laboratory of Thyroid and Parathyroid Disease, Frontiers Science Center for Disease Related Molecular Network, West China Hospital, Sichuan University, Chengdu, Sichuan, China. [3]Sichuan Clinical Research Center for Laboratory Medicine, Chengdu, Sichuan, China. [4]Department of Laboratory Medicine, State Key Laboratory of Biotherapy, West China Second University Hospital, Sichuan University, Chengdu, Sichuan, China. [5]School of Computer Science, Georgia Insitute of Technology, Atlanta, GA, USA. [6]College of Life Science, Sichuan Normal University, Chengdu, Sichuan, China. [7]State Key Laboratory of Hydraulics and Mountain River Engineering, Sichuan University, Chengdu, Sichuan, China. ✉e-mail: pengxi@scu.edu.cn

computational expense by reducing the cell number. On the other hand, metacells alleviate data sparsity by aggregating the features of similar cells. However, despite the promising application prospect, it remains challenging to accurately and efficiently infer metacells. For example, the state-of-the-art method SEACell[19] requires more than one day to compute metacells for 100,000 cells, struggling to handle larger datasets due to significant memory overhead, which makes it less practical. Recently, MetaCell V2[20] improved algorithmic scalability by leveraging a divide-and-conquer strategy, albeit at the cost of achieving local optima. SuperCell[6] employs the efficient Walktrap community detection algorithm[21] to expedite metacell inference. However, like SEACell and MetaCell V2, SuperCell still requires exponentially increasing running time with respect to cell number, limiting its scalability to larger datasets.

The suboptimal scalability and performance of existing methods could be attributed to their identical focus on mining local neighborhoods where cells are similar to each other. Consequently, all existing methods resort to constructing and partitioning pair-wise similarity graphs, which are computationally expensive and limited by the reliability of nearest neighbors. In this work, we introduce a perspective on metacells by drawing an analogy to the hierarchical nature of cell differentiation in multicellular organisms. Specifically, cells develop from a single, low-differentiation state, progressing through various stages. For instance, in the hematopoietic system, pluripotent hematopoietic stem cells differentiate through several stages into mature B cells, including intermediate forms like pro-B cells, pre-B cells, and immature B cells[22]. Upon maturation, B cells further diversify into subtypes with distinct functions in the immune system[23]. Such differentiation is driven by characteristic gene expressions that define the primary cell type and function, while specific feature expressions further refine these cells into subtypes with distinct roles. Similarly, metacells can be viewed as representing a common state of specialization among closely related cells. Analogous to an ancestor in the developmental pathway, each metacell functions as a collective entity that aggregates multiple specialized cells, capturing their shared features. In other words, a subset of biologically similar cells can be effectively derived from a single metacell.

By conceptualizing metacells in this manner, we present MetaQ, a fast, scalable, and accurate metacell algorithm based on single-cell quantization. Unlike existing methods that laboriously mine neighborhood structures, MetaQ quantizes all cells into a codebook with a limited number of entries, where each entry corresponds to a metacell. By viewing each metacell as a collective ancestor of a subgroup of specialized cells, MetaQ encourages each codebook entry to reconstruct all the cells it quantizes. To achieve better reconstruction, the model would naturally quantize biologically similar cells into the same entry, inherently achieving cell grouping for metacell inference. This simple yet effective design of MetaQ allows it to process various types of count data in a fully unsupervised manner. More importantly, MetaQ exhibits linear time complexity with respect to the number of cells, while retaining a constant memory consumption. This makes MetaQ a metacell algorithm that scales to arbitrarily large datasets, setting it apart from existing methods[6,19,20] that suffer from exponential time or memory complexity. Furthermore, while existing metacell algorithms are designed for uni-omics data, MetaQ supports metacell inference from paired multi-omics data by extending the reconstruction target, making it versatile for comprehensive single-cell analysis. Notably, different from previous single-cell clustering and classification methods[12–16], MetaQ pursues homogeneity within fine-grained cell subsets in a generative manner instead of mining discriminative heterogeneity between different cell types.

Extensive experimental results demonstrate the superiority of MetaQ in various downstream tasks, including cell type annotation, developmental trajectory inference, batch integration, clustering, and differential expression analysis. Moreover, MetaQ scales to arbitrarily large datasets, requiring linearly increasing running time and constant memory costs relative to the cell number. In summary, the proposed MetaQ simultaneously enjoys efficiency, scalability, and accuracy for metacell inference, which makes it a promising single-cell analysis tool in the high-throughput single-cell profiling era with a continuously growing number of cells and omics.

Results

MetaQ infers metacells via single-cell quantization

MetaQ is a deep learning-based metacell algorithm that infers metacells through cell quantization in a generative manner. As depicted in Fig. 1, MetaQ builds upon an auto-encoder framework enhanced with a cell quantization mechanism. Specifically, given a raw count matrix as input, MetaQ first learns cell embeddings with the encoder network. In the embedding space, MetaQ quantizes cells into a discrete codebook with learnable entries, where the number of entries corresponds to the user-defined metacell number. During quantization, each cell would be assigned to its nearest codebook entry, while each entry is responsible for reconstructing all the cells it quantizes via the decoder network. Such a design is predicated on our perspective of viewing each metacell as a collective ancestor of a subgroup of specialized cells, allowing similar cells to be effectively derived from a single entry. The embedding, quantization, and reconstruction processes are simultaneously performed in an end-to-end fashion. To improve the reconstruction performance, the model tends to quantize biologically similar cells into the same codebook entry, encapsulating compressed information about those cells. In other words, the cell quantization process essentially identifies homogeneous cell subsets. In addition to the joint optimization with encoder and decoder networks, the codebook entries are further adjusted based on their historical usage, to stabilize the optimization and prevent cells from collapsing into a few entries. Notably, MetaQ naturally supports metacell inference for paired multi-omics data. In brief, MetaQ incorporates multi-omics features in cell embeddings and requires the quantized cell embeddings to reconstruct original count matrices across all modalities. When the training converges, MetaQ infers metacells by averaging the original count values of cells quantized into each codebook entry. The resulting metacell count matrix provides a condensed representation of the original cell population, preserving dense features while significantly reducing the number of cells. These inferred metacells can then be directly used for downstream analyses, acting as an efficient and representative substitute for the original single-cell data.

MetaQ effectively and efficiently infers prototypical metacells for cell type annotation

To evaluate the scalability and performance of MetaQ, we first applied it to the human fetal atlas dataset consisting of 433,395 cells across 54 types. Figures 2a and 2c show UMAP visualizations of the original cells and the metacells inferred by four methods, each metacell labeled by the majority type of original cells it represents. The results indicate that MetaQ effectively separates different cell types while preserving the structure of similar cells. For instance, one could observe a clear grouping of retina cells, including retinal progenitors and Muller glia, photoreceptor cells, and retinal pigment cells, mirroring the original cell groupings (highlighted with red boxes). In contrast, the metacells inferred by the other three methods resulted in a confounded grouping of retina cells with other cell types. We visualized the density maps of the original cells and metacells inferred by MetaQ in Supplementary Figs. 3a and 3c. Overall, the metacell density aligns with that of the original cells, namely, they are denser in areas with a high density of single cells and vice versa. Such a density consistency enables the metacells to reflect the underlying cell type distributions more accurately. Additionally, the metacell assignments depicted in Supplementary Fig. 3b show that the features of original cells, including those rare cell types, are

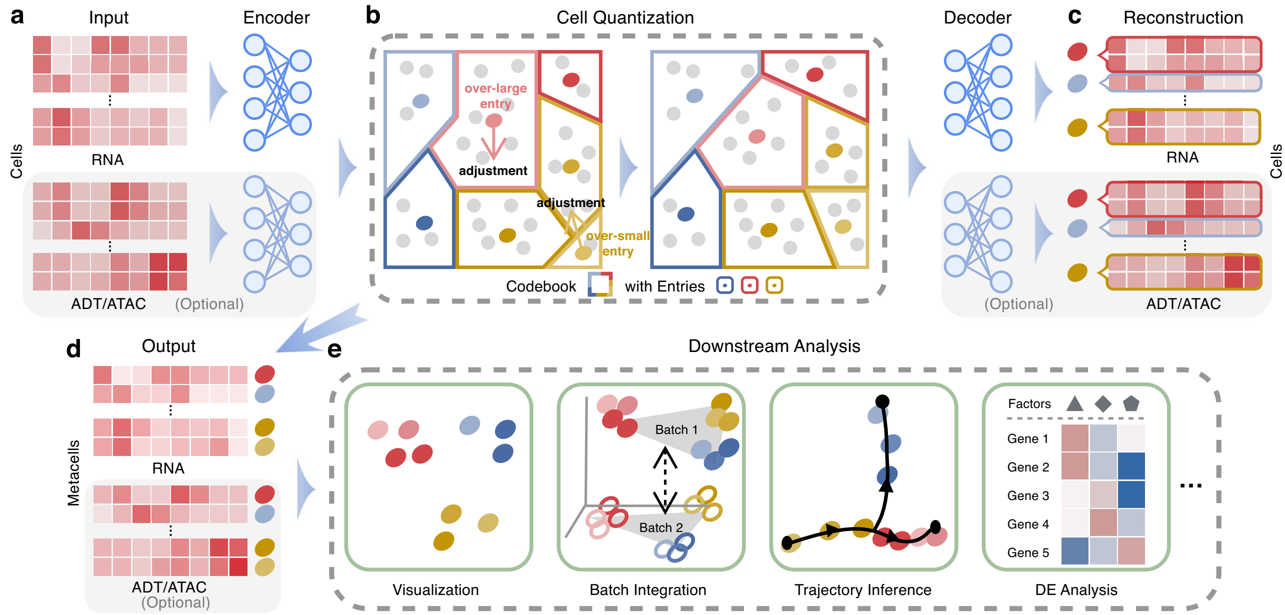
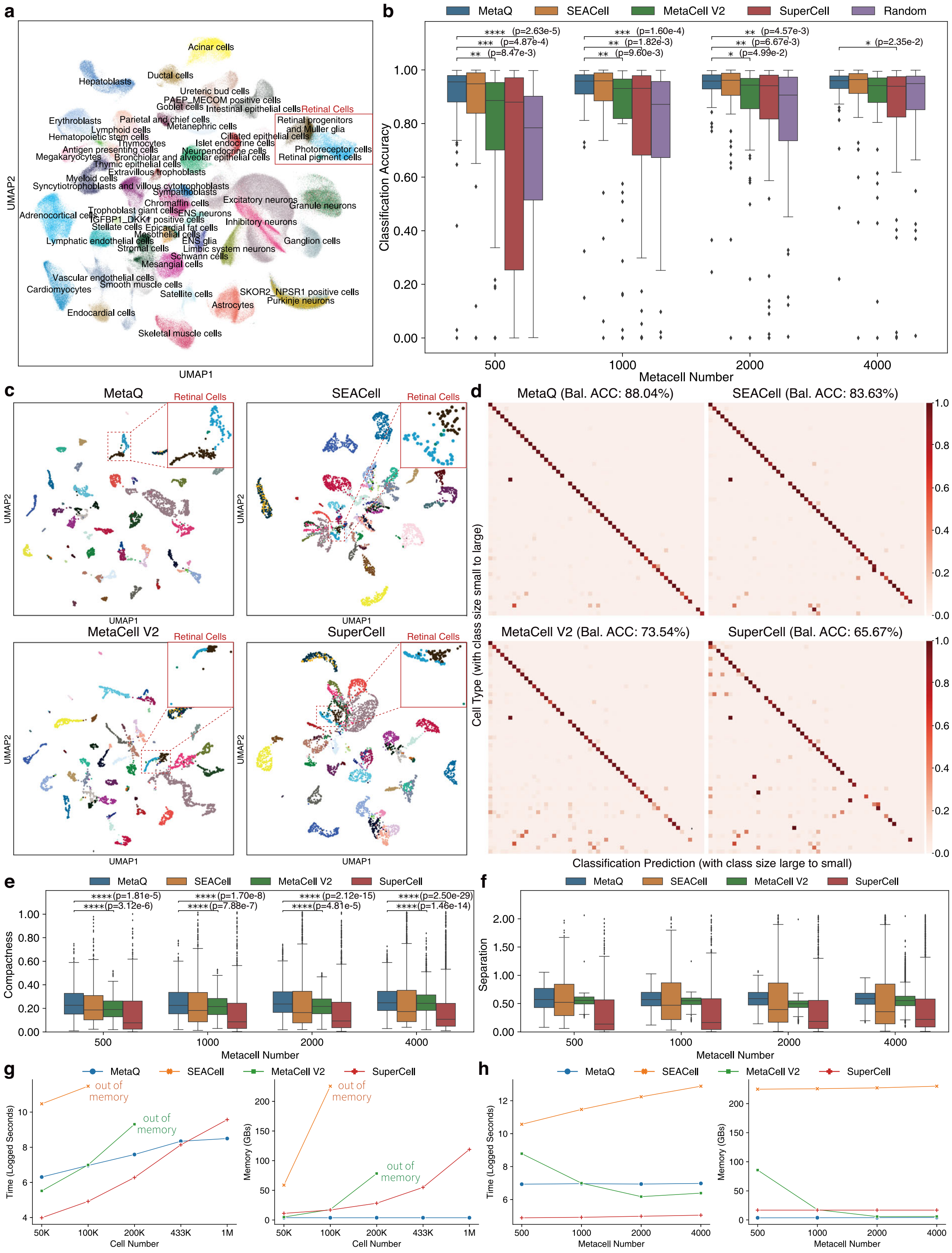


Fig. 1 | Overview of the MetaQ algorithm. a MetaQ accepts uni-omics or paired multi-omics raw count matrices as inputs and learns cell embeddings through the encoder network. **b** In the embedding space, MetaQ introduces a discrete codebook with learnable entries, where each entry corresponds to the embedding of a metacell. Cells are then quantized by assigning each to its nearest entry in the codebook. To prevent degenerated quantization, MetaQ records the entry usage and adjusts the codebook to eliminate over-large or over-small entries. **c** MetaQ encourages each codebook entry to reconstruct the input omics data of all cells it quantizes with the decoder network. For better reconstruction, the model tends to assign biologically similar cells into the same codebook entry, which intrinsically achieves cell grouping for metacell inference. **d** After training, MetaQ computes metacells by averaging the original uni- or multi-omics count data of cells within each codebook entry. **e** The inferred metacells serve as a compressed representation of the original data, which could be seamlessly used for single-cell downstream analysis (DE: Differential Expression).

effectively covered by MetaQ metacells. Beyond intuitive visual comparisons, we quantitatively assessed the compactness and separation of metacell inferred by different algorithms. As shown in Figs. 2e and 2f, MetaQ consistently achieves the highest median scores of compactness and separation across various metacell numbers. Supplementary Fig. 3e also reveals that MetaQ exhibits larger differences between within- and between-metacell cell similarities than baseline methods. These results collectively underscore the superior performance of MetaQ in aggregating homogeneous cells and distinguishing between heterogeneous ones.

In addition to direct comparisons, we further evaluated the metacell quality through a downstream cell type classification task. Specifically, we trained a classifier using metacells to categorize the original cells, where each metacell is labeled according to the most prevalent cell type among its constituent cells. To accurately classify the original cells, the metacells used for training are expected to exhibit both high purity and high prototypicality. High purity ensures that each metacell predominantly contains cells of the same type, leading to reliable annotations. High prototypicality guarantees that the metacells capture representative features, enhancing the generalization ability of the classifier to original cells. In other words, the downstream classification performance reflects the overall quality of metacells. We evaluated MetaQ and three baseline methods with varying metacell numbers, presenting the results in Fig. 2b. To demonstrate the effectiveness of metacells, we also included a naive baseline by randomly sampling the same number of cells from the original data. From 500 to 4000 metacells, the classification model trained by MetaQ metacells consistently outperforms other baselines in terms of average accuracy, showcasing the superior performance of MetaQ. To further elucidate the performance gap, we illustrated the confusion matrix of the predicted labels in Fig. 2d, with full cell type names provided in Supplementary Fig. 2 due to the space limitation. As shown, the model trained with MetaQ metacells better discriminates between cell types, especially rare ones such as thymic epithelial cells, leading to the highest balanced classification accuracy (88.04% compared with the second-best 83.63% by SEACell). Besides classifying original cells by training a classifier on metacells, we alternatively assigned each original cell the majority cell type of its corresponding metacell. The balanced accuracy of this majority-voted prediction, as shown in Supplementary Fig. 3f, also underscores MetaQ's superior performance (92.91% compared with the second-best 83.78% by SEACell). Furthermore, we visualized the cell type purity of metacells in Supplementary Fig. 3g, which shows that MetaQ is the only method capable of identifying PAEP_MECOM positive cells (with a proportion of 0.0597%) originating from placental tissue[2] and epithelial cells from the thymus (with a proportion of 0.0662%), the two rarest cell types. These results collectively demonstrate that MetaQ metacells preserve the information of both common and rare cell types more effectively than baseline methods.

The main purpose of metacell algorithms is to alleviate the substantial computational burden in single-cell analyses as previously discussed. Thus, in addition to the metacell quality, we were also concerned about the efficiency of metacell algorithms. To this end, we measured the (logged) running time and memory costs of all methods on datasets ranging from 50 thousand to 1 million cells. As shown in Fig. 2e, MetaQ exhibits linearly increasing running time and constant memory usage relative to the number of cells, theoretically scaling to arbitrary data sizes (see Supplementary Note 2 for more details). Although SuperCell is efficient on relatively small subsets of less than 200,000 cells, it requires exponentially increasing time and linearly increasing memory, leading to limited scalability for larger datasets. Moreover, as shown in Fig. 2b, SuperCell achieves inferior classification performance compared to other methods, even worse than the naive random sampling baseline with 4,000 metacells. Due to the exponential memory costs, SEACell and MetaCell V2 exceed 512 GB memory—a common configuration for computational servers—when processing 200,000 and 433,000 cells, respectively. Notably, compared to the most competitive baseline SEACell in metacell



quality, the proposed MetaQ achieves approximately a 100 times speedup when processing 100,000 cells (0.3 hours versus 26.7 hours). We further investigated the influence of the metacell number on computational expenses. As shown in Fig. 2f, MetaQ and SuperCell are insensitive to the number of metacells, MetaCell V2 favors larger metacell numbers to activate its divide-and-conquer strategy, and SEACell requires linearly increasing time relative to the metacell number. Notably, one could enable SEACell on the full dataset by inferring metacells in a hierarchical fashion, namely, first inferring metacells within each sample and then performing a secondary metacell aggregation across samples. Supplementary Fig. 4c indicates that hierarchical SEACell achieves performance on par with MetaQ. However, this improvement comes at a significant computational cost. Supplementary Fig. 4d reveals that hierarchical SEACell

Fig. 2 | MetaQ effectively and efficiently infers prototypical metacells. **a** UMAP visualization of the original 433,495 cells from the human fetal atlas. The red box highlights retina cells. **b** Classification accuracy of cell classifiers trained with [500, 1000, 2000, 4000] metacells inferred by MetaQ, SEACell, MetaCell V2, SuperCell, and random sub-sampling on five random experiments. Each boxplot ranges from the upper and lower quartiles with the median as the horizontal line and whiskers extend to 1.5 times the interquartile range. Two-sided T-test results: *$0.01 < p \le 0.05$, **$0.001 < p \le 0.01$, ***$0.0001 < p \le 0.001$, ****$p \le 0.0001$. **c** UMAP visualization of 4000 metacells inferred by the four methods, with cell type colors matching those in **b**. Retina cells are marked with red boxes. **d** Agreement between the ground-truth annotations and the labels predicted by classification models trained with 500 metacells. Matrices with a clearer diagonal structure indicate better classification performance. **e** Compactness of [500, 1000, 2000, 4000] metacells inferred by different methods on five random experiments. Each boxplot ranges from the upper and lower quartiles with the median as the horizontal line and whiskers extend to 1.5 times the interquartile range. Two-sided T-test results: ****$p \le 0.0001$. **f** Separation of [500, 1000, 2000, 4000] metacells inferred by different methods on five random experiments. Each boxplot ranges from the upper and lower quartiles with the median as the horizontal line and whiskers extend to 1.5 times the interquartile range. **g** Running times (logged) and memory cost for inferring 1000 metacells from different numbers of original cells. **h** Running times (logged) and memory cost for inferring different numbers of metacells from 100,000 cells. Source data are provided as a Source Data file.

on full data took over a week to complete, whereas running MetaQ only requires about an hour. Such a dramatic improvement in computational efficiency makes MetaQ more favorable in practical use. In summary, the proposed MetaQ not only infers accurate and prototypical metacells, but also offers the best computational scalability for large datasets, making it an effective and efficient tool for metacell analysis.

MetaQ supports multi-omics analysis and preserves cell developmental trajectory

Advances in single-cell technologies have enabled the simultaneous profiling of cells across multiple layers[24–26], taking advantage of the pairing information in multi-omics analyses. However, existing metacell methods are all designed for uni-omics data. In this case, computing metacells independently for each modality would result in the loss of pairing information between metacells of different modalities. In contrast, the proposed MetaQ can directly infer paired metacells from multi-omics data, by reconstructing inputs across all modalities using the quantized cell embeddings. Further details are provided in the *handling multi-omics data* subsection and Supplementary Fig. 1.

To evaluate the multi-omics metacell inference performance of MetaQ, we applied it to the human bone marrow CITE-seq dataset[27] which includes 30,672 cells with RNA and antibody-derived tag (ADT) data. The original two modalities are visualized in Fig. 3a. As shown, the ADT modality excels in identifying subsets of T and natural killer (NK) cells, while the RNA modality more effectively distinguishes other marrow cells, including progenitors, myeloid cells, and B cells. For comparisons, since existing methods are not tailored for multi-omics data, we reorganized the inputs based on the API interface of different methods to produce paired metacells. Specifically, we constructed the kernel on the concatenated PCA-reduced RNA and ADT data for SEACell. For MetaCell V2 and SuperCell, we normalized the count data in each modality and concatenated them as the input. Given the paired multi-omics metacells, we then utilized WNN to compute the joint embedding. As depicted in Fig. 3b, MetaQ and SuperCell better preserve the original structure than the other two methods, especially for hematopoietic precursors.

For further validation, we applied PAGA[28] trajectory inference on MetaQ metacells. Figs. 3e and 3f depict the developmental trajectory from hematopoietic stem cells (HSCs) to plasmablasts. The analysis of gene expression dynamics along this trajectory reveals a decrease in markers associated with immature B cells, such as *VPREB1* and *MME*, coupled with an increase in markers associated with mature follicular B cells, including *MS4A1* and *CD19*. These mature follicular B cells reside in the lymphoid follicles of the spleen and lymph nodes, comprising both mature-naive (*CD27−*) and memory (*CD27+*) B cells[29]. Upon antigen activation, B cells rapidly proliferate, undergo immunoglobulin class-switch recombination (*IGHA2*, *IGHG4*), and differentiate into short-lived plasmablasts. These plasmablasts, characterized by elevated levels of *MZB1* and *SDC1*, produce antibodies and function as effector cells in the early antibody response. Additionally, we recapitulated the dendritic cell (DC) maturation process, identifying two distinct differentiation trajectories: one leading to plasmacytoid dendritic cells (pDCs) and the other to classical dendritic cells (cDCs), as shown in Supplementary Fig. 5a. Along the pDC developmental path, there is a notable upregulation of pDC lineage genes, such as *IL3RA*[30] and *IRF7*[31], in a subset of Prog_DC, suggesting differentiation towards the pDC phenotype. Similarly, in the cDC2 trajectory, cDC maturation markers *CLEC10A* and *CD1C*[32] exhibit progressive upregulation. Moreover, Supplementary Fig. 5b demonstrates that MetaQ effectively captures the erythroid lineage evolution, from HSCs and progressing to progenitor red blood cells (Prog_RBC). Throughout this progression, *CD34* expression gradually decreases while the expression of hemoglobin complex genes, including human alpha-like (*HBA2*, *HBA1*) and delta-like (*HBD*) globin genes[33], increases. The above results demonstrate that the metacells inferred by MetaQ successfully preserve the developmental trajectories.

To demonstrate the superiority of MetaQ in multi-omics metacell inference, we quantitatively compared metacell purity across different cell types. The purity metric is defined as the frequency of the most represented cell type within the metacell, with higher values indicating better metacell membership. Based on the cell type discriminability of the two modalities, we broadly categorized all cell types into two superclasses in Fig. 3d, with the top and bottom panels corresponding to RNA- and ADT-informative cells, respectively. According to the overall metacell purity for the two superclasses illustrated in Supplementary Fig. 5c, MetaQ achieves comparable metacell purity to the best competitor SEACell for RNA-informative cell types (93.5% to 92.5% on average with T-test p-value of 0.325, degrees of freedom = 662, 95% confidence interval = [−0.0095, 0.0288]). On ADT-informative cell types, however, MetaQ significantly outperforms SEACell (93.0% to 90.0% on average with T-test p-value of 0.005, degrees of freedom = 560, 95% confidence interval = [0.0092, 0.0518]), particularly on CD8 memory and effector T cells. These results indicate that MetaQ better integrates information from both modalities during the metacell inference. Moreover, we evaluated the compactness and separation of metacells in both RNA and ADT modalities. As depicted in Fig. 3c, MetaQ and SEACell outperform MetaCell V2 and SuperCell in average performance metrics. Additionally, MetaQ and MetaCell V2 demonstrate superior stability in metacell quality, as evidenced by the more concentrated distributions in the boxplot.

In addition to evaluating MetaQ on CITE-seq RNA+ADT data, we further tested its efficacy using the 10x multiome mouse kidney dataset[34], encompassing 14,527 cells with paired gene expression and chromatin accessibility profiles. Alongside the three previous baselines, we also incorporated EpiCarousel[35], a recent metacell algorithm specifically designed for scATAC-seq data for comparisons. We first compared the performance of MetaQ against baseline methods on the scATAC-seq uni-omics peak data. As depicted in Supplementary Fig. 6a, MetaQ and SEACell exhibit superior information retention on rare cell types compared to MetaCell V2, SuperCell, and EpiCarousel. Such a result is further corroborated by the cell type classification results in Supplementary Fig. 6b, where cells are assigned to the predominant type within each metacell. A higher balanced classification accuracy indicates better metacell purity. Notably, MetaQ with Poisson distribution modeling achieves performance on par

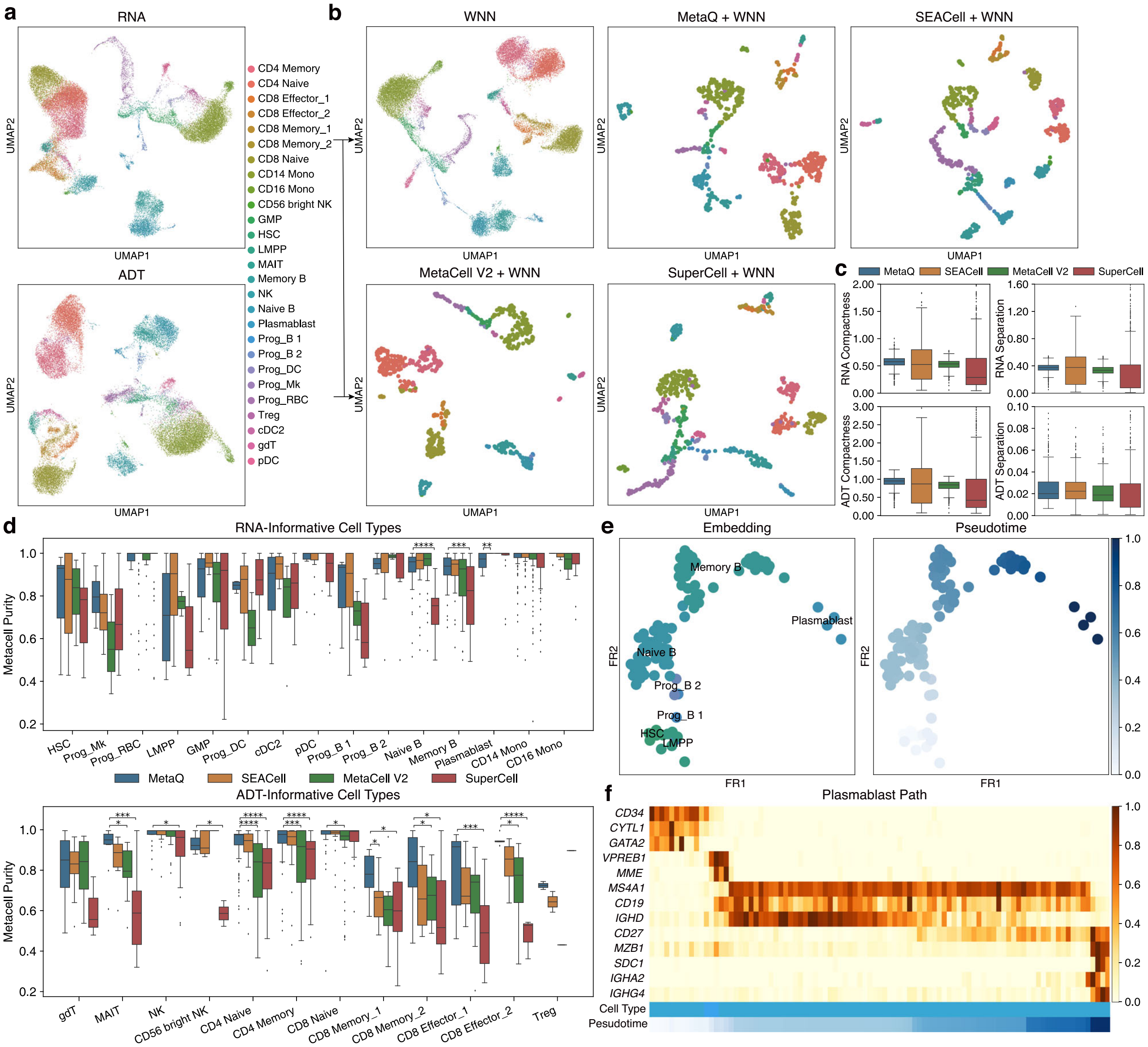


Fig. 3 | MetaQ supports multi-omics metacell inference and preserves developmental trajectory. a UMAP visualization of the original 30,672 cells from human bone marrow in RNA and ADT modalities (CD4 Memory Memory CD4+ T cell, CD4 Naive Naive CD4+ T cell, CD8 Effector Effector CD8+ T cell, CD8 Memory Memory CD8+ T cell, CD8 Naive Naive CD8+ T cell, CD14 Mono CD14 Monocytes, CD16 Mono CD16 Monocytes, CD56 bright NK CD56 bright natural killer, GMP Granulocyte-macrophage progenitors, HSC Hematopoietic stem cell, LMPP Lymphoid-primed multipotent progenitors, MAIT Mucosal-associated invariant T cell, NK Natural killer, Prog_B Progenitors of B cell, Prog_DC Progenitors of dendritic cell lineages, Prog_Mk Progenitor of megakaryocyte, Prog_RBC Progenitors of erythroid, Treg Regulatory T cell, cDC2 Type 2 conventional dendritic cell, gdT Gamma delta T cell, pDC Plasmacytoid dendritic cell). **b** UMAP visualization of WNN results on original cells and 613 metacells (a 50-fold reduction) inferred by MetaQ, SEACell, MetaCell V2, and SuperCell. **c** Compactness and separation of 613 metacells inferred by different methods, calculated separately for RNA and ADT modalities. Each boxplot ranges from the upper and lower quartiles with the median as the horizontal line and whiskers extend to 1.5 times the interquartile range. **d** Purity of 613 metacells inferred by different methods across RNA-informative (top panel) and ADT-informative (bottom panel) cell types. Each boxplot ranges from the upper and lower quartiles with the median as the horizontal line and whiskers extend to 1.5 times the interquartile range. Two-sided T-test results: $^*0.01 < p \leq 0.05$, $^{**}0.001 < p \leq 0.01$, $^{***}0.0001 < p \leq 0.001$, $^{****}p \leq 0.0001$. **e** PAGA cell embedding of metacells along the plasmablast developmental path inferred by MetaQ, with metacells colored by type and pseudotime, respectively. **f** Annotation and marker gene expression changes along the plasmablast developmental path. Source data are provided as a Source Data file.

with SEACell, while delivering approximately threefold time savings. In comparison, metacells inferred by the other two methods collapse into a small number of the most frequent cell types. Subsequently, we applied MetaQ to the paired RNA+ATAC multi-omics data. As shown in Supplementary Fig. 6c, when modeling peak data with Poisson distribution, MetaQ consistently outperforms the baseline methods. Moreover, we investigated the correlation between gene expression and chromatin accessibility within each metacell, leveraging peak-to-gene correspondences identified by Signac[36]. Supplementary Fig. 6d shows that MetaQ metacells achieve the highest Pearson correlation across the two omics, underscoring MetaQ's superior performance in aggregating and collaborating information from both omics. These results collectively highlight MetaQ as a powerful tool for scATAC-seq data analysis.

MetaQ facilitates single-cell batch integration

In addition to handling paired multi-omics data, we demonstrated that MetaQ is also effective in processing multi-batch data. Specifically, we evaluated MetaQ on the human pancreas dataset[37–41], which consists of

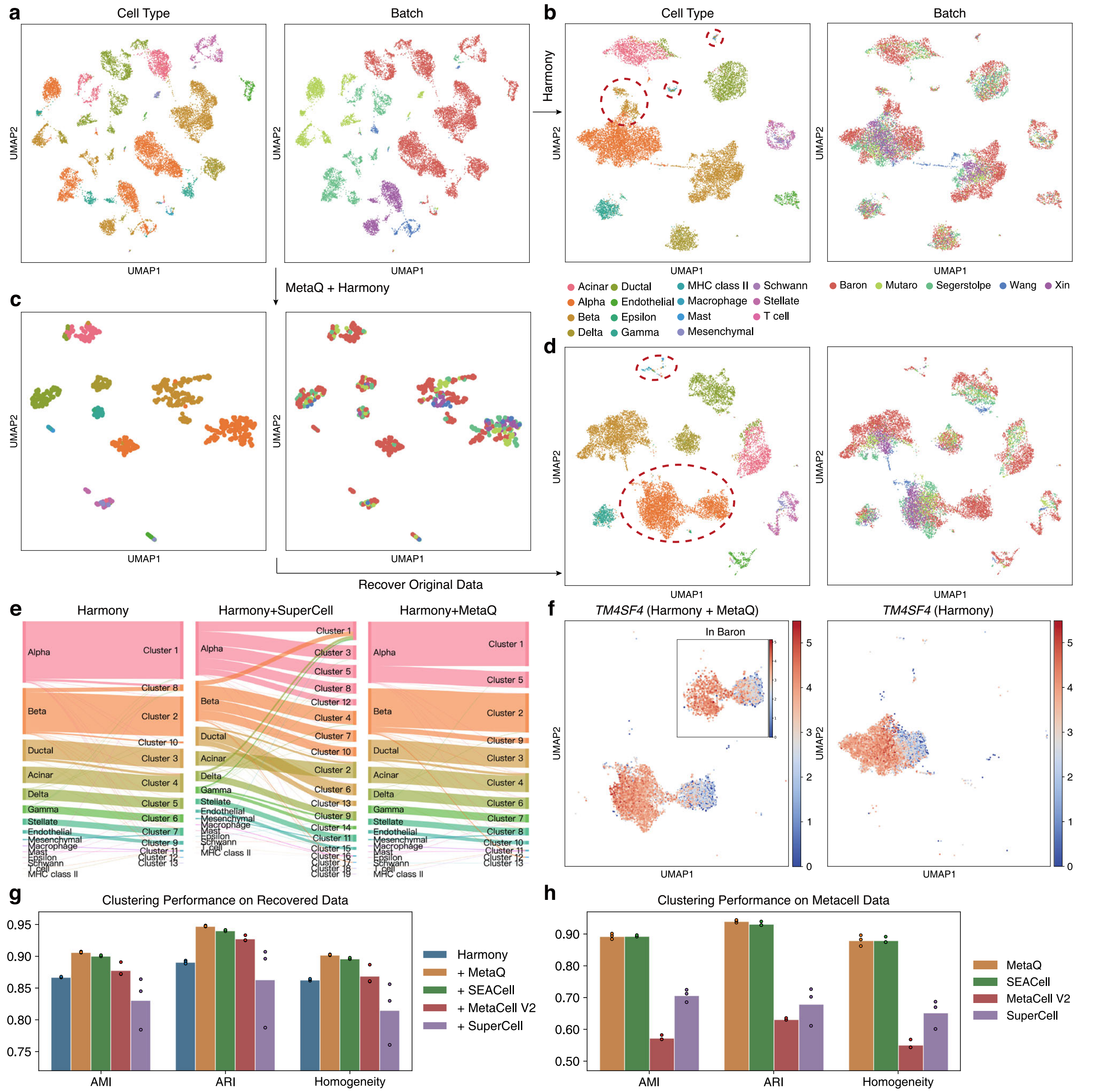


Fig. 4 | MetaQ facilitates batch integration. a UMAP visualization of the original 14,767 cells from the human pancreas dataset, with cells colored by types and batches, respectively. **b** UMAP visualization of integrated cell embeddings obtained by performing Harmony on the original data (MHC class II: Major histocompatibility complex Class II). **c** UMAP visualization of 590 MetaQ metacells (a 25-fold reduction) integrated by Harmony. **d** UMAP visualization of the integrated embeddings of original cells recovered by MetaQ. **e** Sankey plots showing Louvain cluster assignments on cell embeddings integrated by Harmony, recovered by SuperCell, and recovered by MetaQ. **f** Normalized expression of the marker gene *TM4SF4* projected on the UMAP plot of alpha cell embeddings by MetaQ and Harmony, respectively. The subplot in the left plot shows the results for the Baron batch. **g** AMI, ARI, and Homogeneity scores of Louvain clustering with three different resolutions of [0.5, 1.0, 2.0] on cell embeddings obtained by Harmony and recovered by MetaQ, SEACell, MetaCell V2, and SuperCell. **h** AMI, ARI, and Homogeneity scores of Louvain clustering with three different resolutions of [1.0, 2.0, 5.0] on 590 metacells inferred by different metacell algorithms. Source data are provided as a Source Data file.

14,767 cells from five different sources using four scRNA-seq protocols as visualized in Fig. 4a. Following the standard metacell inference and data integration pipeline, we first computed metacells using MetaQ and then applied the Harmony integration algorithm[42] to the inferred metacells. Fig. 4c shows promising batch mixing and cell type grouping, suggesting that single-cell-oriented batch integration methods are also suitable for metacells inferred by MetaQ. To provide a quantitative evaluation, we adopted the Louvain algorithm[43] to cluster batch-integrated metacells and mapped the cluster assignment of each metacell to the original cells it aggregates. The clustering AMI, ARI, and Homogeneity Score are illustrated in Fig. 4h, which shows that MetaQ and the best competitor SEACell outperform the other two baseline methods.

Beyond integrating metacells themselves, we further explored recovering the integrated embedding of original cells using metacell integration results. Specifically, we trained a simple neural network to map from raw data space to Harmony-integrated space, leveraging the original and integrated metacells. More details are provided in the *data integration and clustering* subsection. The trained network was then used to map the original cells to the integrated space, thereby recovering the integration results for single cells. To better integrate original cells via the mapping, the metacells should be highly prototypical of the corresponding cell populations, ensuring the mapping generalizes well from metacells to original cells. Additionally, the integrated metacells should contain batch effects as little as possible, ensuring the mapping can effectively correct the batch effects. In this context, the batch correction performance of the mapped original cells reflects not only the prototypicality of metacells, but also how metacell algorithms collaborate with batch integration methods. Thanks to the small number of metacells, such a mapping process only requires a few seconds. The integration results of original cells recovered by MetaQ are illustrated in Fig. 4d. Compared with the baseline result of directly performing Harmony on the original data (Fig. 4b), one could observe two apparent advantages of the MetaQ-recovered results highlighted by red circles in the figure. First, MetaQ alleviates the over-integration problem of Harmony, leading to better separation between cells of rare types. Second, MetaQ corrects a subset of beta cells that were falsely integrated with the alpha cells by Harmony. Intriguingly, we observed that the cell embeddings recovered by MetaQ form two sub-clusters within the alpha cells, both characterized by the canonical marker glucagon (*GCG*)[44] as shown in Supplementary Fig. 7a. To further investigate this phenomenon, Fig. 4f illustrates distinct expression patterns of *TM4SF4*[45]—a tetraspanin family member associated with pancreatic development—across these two subpopulations. Additionally, Supplementary Figs. 7b–7d demonstrate that the right subpopulation of alpha cells shows elevated expression of *NLRP1*, which nucleates inflammasomes[46], and *TNFRSF12A*[47,48], a member of the tumor necrosis factor receptor superfamily. Both genes are pivotal in mediating inflammatory responses. This observation also aligns with the elevated expression of chronic pancreatitis risk genes such as *PRSS1*[49]. These findings may suggest a potential involvement of this alpha cell subpopulation in the immune and inflammatory responses of the pancreas, indicating a broader functional spectrum beyond the traditional role in glucagon secretion regulation. Importantly, these results are not attributable to batch effects, as the distinct expression patterns between the two subclusters also exist within the Baron batch of data. In contrast, the Harmony integration results roughly aggregate all alpha cells together, thereby overlooking cell heterogeneity.

Finally, we compared the Louvain clustering results on cell embeddings computed by Harmony and those recovered by MetaQ and other baseline methods. As depicted in Fig. 4e, MetaQ achieves a generally consistent cluster partition with Harmony, while correcting the grouping of a subset of beta cells. Fig. 4g demonstrates that MetaQ outperforms other metacell algorithms, as well as Harmony on original cells, in all three clustering metrics. To evaluate the batch integration performance, we further computed the cLISI and iLISI metrics on the recovered cells in Supplementary Fig. 8a. As shown, cells recovered by MetaQ achieve the best or second-best performance in terms of the two metrics, outperforming harmony-integrated single cells in cell type grouping. These results demonstrate that MetaQ not only enhances batch integration at the metacell level, but can also be effectively incorporated with batch correction methods to improve performance on original single-cell data.

MetaQ is consistent with differential expression analysis

The above downstream tasks primarily assess the cell-level performance of metacell algorithms. Here, we extend our evaluation to the feature level. Specifically, we applied MetaQ to the human PBMC perturbation dataset[50], which comprises 240,090 immune cells of six types, three donors, and 144 perturbations, as depicted in Fig. 5a and Supplementary Fig. 9. Inspired by the pseudo-bulk operation, we inferred metacells within cells of the same type, donor, and perturbation, with a reduction rate of 10. These metacells were then concatenated across different groups to compute differential expression (DE) values concerning cell types and perturbations, respectively.

For the cell type DE analysis, we utilized metacells from different cell types within the negative control group. To evaluate how well MetaQ preserves feature-level characteristics, we compared the DE ranks of the most differential genes between the original cells and MetaQ metacells. As shown in Fig. 5b, MetaQ maintains high consistency with the original data in identifying top expressed genes. To quantitatively compare different metacell methods, we calculated Kendall's tau correlation to measure rank consistency between the original and metacell DE results. Fig. 5c demonstrates that MetaQ and SuperCell preserve gene expression patterns more effectively than the other two baselines.

In the perturbation DE analysis, we used metacells from the same cell type, including two positive controls, one negative control, and all 144 perturbations. The DE analysis was conducted independently for each cell type. We compared the Pearson correlation of DE values between the original data and metacells inferred by different methods. Fig. 5e reveals that MetaQ achieves the highest or second-highest correlation in five of six cell types, underscoring its superiority in preserving biological features. For a more intuitive understanding, we visualized the DE values for CD8+ T cells computed on original cells and metacells in Fig. 5d. Red rectangles highlight two instances where MetaQ outperforms baseline methods. On the one hand, SEACell incorrectly identifies a strong influence of the compound ABT-737, a Bcl-2 family inhibitor[51], on genes *NKG7*, *GZMA*, and *CCL5*. On the other hand, SuperCell underestimates the impact of Raloxifene on gene *POU2F2*, while overestimating its effect on gene *AC022706.1*. Overall, as depicted in Supplementary Figs. 10 and 11, the DE values of MetaQ metacells exhibit a high consistency with those of the original data, emphasizing MetaQ's promising ability to accurately summarize biological features.

MetaQ is a stable and robust algorithm for metacell inference

We performed a series of experiments on the human thyroid cancer dataset to evaluate the stability and robustness of the proposed MetaQ algorithm. To begin, we assessed the consistency of metacell assignments by testing MetaQ across varying numbers of metacells, corresponding to reduction rates ranging from 25 to 150. The agreement in metacell assignments across different reduction rates is depicted in Fig. 6b. Notably, in most instances, cells assigned to the same metacell at a lower reduction rate remain grouped together at a higher reduction rate. This indicates that metacells formed at higher reduction rates represent further aggregations of those formed at lower reduction rates. As shown in Fig. 6c, the homogeneity scores between metacell assignments consistently exceed 0.5 across all tested reduction rates, demonstrating the robustness of MetaQ against varying target metacell numbers.

Next, to examine the capacity of MetaQ in identifying rare cell types, we assigned each original cell the majority cell type of its corresponding metacell. The accuracy of such a majority-voted prediction reflects how well metacells cover different cell types. As shown in Fig. 6d, MetaQ achieves the highest balanced accuracy, outperforming existing metacell methods in rare cell type identification. Furthermore, we conducted subsampling experiments on the two rarest cell types, tumor-associated myeloid cell (TAMC) and parafollicular cell, to explore the minimum cell type frequency that could be captured under different reduction rates. As illustrated in Fig. 6e and Supplementary Fig. 12a, under the reduction rate of 50, MetaQ is the only method capable of accurately identifying cell types present at frequencies as low as 0.01%. Even when reducing the data size by 100, MetaQ still effectively captures cell types with frequencies above

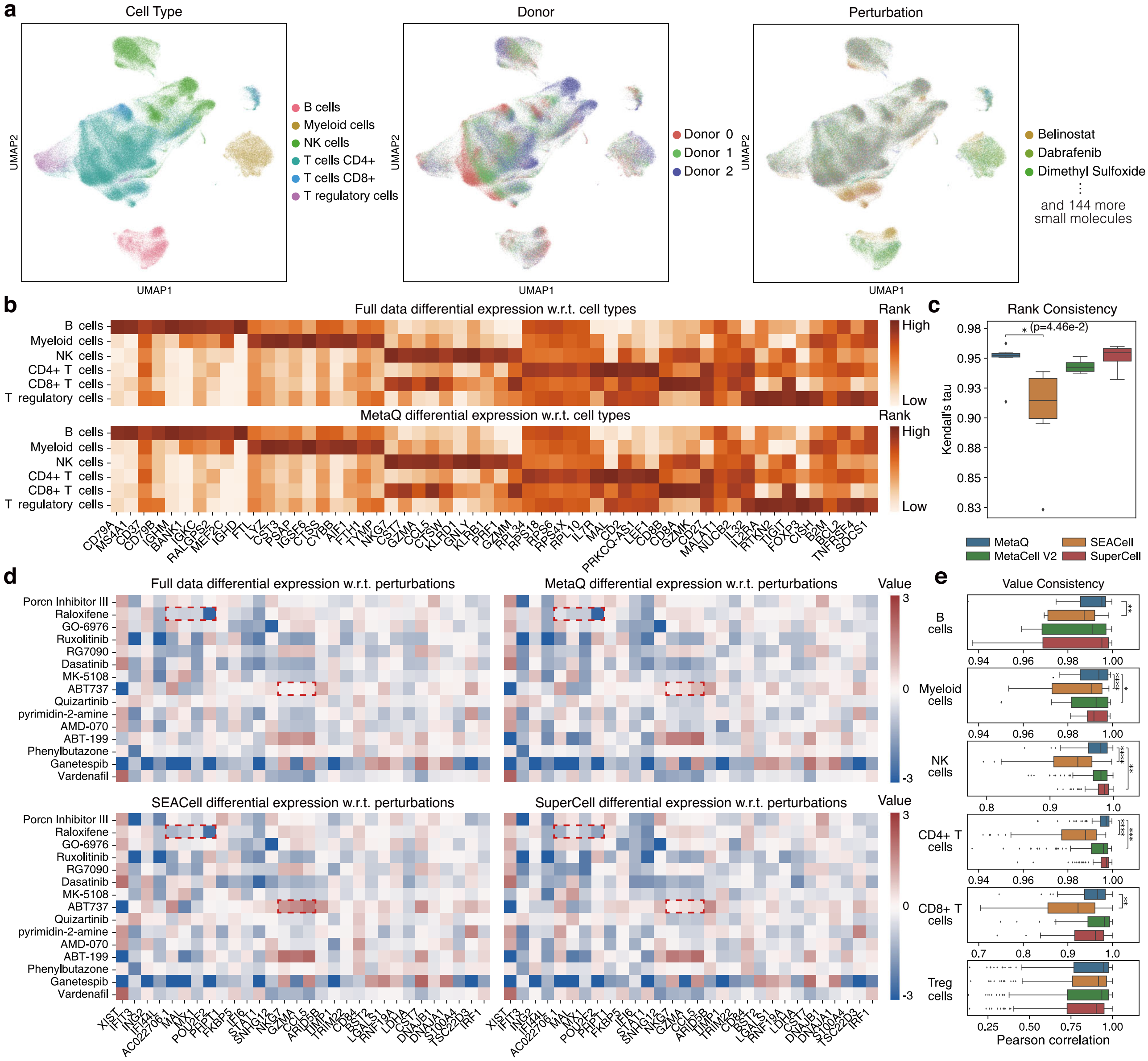


Fig. 5 | MetaQ preserves differential expressions with respect to cell types and perturbations. a UMAP visualization of the original 240,090 cells from human PBMC perturbation data, with cells colored by types, donors, and perturbations, respectively. **b** Rank of top differentially expressed genes concerning cell types, computed on the original cells (Top) and MetaQ metacells (Bottom) with a reduction rate of 10. **c** Consistency between the ranking of top 2000 differentially expressed genes computed on original cells and 24,009 metacells using different methods. Each boxplot ranges from the upper and lower quartiles with the median as the horizontal line and whiskers extend to 1.5 times the interquartile range. Two-sided T-test results: *0.01 < $p \le 0.05$. **d** Differential expression values relative to perturbations on CD8+ T cells, calculated on original cells and 24,009 metacells inferred by MetaQ, SEACell, and SuperCell. Perturbations and genes that differ the most between different methods are shown for clearer comparisons. **e** Pearson correlation between perturbation differential expression values computed on original cells and 24,009 metacells using different methods. Each boxplot ranges from the upper and lower quartiles with the median as the horizontal line and whiskers extend to 1.5 times the interquartile range. Two-sided T-test results: *0.01 < $p \le 0.05$, **0.001 < $p \le 0.01$, ***0.0001 < $p \le 0.001$, ****$p \le 0.0001$. Source data are provided as a Source Data file.

0.07%. These results demonstrate that MetaQ is a reliable tool for identifying rare cell types in metacell analysis.

Then, to evaluate the robustness of MetaQ against algorithmic configurations, we performed ablation studies on the discrete codebook, the core design in MetaQ for metacell assignments. Specifically, in addition to randomly initializing codebook entries by default, we experimented with two alternative initialization strategies, namely, Kmeans[52] and geometric sketching[53]. As shown in Fig. 6f, MetaQ maintains stable performance across different initialization strategies and random seeds. Supplementary Figs. 12b and 12c demonstrate that MetaQ performs consistently under both cosine and Euclidean distance measures between cell embeddings and codebook entries. Additionally, we conducted a parameter analysis on the momentum used in updating the historical codebook entry usage in Eq. (11). Fig. 6g illustrates that MetaQ is stable across momentum values ranging from 0.85 to 0.95 (with 0.9 as the default setting). However, when the momentum deviates significantly from this range, either toward lower or higher values, the entry usage update becomes too frequent or infrequent, hindering proper adjustment of over-large and over-small entries and ultimately resulting in degraded performance.

Lastly, as MetaQ requires manually setting the target metacell number, we provide a practical guideline for selecting an appropriate metacell number. Since MetaQ makes consistent metacell assignments across different reduction rates, we recommend simply setting the

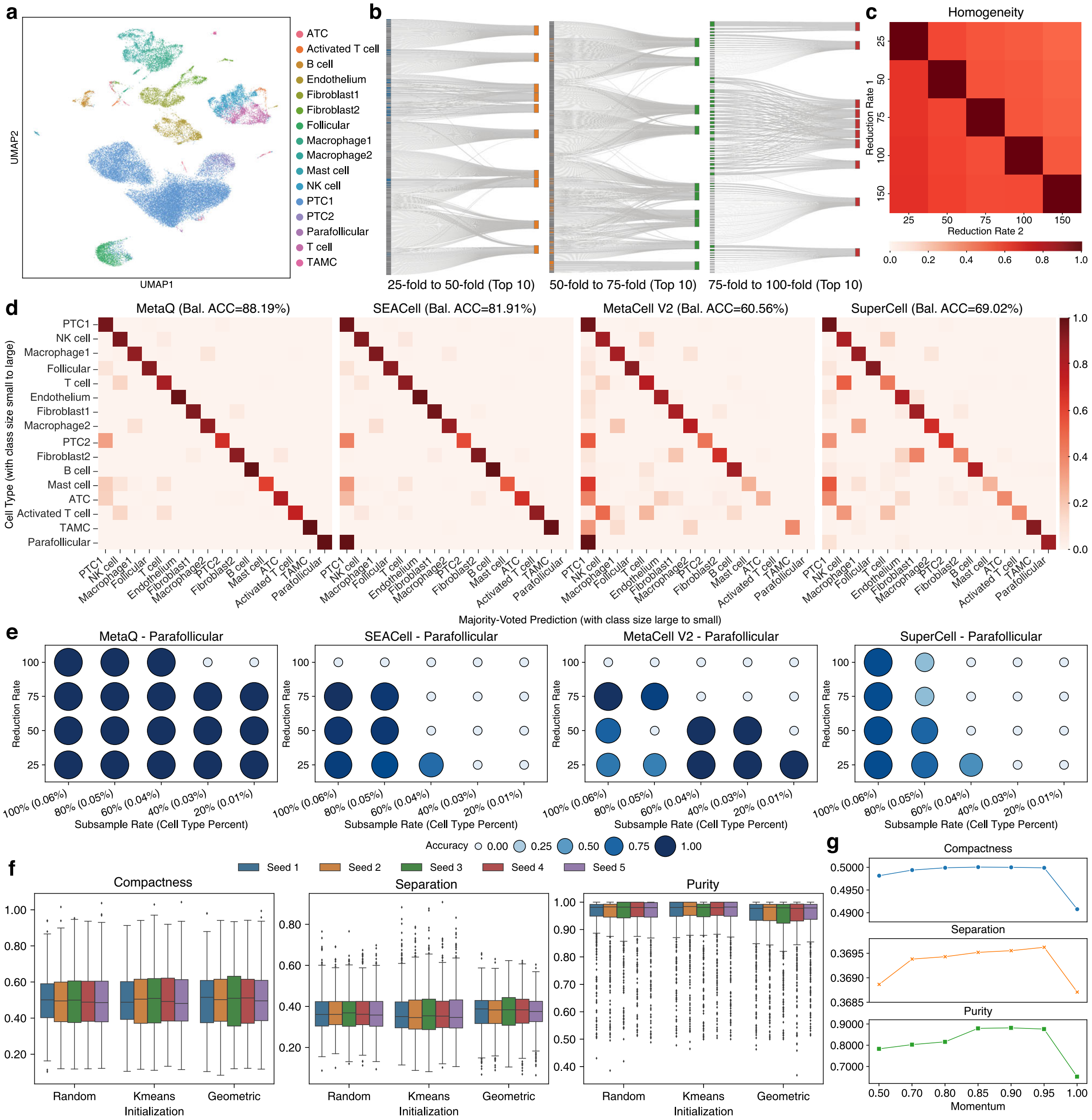


Fig. 6 | MetaQ is a stable and robust metacell algorithm. a UMAP visualization of the 46,205 cells from the human thyroid cancer dataset with cells colored by types (ATC Anaplastic thyroid cancer, NK cell Natural killer cell, PTC Papillary thyroid cancer, TAMC Tumor-associated myeloid cell). **b** Sankey plots showing the agreement of metacell assignments across different reduction rates. For clarity, only cells from the ten largest metacells at each reduction rate are visualized. **c** Homogeneity of metacell assignments across different reduction rates. **d** Agreement between ground-truth annotations and majority-voted labels within each of 462 metacells. Matrices with a clearer diagonal structure indicate higher metacell purity. **e** Accuracy on the rarest cell type, Parafollicular, of various metacell algorithms across different reduction and subsampling rates. **f** Compactness, separation, and purity of 462 MetaQ metacells under different codebook initialization strategies and random seeds. Each boxplot ranges from the upper and lower quartiles with the median as the horizontal line and whiskers extend to 1.5 times the interquartile range. **g** Compactness, separation, and purity of 462 MetaQ metacells across various momentum parameters. Source data are provided as a Source Data file.

metacell number to achieve a common 50-fold or 100-fold reduction when using MetaQ in practice. To find a more precise estimation of the metacell number that balances data compression and information preservation, we tested metacell numbers ranging from 50 to 1000 and tracked three algorithmic metrics, namely, the proportion of the original ($L_{\mathrm{NB}}/L_{\mathrm{Pois}}$) to the quantized ($L_{\hat{\mathrm{NB}}}/L_{\hat{\mathrm{Pois}}}$) reconstruction loss (referred to as reconstruction proportion), the difference in similarity between cells' closest and second-closest codebook entries (referred to as similarity difference), and the codebook loss (L_C). In parallel, we recorded two metacell quality metrics, including metacell purity and balanced accuracy, by assigning each original cell the majority cell type of its corresponding metacell. In addition to the discrete thyroid cancer data, we applied the same evaluation to the continuous bone marrow data. Subsampling was applied to the thyroid data to keep the original cell number consistent between the two datasets. As depicted in Supplementary Fig. 12d, the metacell quality improves progressively with the metacell number on both datasets. Notably, the thyroid cancer data, being more discrete, requires fewer metacells (~400) to reach the balanced accuracy plateau compared to the bone marrow data (~600), likely due to the greater diversity of cells along the continuous developmental path in the latter. Supplementary Fig. 12e indicates that compared to the codebook loss, the reconstruction proportion and similarity difference exhibit stronger correlations with metacell purity and balanced accuracy. Given that the reconstruction proportion tends to increase continuously with the number of metacells, we recommend first plotting the trend of similarity difference against the metacell number, and then selecting the point at which the decline in similarity difference plateaus as a more precise estimation.

Discussion

Towards the rapidly increasing volume of sequencing data, metacell methods provide a promising solution to reduce the computational burden by aggregating biologically similar cells. However, existing metacell algorithms, despite their intended purpose of alleviating computational complexity, are themselves computationally demanding and struggle to handle large-scale data. Essentially, these methods shift the computational bottleneck from downstream analysis to the metacell inference stage, sidestepping rather than ultimately solving the core issue.

MetaQ is a metacell algorithm that scales to arbitrarily large datasets, with linear time and constant memory costs relative to the cell number. Such superior efficiency and scalability set MetaQ apart from existing methods that suffer from exponential time or memory complexity. For instance, MetaQ achieves about 100 times speedup and 50 times memory savings when processing 100,000 cells, compared to the most competitive baseline SEACell.

The design of MetaQ is motivated by the cell differentiation process in multicellular organisms. Specifically, we conceptualize each metacell as a collective ancestor of a subgroup of specialized cells, which can thus effectively derive the latter. Following this idea, MetaQ employs a generative single-cell quantization approach to identify homogeneous cell subsets for metacell inference. Powered by the feature extraction capabilities of deep neural networks, MetaQ could precisely capture biological states, resulting in accurate and prototypical metacell construction. Extensive experiments demonstrate that even with significantly reduced computational complexity, MetaQ still achieves comparable, and in most cases slightly better, performance than existing metacell algorithms across various downstream tasks, including cell type annotation, developmental trajectory inference, batch integration, clustering, and differential expression analysis.

While current metacell algorithms are all designed for uni-omics data, MetaQ could easily extend to paired multi-omics analysis thanks to its simple yet effective design. By requiring the quantized cell embeddings to reconstruct all modalities, MetaQ is able to infer metacells for each modality while preserving pairing information. Such native support for multi-omics analysis aligns with the evolving capabilities of sequencing technologies, which increasingly enable simultaneous profiling of single cells across multiple layers.

Regarding the stability and generalizability of MetaQ, we simplified its hyper-parameters to avoid laborious tuning across different datasets. In this study, we fixed parameter configurations in all experiments and found that MetaQ consistently achieves promising results. In other words, users only need to specify the target number of metacells. Additionally, guidance on estimating the appropriate number of metacells is provided to facilitate practical application.

To further improve metacell analysis in future research, several promising avenues could be explored. First, while MetaQ currently learns cell embeddings using an autoencoder network, leveraging more advanced large-scale pre-trained models for single-cell data may improve feature extraction ability and, accordingly, the metacell quality. Second, MetaQ could currently handle various omics data, including gene expression, protein, and chromatin accessibility data. It is worth exploring its application in other omics, such as DNA methylation, by designing more appropriate modeling strategies. Third, while this paper demonstrates the effectiveness of MetaQ in inferring metacells on the continuous developmental data, it remains unexplored how MetaQ behaves on actual time-series sequencing data. Understanding how metacell algorithms could facilitate links between snapshots sampled at different time points presents an intriguing opportunity for future research. We anticipate that future developments could further enhance the performance, generalization capabilities, and applications of MetaQ, establishing it as a handy and powerful tool for metacell inference in the era of high-throughput profiling.

In conclusion, MetaQ is an efficient, scalable, and effective metacell algorithm that could be seamlessly incorporated into existing single-cell analysis pipelines. By reducing the number of cells while preserving biological characteristics, MetaQ enables existing single-cell analysis tools to handle arbitrarily large datasets, breaking the computational bottleneck. With the growing volume of cells and omics in high-throughput profiling data, we believe MetaQ will become a pivotal tool with broad applications across various downstream analyses.

Methods

The MetaQ algorithm

The inputs to MetaQ include the number of metacells $\hat{N}$ and the raw count matrix $X \in \mathbb{R}^{N \times M}$, where N and M denote the number of cells and features (e.g., genes, proteins, peaks), respectively. MetaQ views each metacell as a collective ancestor of a subgroup of specialized cells, which could effectively derive these homogeneous cells. To implement the idea, MetaQ quantizes all cells into a D-dimensional codebook $C \in \mathbb{R}^{\hat{N} \times D} (\hat{N} < N)$ consists of limited entries, aiming to reconstruct each cell using its most similar entry. For better reconstruction, cells with similar biological states would be quantized into the same entry. Consequently, each codebook entry intrinsically corresponds to a metacell, representing all cells it quantizes. MetaQ employs deep neural networks to perform the above cell quantization and reconstruction process, with further details provided below.

Count data modeling with the negative binomial and Poisson distribution. To endow deep neural networks with feature extraction capabilities, MetaQ first models the raw count matrix X using the negative binomial (NB) distribution[54–56] for gene expression and protein data (detailed derivations are provided in Supplementary Note 1), and Poisson distribution[57,58] for chromatin accessibility data. The count matrix is modeled by the two distributions using an auto-encoder. Specifically, for the i-th cell $x_i \in \mathbb{R}^M$, an encoder $f(\cdot)$ is first employed to learn the cell embedding e_i, followed by a decoder $g(\cdot)$ to estimate the mean $\mu_i \in \mathbb{R}^M$ and dispersion $r_i \in \mathbb{R}^M$ of the NB distribution, or the mean $\lambda_i \in \mathbb{R}^M$ of the Poisson distribution. The learning objective is to maximize the following distribution log-likelihoods:

$$\begin{aligned} L_{\mathrm{NB}} &= \frac{1}{N}\sum_{i=1}^{N} -\log\left(\mathrm{NB}(x_i|\mu_i, r_i)\right) \\ &= \frac{1}{N}\sum_{i=1}^{N} -\log\left[\frac{\Gamma(x_i+r_i)}{x_i!\Gamma(r_i)}\left(\frac{r_i}{r_i+\mu_i}\right)^{r_i}\left(\frac{\mu_i}{r_i+\mu_i}\right)^{x_i}\right], \end{aligned} \tag{1}$$

$$\mu_i = \mathrm{diag}(s_i) \times \exp\left(W_\mu d_i\right),\ r_i = \exp\left(W_r d_i\right),\ d_i = g(e_i), \tag{2}$$

$$L_{\mathrm{Pois}} = \frac{1}{N}\sum_{i=1}^{N} -\log\left(\mathrm{Pois}(x_i|\lambda_i)\right) = \frac{1}{N}\sum_{i=1}^{N} -\log\frac{\lambda_i^{x_i}\exp^{-\lambda_i}}{x_i!}, \tag{3}$$

$$\lambda_i = \mathrm{diag}(s_i) \times \exp\left(W_\lambda e_i\right), e_i = f(x_i), \tag{4}$$

where s_i represents the size factor each cell scaled to meet 10,000 counts during data preprocessing, W_μ, W_r, and W_λ are independent fully connected layers. Notably, rather than modeling the entire dataset with a single NB or Poisson distribution, MetaQ assigns distinct distribution parameters to each cell. While the decoder network $g(\cdot)$ is shared across all cells, their distribution parameters differ due to the unique embeddings e_i associated with each cell. Supplementary Figs. 6c and 6d demonstrate that modeling chromatin accessibility data with the Poisson distribution outperforms that with the negative binomial distribution, in both uni- and multi-omics scenarios.

Cell quantization with a discrete codebook. To discover biologically similar cells, MetaQ quantizes all cells into a discrete codebook C with $\hat{N}$ learnable entries. Specifically, given a cell embedding e_i, MetaQ employs the quantizer $q(\cdot)$ to assign it to the closest entry in the codebook, namely,

$$\hat{e}_i = q(e_i) = c_k,\ k = \underset{c_k \in C}{\operatorname{argmax}}\ \cos(e_i, c_k), \tag{5}$$

where c_k denotes the k-th entry in the codebook, which has the same dimensionality as e_i, and $\cos(\cdot,\cdot)$ refers to the cosine similarity.

The codebook entries are randomly initialized by default, given its simplicity and efficiency. Notably, MetaQ also supports alternative initialization strategies such as Kmeans[52] and geometric sketching[53]. Fig. 6f demonstrates that MetaQ performs consistently under different initialization strategies. After initialization, the codebook entries will be optimized through quantized cell reconstruction and adjusted with usage recording, as elaborated below.

Codebook optimization with quantized cell reconstruction. As compact proxies of the original cells, metacells ought to retain as much information from the original data as possible. In other words, a prototypical metacell is expected to effectively reconstruct a subgroup of homogeneous cells. To achieve this, MetaQ aims at reconstructing each original cell using its quantized cell embedding $\hat{e}_i$, with the following losses:

$$L_{\widehat{\mathrm{NB}}} = \frac{1}{N}\sum_{i=1}^{N} -\log\left(\mathrm{NB}\left(x_i|\hat{\mu}_i, \hat{r}_i\right)\right), \tag{6}$$

$$\hat{\mu}_i = \mathrm{diag}(s_i) \times \exp\left(\hat{W}_\mu \hat{d}_i\right),\ \hat{r}_i = \exp\left(\hat{W}_r \hat{d}_i\right),\ \hat{d}_i = \hat{g}(\hat{e}_i) \tag{7}$$

$$L_{\widehat{\mathrm{Pois}}} = \frac{1}{N}\sum_{i=1}^{N} -\log\left(\mathrm{Pois}\left(x_i|\hat{\lambda}_i\right)\right), \tag{8}$$

$$\hat{\lambda}_i = \mathrm{diag}(s_i) \times \exp\left(\hat{W}_\lambda \hat{e}_i\right), \tag{9}$$

where $\hat{W}_\mu$, $\hat{W}_r$, $\hat{W}_\lambda$ and $\hat{g}$ refer to a copy of the decoder parameters for the quantized cell reconstruction. The premise behind cell quantization is that biologically similar cells could be effectively reconstructed by the same entry in the codebook. In other words, the quantization operation naturally and intrinsically achieves the metacell assignment.

In addition to reconstructing original cell counts in the raw space, MetaQ further aligns codebook entries with their corresponding cell embeddings in the embedding space, namely,

$$L_{\mathrm{C}} = \frac{1}{N}\sum_{i=1}^{N} \|q(e_i) - \mathrm{sg}[e_i]\|_2^2, \tag{10}$$

where $\mathrm{sg}[\cdot]$ denotes the stop-gradient operator[59], which prevents the loss from influencing original cell embeddings. Otherwise, the cell embeddings might be disturbed when approximating randomly initialized codebook entries during the early stages of training. Aligning codebook entries with corresponding cell embeddings provides two primary benefits. First, it accelerates the convergence of the quantized cell reconstruction losses in Eqs. (1) or (3), by reducing the gap between the quantized and original data distributions. Second, it helps metacells identify more biologically similar cells, by guiding codebook entries toward biologically meaningful regions in the embedding space.

Codebook entry adjustment with usage recording. The above cell quantization strategy could discover metacells by learning a discrete codebook. However, it might encounter the error accumulation problem. Specifically, active entries frequently used to quantize cells would be optimized more often, increasing their likelihood of being selected for quantizing more cells. Conversely, inactive entries that are rarely used would be less or even never optimized, making them unlikely to represent other cells. As a result, only a portion of the discrete codebook would be effectively leveraged and optimized, leading to highly unbalanced metacell groupings. Biologically, when a metacell aggregates too few cells, it becomes more susceptible to technical noise and random fluctuations. Conversely, when a metacell represents too many cells, it may encompass diverse cell types or states, diluting the unique characteristics of a particular population.

To prevent such a degenerated solution, MetaQ records the usage of each codebook entry and adjusts those excessively large or small ones during the training process. Formally, the historical usage of entries is recorded with the exponential moving average as follows:

$$U_k^t = \eta \cdot U_k^{t-1} + (1-\eta)\cdot \frac{N_k^t}{N},\ U_k^0 = 0, \tag{11}$$

where U_k^{t-1} refers to the historical usage of entry c_k, N_k^t denotes the number of cells it quantizes in the t-th iteration, and η is the momentum parameter. Based on the recorded codebook usage, MetaQ first addresses the over-small entries by relocating them to the most distant cells, whose information is least captured by the current metacells. Specifically, the distance between the i-th cell and the codebook is defined by

$$d_i^s = \max_j \frac{\exp\left(1-\cos(e_i, c_j)\right)}{\sum_{k=1}^{\hat{N}} \exp\left(1-\cos(e_i, c_k)\right)}. \tag{12}$$

MetaQ randomly selects $\hat{N}$ cells as the target with the probability of $[d_1^s, \cdots, d_N^s]$, so that distant cells are more likely to be selected. After that, MetaQ updates the codebook entries by pushing them to the selected distant cells $E^s = [e_1^s, \cdots, e_{\hat{N}}^s]$, namely,

$$c_k^t = (1-\beta^s)\cdot c_k^{t-1} + \beta^s \cdot e_k^s,\ \beta^s = \exp(-100 \cdot U_k^t \cdot \hat{N} - \epsilon), \tag{13}$$

where ϵ is a small constant to stabilize the momentum update, and β^s ensures that over-small entries are updated more frequently to quantize more cells.

In addition to the over-small entries, some entries might be overly used during the quantization process. However, a single metacell cannot fully describe the biological states of all the corresponding cells, leading to inferior prototypicality of metacells. Therefore, we propose to disturb those over-large codebook entries, allowing a subset of cells they quantize to be taken over by other metacells. In practice, we found that reallocating codebook entries to the median-distance cells serves as a moderate and effective disturbance.

Specifically, MetaQ randomly selects one median-distance cell for each entry following the probability:

$$d_{ik}^{l}=\frac{\exp(-|\cos(e_i,c_k)-m_k|)}{\sum_{j=1}^{N}\exp(-|\cos(e_j,c_k)-m_k|)},\ m_k=\underset{i}{\mathrm{median}}\ \cos(e_i,c_k), \tag{14}$$

where d_{ik}^{l} denotes the probability of the i-th cell being selected by the k-th entry, and m_k represents the median distance between the k-th entry and cell embeddings. Let $E^l=[e_1^l,\cdots,e_{\hat{N}}^l]$, $e_k^l\sim[d_{1k}^l,\cdots,d_{Nk}^l]$ be the selected median-distance cells, MetaQ disturbs the codebook entries by:

$$c_k^t=(1-\beta^l)\cdot c_k^{t-1}+\beta^l\cdot e_k^l,\ \beta^l=\exp(-10\cdot\frac{\hat{N}}{N}\cdot\frac{1}{U_k^t}-\epsilon), \tag{15}$$

where ϵ is the same small constant as in Eq. (13), and β^l strengthens the disturbance on large codebook entries. By adjusting excessively large and small codebook entries, MetaQ is able to produce more balanced metacell assignments, ensuring that each metacell captures the biological states for a moderate number of cells.

By combining Eqs. (1), (3), (6), (8), and (10), the overall objective function of MetaQ lies in the form of

$$L_{\mathrm{MetaQ}}=\begin{cases}L_{\mathrm{NB}}+L_{\hat{\mathrm{NB}}}+L_{\mathrm{C}}, & \text{for gene expression and protein data,}\\ L_{\mathrm{Pois}}+L_{\hat{\mathrm{Pois}}}+L_{\mathrm{C}}, & \text{for chromatin accessibility data.}\end{cases} \tag{16}$$

The above objective simultaneously optimizes the parameters of the encoder $f(\cdot)$, the decoders $g(\cdot),\hat{g}(\cdot)$, $W_\mu(\hat{W}_\mu)$, $W_r(\hat{W}_r)$, $W_\lambda(\hat{W}_\lambda)$, and the codebook C via gradient descent. Furthermore, the codebook would be adjusted via Eq. (13) and (15) in every iteration.

Metacell inference with cell quantization results. After training, each cell would be quantized into one of the codebook entries. To derive the metacell count matrix $\hat{X}\in\mathbb{R}^{\hat{N}\times M}$, MetaQ simply averages the raw count value of cells quantized into the same entry, as these cells are likely to have similar features. Formally, the i-th metacell $\hat{x}_i$ is computed as

$$\hat{x}_i=\frac{1}{\hat{N}_i}\sum_{j=1}^{N}x_j,\ s.t.\ q(e_j)=c_i, \tag{17}$$

where $\hat{N}_i$ denotes the number of cells quantized into the i-th codebook entry.

Implementation details. MetaQ is implemented in Python using the PyTorch[60] framework, v.2.1.1. The encoder network $f(\cdot)$ is a fully connected network (FCN) consisting of three layers—an input layer, a hidden layer, and an output layer with 512, 128, and 32 neurons, respectively. Each of the input M features is connected to all neurons in the input layer, and each subsequent neuron is fully connected to the neurons in the next layer. The decoder networks $g(\cdot)$ and $\hat{g}(\cdot)$ are similarly structured FCNs with two layers of 128 and 512 neurons, respectively. The 32-dimensional cell embedding connects to all neurons in the first layer, and each neuron is further connected to all the neurons in the second layer. To estimate the NB distribution, two subsequent one-layer FCNs $W_\mu(\hat{W}_\mu)$ and $W_r(\hat{W}_r)$ project the 512-dimensional feature to M-dimensional mean μ and dispersion r parameters. For the Poisson distribution, a one-layer FCN $W_\lambda(\hat{W}_\lambda)$ projects cell embeddings to the M-dimensional mean parameter λ. In all experiments, we trained MetaQ for 300 epochs using the Adam[61] optimizer with a learning rate of 1e − 3 and a weight decay of 1e − 2. In addition to the joint optimization with network parameters through standard gradient descent[62], the codebook C is further updated via Eqs. (13) and (15) at each mini-batch. We fixed the momentum parameter η = 0.9 and small constant ϵ = 1e − 3 for all datasets. To expedite training, we early stopped the optimization when the changes in losses $L_{\hat{\mathrm{NB}}}$ (for RNA and ADT data), $L_{\hat{\mathrm{Pois}}}$ (for ATAC data), and L_{C} were less than 1e − 5 for ten consecutive epochs. All experiments were conducted on an NVIDIA RTX 3090 GPU with CUDA 12.2 on the Ubuntu 20.04 OS.

Handling multi-omics data. In the preceding sections, we introduced MetaQ on uni-omics data for clarity. The design of MetaQ naturally supports metacell inference for paired multi-omics data. As illustrated in Supplementary Fig. 1, MetaQ makes two primary adjustments to accommodate multi-omics data. First, MetaQ concatenates the multi-omics information when computing the cell embeddings. Second, MetaQ requires the quantized cell embeddings to reconstruct the original count matrices across all modalities. Specifically, let $X^1,X^2,\ldots,X^T(X^j\in\mathbb{R}^{N\times M^j})$ be the paired multi-omics data of T modalities, MetaQ first extracts the feature of each modality and then concatenate them to form the cell embedding, namely,

$$e_i'=\mathrm{concat}(e_i^1,e_i^2,\ldots,e_i^T),\ e_i^j=f^j(x_i^j), \tag{18}$$

where $e_i'\in\mathbb{R}^{N\times T\cdot D}$ is the multi-omics embedding of the i-th cell, x_i^j and e_i^j denote its raw count and embedding in the j-th modality, respectively. Moreover, $f^j(\cdot)$ refers to the encoder for modality j trained with L_{NB}^j or L_{Pois}^j consistent with Eqs. (1) and (3). The quantized cell embedding $\hat{e}_i=q(e_i')$ is expected to reconstruct all modalities by minimizing $L_{\hat{\mathrm{NB}}}^j$ or $L_{\hat{\mathrm{Pois}}}^j, j\in[1,T]$ consistent with Eqs. (6) and (8). The codebook entry adjustment strategy in Eqs. (13) and (15) remains the same as for uni-omics data, with the codebook size extending to $\hat{N}\times T\cdot D$ catering to the multi-omics cell embeddings.

In summary, the overall objective function of MetaQ for multi-omics data lies in the form of

$$L_{\mathrm{MetaQ}}'=L_{\mathrm{C}}+\sum_{j=1}^{T}\begin{cases}L_{\mathrm{NB}}^j+L_{\hat{\mathrm{NB}}}^j, & \text{if the j}-\text{th omics is gene expression or protein ,}\\ L_{\mathrm{Pois}}^j+L_{\hat{\mathrm{Pois}}}^j, & \text{if the }j-\text{th omics is chromatin accessibility.}\end{cases} \tag{19}$$

After training, MetaQ averages the raw counts of cells from the same codebook entry in each modality according to Eq. (17), resulting in the paired multi-omics metacells for downstream analyses.

Data preprocessing

To preprocess the input raw count matrix, we first normalized each cell by dividing each count against its total number of counts, then multiplied the counts by 10,000 to standardize total counts across cells. After that, we log normalized the counts and scaled the data to have unit variance and zero mean. The detailed preprocessing steps for each dataset are elaborated below:

- Human fetal atlas data. The human fetal atlas data was downloaded from NCBI GEO accession number GSE156793[2], including the raw gene expression and cell-type information. We preprocessed the data following the previous work scJoint[9]. To construct relatively balanced data, for cell type k with number of cells $n_k>10{,}000$, we subsampled $\max\{0.05\cdot n_k, 10,000\}$ cells. All cells were kept for cell types with less than 10,000 cells, resulting in 433,695 cells of 54 cell types. Data upsampling was performed when evaluating the running time and memory costs of metacell algorithms.
- Human bone marrow data. The human bone marrow data (GSE128639[27]) was downloaded with the SeuratData package[27], v.0.2.2.9001. We used the *InstallData* function to download the *bmcite* dataset, which includes 30,672 scRNA-seq profiles and a panel of 25 antibodies. The cell type information was obtained from the *meta.data$celltype.l2* field of the Seurat object.

- Mouse kidney data. The gene expression and peak-by-cell matrix were downloaded from https://www.10xgenomics.com/resources/datasets/mouse-kidney-nuclei-isolated-with-chromium-nuclei-isolation-kit-saltyez-protocol-and-10x-complex-tissue-dp-ct-sorted-and-ct-unsorted-1-standard[34], which includes 14,527 cells with 20,105 genes and 32,285 peaks. The cell types were manually annotated according to the reported cell-type markers[1].
- Human pancreas data. The human pancreas dataset was downloaded from https://hemberg-lab.github.io/scRNA.seq.datasets/human/pancreas. The data was generated using four different scRNA-seq protocols from five different sources, including inDrop (GSE84133, 8569 cells)[37], CEL-Seq2 (GSE85241, 2122 cells)[38], Smart-Seq2 (E-MTAB-5061, 2127 cells)[39], and SMARTer (GSE83139, 457 cells and GSE81608, 1492 cells)[40,41]. The sequencing data from five experiments was concatenated by keeping commonly detected genes. Cells annotated as "unclear", "co-expression", "not applicable", "unclassified", "unclassified endocrine", "dropped", "alpha.contaminated", "beta.contaminated", "delta.contaminated", or "gamma.contaminated" were removed. Cell type annotations "activated_stellate", "PSC (Pancreatic Stellate Cell)", and "quiescent_stellate" were renamed to "Stellate", while "mesenchyme" cells were renamed to "Mesenchymal". The above preprocessing results in 14,767 cells of 15 different cell types.
- Human PBMC perturbation data. The human PBMC perturbation data[50] was downloaded from https://www.kaggle.com/competitions/open-problems-single-cell-perturbations/data. The publicly accessible training split was used, including 240,090 cells of six cell types, 144 compounds as perturbations, two positive controls Dabrfenib and Belinostat, and one negative control DMSO. More specially, B and Myeloid cells include 15 compounds, while T cells (CD4+, CD8+, regulatory) and NK cells include all 144 compounds.
- Human thyroid cancer data. The human thyroid cancer data[63] was downloaded from https://ngdc.cncb.ac.cn/gsa-human/browse/HRA000686. This dataset comprises single-cell sequencing of thyroid cancer samples from one normal thyroid tissue, three anaplastic thyroid cancer (ATC), and three papillary thyroid cancer (PTC) cases, encompassing a total of 46,205 cells of 16 different cell types.

Performance and benchmarking

Baseline methods. Four existing metacell algorithms were benchmarked for comparisons, including SEACell[19], MetaCell V2[20], SuperCell[6], and EpiCarousel[35].

For SEACell, we used its Python package (https://github.com/dpeerlab/SEACells), v.0.3.3. Following its official tutorial, we used the *SEACells*, *construct_kernel_matrix*, *initialize_archetypes*, and *fit* functions to build and fit the model. After training, we inferred the metacell assignments through the*summarize_by_SEACell* function. We kept the default parameters except for *n_SEACells*, which was tuned to adjust the number of metacells.

For MetaCell V2, we used its Python package (https://github.com/tanaylab/metacells), v.0.9.4. Notably, the algorithm itself does not directly support specifying the number of metacells. Thus, for fair comparisons, we searched for the *target_metacell_umis* parameter as suggested in its tutorial to approximate the expected metacell number, with other parameters set as the default. The *divide_and_conquer_pipeline* and *collect_metacells* functions were utilized to infer the metacell grouping. In practice, we also found that in some cases MetaCell V2 fails with the default parameters. As a solution, among its hundreds of tunable parameters, we manually tuned the *min_metacell_size*, *quality_min_gene_total*, *target_metacell_size*, and *project_min_significant_gene_umis* values until the algorithm gives proper outputs.

For SuperCell, we used its official R package (https://github.com/GfellerLab/SuperCell), v.1.0. Following its default pipeline, we first constructed a k-nearest neighbor single-cell network and then merged densely connected cells to infer metacell membership, using the *SCimplify* function. All parameters were set as default except for *gamma*, which was tuned to adjust the number of metacells.

For EpiCarousel, we used its Python package (https://github.com/BioX-NKU/EpiCarousel/tree/main), v.0.0.8. Notably, as EpiCarousel is designed for scATAC-seq data, we evaluated it on the mouse kidney dataset. Following its official tutorial, we first used the *data_split* function to partition data into chunks, and then used the *identify_metacells* and *merge_metacells* functions to compute metacells. We kept the default parameters except for *carousel_resolution*, which was tuned to adjust the metacell number.

Cell type classification. To classify Human fetal atlas cells, we used the inferred metacells to train a classification network. The network is an FCN with the dimension of M-512-128-K, where M and K denote the number of input features and cell types, respectively. We adopted the following cross-entropy loss to train the network:

$$L_{\mathrm{CE}} = \frac{1}{\hat{N}}\sum_{i=1}^{\hat{N}} -\log\left(\frac{\exp\left(p_i[\hat{y}_i]\right)}{\sum_{k=1}^{K}\exp\left(p_i[k]\right)}\right), \tag{20}$$

where p_i denotes the predicted soft label of the i-th metacell whose label $\hat{y}_i$ is given by the majority of original cells it represents. The network was trained for 50 epochs with a batch size of 512, by the Adam optimizer with default parameters.

We used the network trained on the metacells to classify all original cells. To evaluate the performance, we computed the classification accuracy and balanced accuracy score[64,65] defined as

$$\mathrm{ACC} = \frac{1}{N}\sum_{i=1}^{N}\delta(\tilde{y}_i, y_i), \ \ \delta(a,b) = \begin{cases} 1 & \text{if a} = \text{b}, \\ 0 & \text{otherwise}, \end{cases} \tag{21}$$

$$\text{Balanced ACC} = \frac{1}{\sum_i w_i}\sum_{i=1}^{N}\delta(\tilde{y}_i, y_i) w_i, \ \ w_i = \frac{1}{\sum_j \delta(y_j, y_i)}, \tag{22}$$

where $\tilde{y}_i$ and y_i denote the predicted and ground-truth annotation of the i-th cell. The balanced accuracy score accounts for the cell type distribution, highlighting the classification performance on rare types.

Metacell compactness and separation. To evaluate the homogeneity of cells within each metacell and the heterogeneity of cells across different metacells, we introduced the following compactness and separation metrics:

$$\text{Compactness} = \frac{\hat{N}}{N}\sum_{i\in\mathbb{M}}\frac{1}{|\mathbb{M}|}\sum_{j\in\mathbb{M}} s(x_i, x_j), \tag{23}$$

$$\text{Separation} = \frac{\hat{N}}{N}\sum_{i\in\mathbb{M}}\operatorname*{argmin}_{j\notin\mathbb{M}}\left[1 - s(x_i, x_j)\right], \tag{24}$$

where N, $\hat{N}$ are the numbers of original cells and metacells, $\mathbb{M}$ denotes the index set of cells grouped into the same metacell, and $s(\cdot,\cdot)$ refers to the Pearson correlation coefficient ranging from $[-1, 1]$. Here, we chose Pearson correlation in the raw space as the cell similarity measure, to avoid the influence of different dimensional reduction techniques employed by various methods. Additionally, we accounted for metacell sizes by keeping the magnitude $|\mathbb{M}|$ in the outer summation, thereby mitigating the potential bias arising from extremely imbalanced metacell assignments. For example, assigning $|\mathbb{M}| - 1$ cells into $|\mathbb{M}| - 1$ metacells in a one-to-one fashion, while grouping all remaining cells into a single metacell would artificially inflate metric scores without truly improving metacell grouping. Lastly, the factor of $\hat{N}/N$ was included to account for the magnitude differences across varying numbers of metacells. Higher values of both metrics indicate a more effective metacell grouping.

Multi-omics analysis and trajectory inference. For paired multi-omics analysis on human bone marrow data, we applied the WNN integration algorithm[27] implemented by the muon[66] Python package, v.0.1.5, to the inferred metacells. The neighboring information was then passed to the PAGA algorithm[28] provided in the scanpy[17] Python package, v.1.9.6, for trajectory inference. A random hematopoietic stem cell was set as the root for the developmental trajectory. The peak-to-gene correspondence on the mouse kidney data was obtained by Signac[36], v1.8.0.

Data integration and clustering. To integrate human pancreas data, we adopted the official harmonypy[42] Python package, v.0.0.6. After correcting batch effects in metacells, we built a neural network to learn the mapping from the raw space to the batch-corrected PCA space. The mapping network $m(\cdot)$ is of the dimension of M-256-50, where M refers to the number of input features and 50 is the default PCA dimension suggested by Harmony. We trained the network by minimizing the following mean squared error:

$$L_{\mathrm{MSE}} = \frac{1}{\hat{N}}\sum_{i=1}^{\hat{N}} \| m(\hat{x}_i) - \hat{z}_i\|_2^2, \tag{25}$$

where $\hat{z}_i$ denotes the Harmony integrated PCA embedding of the i-th metacell. The network was trained for 1000 epochs with a batch size of 512 by the Adam optimizer. After that, we mapped original cells via $z_i = m(x_i)$, where z_i corresponds to the batch-corrected embedding of the i-th cell.

To cluster batch-corrected data, we adopted the Louvain clustering algorithm[43] provided in the scanpy python package, v.1.9.6. The following AMI[67], ARI[68], and Homogeneity Score[69] metrics were used to evaluate the clustering performance:

$$\mathrm{AMI} = \frac{MI(U,V) - E\{MI(U,V)\}}{\max\{H(U),H(V)\} - E\{MI(U,V)\}}, \tag{26}$$

$$\begin{aligned} MI(U,V) &= \sum_{p=1}^{K'}\sum_{q=1}^{K}|U_p \cap V_q| \log\frac{N|U_p \cap V_q|}{|U_p| \times |V_q|}, \\ H(U) &= -\sum_{p=1}^{K'}\frac{|U_p|}{N}\log\frac{|U_p|}{N}, \quad H(V) = -\sum_{q=1}^{K}\frac{|V_q|}{N}\log\frac{|V_q|}{N} \end{aligned} \tag{27}$$

where $MI(U, V)$ is the mutual information between the cluster assignments U and ground-truth labels V, $H(U)$, $H(V)$ are the entropies, and K', K refers to the number of clusters and cell types, respectively.

$$\mathrm{ARI} = \frac{\sum_{p=1}^{K'}\sum_{q=1}^{K}\binom{|U_p \cap V_q|}{2} - \left[\sum_{p=1}^{K'}\binom{|U_p|}{2}\sum_{q=1}^{K}\binom{|V_q|}{2}\right]\Big/\binom{N}{2}}{\frac{1}{2}\left[\sum_{p=1}^{K'}\binom{|U_p|}{2} + \sum_{q=1}^{K}\binom{|V_q|}{2}\right] - \left[\sum_{p=1}^{K'}\binom{|U_p|}{2}\sum_{q=1}^{K}\binom{|V_q|}{2}\right]\Big/\binom{N}{2}}. \tag{28}$$

where $\binom{n}{2} = n(n-1)/2$ refers to the number of pairs in n samples.

$$\text{Homogeneity Score} = 1 - \frac{H(V|U)}{H(V)}, \tag{29}$$

$$H(V|U) = -\sum_{p=1}^{K'}\frac{|U_p|}{N}\sum_{q=1}^{K}\frac{|U_p \cap V_q|}{|U_p|}\log\frac{|U_p \cap V_q|}{|U_p|}, \tag{30}$$

where $H(V|U)$ is the conditional entropy of the ground-truth labels given the cluster assignments, and the entropy $H(V)$ is defined the same as in Eq. (27). A larger value of the three metrics indicates better agreements between the cluster assignments and ground truth labels, namely, a better clustering result.

Moreover, we employed the cLISI and iLISI metrics introduced in Harmony[42] to evaluate the batch integration performance. For each cell, the two metrics were computed by:

$$\mathrm{cLISI} = \frac{1}{\sum_{q=1}^{K} p(q)}, \quad \mathrm{iLISI} = \frac{1}{\sum_{b=1}^{B} p(b)}, \tag{31}$$

where B denotes the number of batches, and $p(q)$, $p(b)$ refer to the cell type and batch probabilities in the Gaussian kernel-based neighborhood distributions with a perplexity of 30. To balance the significance of major and rare cell types, we averaged the two metrics within each cell type. The original cLISI and iLISI range in $[1, K]$ and $[1, B]$, respectively. For clarity, we normalized them to [0, 1] and reported the 1 - cLISI and iLISI values. In other words, a higher 1 - cLISI value indicates more accurate cell type grouping, while a higher iLISI value indicates better batch mixing.

Differential expression analysis. To compute the differential expression (DE) for the human PBMC perturbation data, we used the *rank_genes_groups* function with Wilcoxon rank-sum test[70], provided in the scanpy[17] package, v.1.9.6. We conducted DE analyses with respect to cell types and compound perturbations respectively, with the *logfoldchanges* values reported. The cell type DE was computed on cells from the negative control, while the perturbation DE was calculated on each cell type independently.

Visualization. We used the *umap* function provided in the scanpy[17] Python package, v.1.9.6, to reduce the dimension to two for cell and metacell visualization. Boxplots, heatmaps, lineplots, and barplots were illustrated using the seaborn[71] Python package, v.0.12.2.

Statistics & reproducibility. Statistical analyses were performed by the SciPy Python package[72], v1.11.3. The p-value was determined by the two-sided T-test, and the p-value < 0.05 is considered statistically significant. Experiments are conducted under five randomizations with different random seeds. No statistical method was used to predetermine the sample size. We subsampled over-large cell types to construct relatively balanced data for the human fetal atlas and kept commonly detected genes across different batches for the human pancreas data. No other data were excluded from the analyses. The investigators were not blinded to allocation during experiments and outcome assessment.

Reporting summary

Further information on research design is available in the Nature Portfolio Reporting Summary linked to this article.

Data availability

All datasets used in this work are publicly available. The human fetal atlas data[2] used in this study are available in the GEO database under accession code GSE156793. The human bone marrow data[27] used in this study are available in the GEO database under accession code GSE128639, which could be downloaded with the SeuratData package from https://github.com/satijalab/seurat-data. The mouse kidney data[34] are available at https://www.10xgenomics.com/resources/datasets/mouse-kidney-nuclei-isolated-with-chromium-nuclei-isolation-kit-saltyez-protocol-and-10x-complex-tissue-dp-ct-sorted-and-ct-unsorted-1-standard. The human pancreas data used in this study are available in the GEO database under accession codes GSE84133[37] [https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE84133], GSE85241[38] [https://www.ncbi.nlm.nih.gov/geo/query/acc.

cgi?acc=GSE85241, E-MTAB-5061[39] [https://www.ebi.ac.uk/biostudies/arrayexpress/studies/E-MTAB-5061], GSE83139[40] [https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE83139], and GSE81608[41] [https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE81608], which are also accessible on https://hemberg-lab.github.io/scRNA.seq.datasets/human/pancreas. The human PBMC perturbation data[50] are available at the Kaggle competition site https://www.kaggle.com/competitions/open-problems-single-cell-perturbations/data. The human thyroid cancer data[63] are available under restricted access due to sharing principles in the Genome Sequence Archive under accession code HRA000686. The access can be obtained following the official data application guideline at https://ngdc.cncb.ac.cn/gsa-human/document/GSA-Human_Request_Guide_for_Users_us.pdf, which provides detailed instructions on how to submit a data access request, along with the criteria for approval. The expected timeframe for response to access requests is typically within four weeks. Source data are provided with this paper.

Code availability

The code used to develop the model and generate results in this study is publicly available and has been deposited in GitHub at https://github.com/XLearning-SCU/MetaQ, under MIT license. The specific version of the code associated with this publication is archived in Zenodo and is accessible via https://doi.org/10.5281/zenodo.14271480[73].

References

1. Consortium, T. M. et al. Single-cell transcriptomics of 20 mouse organs creates a tabula muris. *Nature* **562**, 367–372 (2018).
2. Cao, J. et al. A human cell atlas of fetal gene expression. *Science* **370**, eaba7721 (2020).
3. Azizi, E. et al. Single-cell map of diverse immune phenotypes in the breast tumor microenvironment. *Cell* **174**, 1293–1308 (2018).
4. La Manno, G. et al. Rna velocity of single cells. *Nature* **560**, 494–498 (2018).
5. Stuart, T. et al. Comprehensive integration of single-cell data. *Cell* **177**, 1888–1902 (2019).
6. Bilous, M. et al. Metacells untangle large and complex single-cell transcriptome networks. *BMC Bioinforma.* **23**, 336 (2022).
7. Arisdakessian, C., Poirion, O., Yunits, B., Zhu, X. & Garmire, L. X. Deepimpute: an accurate, fast, and scalable deep neural network method to impute single-cell rna-seq data. *Genome Biol.* **20**, 1–14 (2019).
8. Eraslan, G., Simon, L. M., Mircea, M., Mueller, N. S. & Theis, F. J. Single-cell rna-seq denoising using a deep count autoencoder. *Nat. Commun.* **10**, 390 (2019).
9. Lin, Y. et al. scjoint integrates atlas-scale single-cell rna-seq and atac-seq data with transfer learning. *Nat. Biotechnol.* **40**, 703–710 (2022).
10. Zhao, J. et al. Adversarial domain translation networks for integrating large-scale atlas-level single-cell datasets. *Nat. Computational Sci.* **2**, 317–330 (2022).
11. Li, Y. et al. scbridge embraces cell heterogeneity in single-cell rna-seq and atac-seq data integration. *Nat. Commun.* **14**, 6045 (2023).
12. Tian, T., Wan, J., Song, Q. & Wei, Z. Clustering single-cell rna-seq data with a model-based deep learning approach. *Nat. Mach. Intell.* **1**, 191–198 (2019).
13. Tian, T., Zhang, J., Lin, X., Wei, Z. & Hakonarson, H. Model-based deep embedding for constrained clustering analysis of single cell rna-seq data. *Nat. Commun.* **12**, 1–12 (2021).
14. Liu, Q., Chen, S., Jiang, R. & Wong, W. H. Simultaneous deep generative modelling and clustering of single-cell genomic data. *Nat. Mach. Intell.* **3**, 536–544 (2021).
15. Hu, J. et al. Iterative transfer learning with neural network for clustering and cell type classification in single-cell rna-seq analysis. *Nat. Mach. Intell.* **2**, 607–618 (2020).
16. Yang, F. et al. scbert as a large-scale pretrained deep language model for cell type annotation of single-cell rna-seq data. *Nat. Mach. Intell.* **4**, 852–866 (2022).
17. Wolf, F. A., Angerer, P. & Theis, F. J. Scanpy: large-scale single-cell gene expression data analysis. *Genome Biol.* **19**, 1–5 (2018).
18. Baran, Y. et al. Metacell: analysis of single-cell rna-seq data using k-nn graph partitions. *Genome Biol.* **20**, 1–19 (2019).
19. Persad, S. et al. Seacells infers transcriptional and epigenomic cellular states from single-cell genomics data. *Nat. Biotechnol.* **41**, 1746–1757 (2023).
20. Ben-Kiki, O., Bercovich, A., Lifshitz, A. & Tanay, A. Metacell-2: a divide-and-conquer metacell algorithm for scalable scrna-seq analysis. *Genome Biol.* **23**, 100 (2022).
21. Pons, P. & Latapy, M. Computing communities in large networks using random walks. In *Computer and Information Sciences-ISCIS 2005: 20th International Symposium, Istanbul, Turkey, October 26-28, 2005. Proceedings 20*, 284–293 (Springer, 2005).
22. Matthias, P. & Rolink, A. G. Transcriptional networks in developing and mature b cells. *Nat. Rev. Immunol.* **5**, 497–508 (2005).
23. Wang, Y., Liu, J., Burrows, P. D. & Wang, J.-Y. in *B Cells in Immunity and Tolerance* (ed. Wang, J.-Y.) 1–22 (Springer, 2020).
24. Stoeckius, M. et al. Simultaneous epitope and transcriptome measurement in single cells. *Nat. methods* **14**, 865–868 (2017).
25. Chen, S., Lake, B. B. & Zhang, K. High-throughput sequencing of the transcriptome and chromatin accessibility in the same cell. *Nat. Biotechnol.* **37**, 1452–1457 (2019).
26. Mimitou, E. P. et al. Scalable, multimodal profiling of chromatin accessibility, gene expression and protein levels in single cells. *Nat. Biotechnol.* **39**, 1246–1258 (2021).
27. Hao, Y. et al. Integrated analysis of multimodal single-cell data. *Cell* **184**, 3573–3587 (2021).
28. Wolf, F. A. et al. Paga: graph abstraction reconciles clustering with trajectory inference through a topology preserving map of single cells. *Genome Biol.* **20**, 1–9 (2019).
29. Kurosaki, T., Kometani, K. & Ise, W. Memory b cells. *Nat. Rev. Immunol.* **15**, 149–159 (2015).
30. Olweus, J. et al. Dendritic cell ontogeny: a human dendritic cell lineage of myeloid origin. *Proc. Natl Acad. Sci.* **94**, 12551–12556 (1997).
31. Qian, J. et al. A pan-cancer blueprint of the heterogeneous tumor microenvironment revealed by single-cell profiling. *Cell Res.* **30**, 745–762 (2020).
32. Shin, J.-Y., Wang, C.-Y., Lin, C.-C. & Chu, C.-L. A recently described type 2 conventional dendritic cell (cdc2) subset mediates inflammation. *Cell. Mol. Immunol.* **17**, 1215–1217 (2020).
33. Cromer, M. K. et al. Gene replacement of α-globin with β-globin restores hemoglobin balance in β-thalassemia-derived hematopoietic stem and progenitor cells. *Nat. Med.* **27**, 677–687 (2021).
34. Zheng, G. X. Y. et al. Massively parallel digital transcriptional profiling of single cells. *Nat. Commun.* **8**, 14049 (2023).
35. Li, S. et al. Epicarousel: memory-and time-efficient identification of metacells for atlas-level single-cell chromatin accessibility data. *Bioinformatics* **40**, btae191 (2024).
36. Stuart, T., Srivastava, A., Madad, S., Lareau, C. A. & Satija, R. Single-cell chromatin state analysis with signac. *Nat. methods* **18**, 1333–1341 (2021).
37. Baron, M. et al. A single-cell transcriptomic map of the human and mouse pancreas reveals inter-and intra-cell population structure. *Cell Syst.* **3**, 346–360 (2016).
38. Muraro, M. J. et al. A single-cell transcriptome atlas of the human pancreas. *Cell Syst.* **3**, 385–394 (2016).
39. Segerstolpe, Å et al. Single-cell transcriptome profiling of human pancreatic islets in health and type 2 diabetes. *Cell Metab.* **24**, 593–607 (2016).
40. Wang, Y. J. et al. Single-cell transcriptomics of the human endocrine pancreas. *Diabetes* **65**, 3028–3038 (2016).

41. Xin, Y. et al. Rna sequencing of single human islet cells reveals type 2 diabetes genes. *Cell Metab.* **24**, 608–615 (2016).
42. Korsunsky, I. et al. Fast, sensitive and accurate integration of single-cell data with harmony. *Nat. methods* **16**, 1289–1296 (2019).
43. Blondel, V. D., Guillaume, J.-L., Lambiotte, R. & Lefebvre, E. Fast unfolding of communities in large networks. *J. Stat. Mech. Theory Exp.* **2008**, P10008 (2008).
44. Olaniru, O. E. et al. Single-cell transcriptomic and spatial landscapes of the developing human pancreas. *Cell Metab.* **35**, 184–199 (2023).
45. Anderson, K. R. et al. The l6 domain tetraspanin tm4sf4 regulates endocrine pancreas differentiation and directed cell migration. *Development* **138**, 3213–3224 (2011).
46. Barnett, K. C., Li, S., Liang, K. & Ting, J. P.-Y. A 360 view of the inflammasome: Mechanisms of activation, cell death, and diseases. *Cell* **186**, 2288–2312 (2023).
47. Liao, M. et al. Hepatic tnfrsf12a promotes bile acid-induced hepatocyte pyroptosis through nfκb/caspase-1/gsdmd signaling in cholestasis. *Cell Death Discov.* **9**, 26 (2023).
48. Li, X. et al. Combined plasma olink proteomics and transcriptomics identifies cxcl1 and tnfrsf12a as potential predictive and diagnostic inflammatory markers for acute kidney injury. *Inflammation* **47**, 1547–1563 (2024).
49. Singh, V. K., Yadav, D. & Garg, P. K. Diagnosis and management of chronic pancreatitis: a review. *Jama* **322**, 2422–2434 (2019).
50. Artur, S. et al. A benchmark for prediction of transcriptomic responses to chemical perturbations across cell types. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track* (openreview.net, 2024).
51. Kline, M. et al. Abt-737, an inhibitor of bcl-2 family proteins, is a potent inducer of apoptosis in multiple myeloma cells. *Leukemia* **21**, 1549–1560 (2007).
52. MacQueen, J. et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, vol. 1, 281–297 (Oakland, CA, USA, 1967).
53. Hie, B., Cho, H., DeMeo, B., Bryson, B. & Berger, B. Geometric sketching compactly summarizes the single-cell transcriptomic landscape. *Cell Syst.* **8**, 483–493 (2019).
54. Lopez, R., Regier, J., Cole, M. B., Jordan, M. I. & Yosef, N. Deep generative modeling for single-cell transcriptomics. *Nat. Methods* **15**, 1053–1058 (2018).
55. Gayoso, A. et al. Joint probabilistic modeling of single-cell multi-omic data with totalvi. *Nat. Methods* **18**, 272–282 (2021).
56. Lin, X., Tian, T., Wei, Z. & Hakonarson, H. Clustering of single-cell multi-omics data with a multimodal deep learning method. *Nat. Commun.* **13**, 7705 (2022).
57. Li, G. et al. A deep generative model for multi-view profiling of single-cell rna-seq and atac-seq data. *Genome Biol.* **23**, 20 (2022).
58. Martens, L. D., Fischer, D. S., Yépez, V. A., Theis, F. J. & Gagneur, J. Modeling fragment counts improves single-cell atac-seq analysis. *Nat. Methods* **21**, 28–31 (2024).
59. Paszke, A. et al. Automatic differentiation in pytorch. In *NIPS 2017 Workshop on Autodiff* (NIPS, 2017).
60. Paszke, A. et al. Pytorch: An imperative style, high-performance deep learning library. *Adv. neural Inf. Process. Syst.* **32**, 8026–8037 (2019).
61. Kingma, D. P. & Ba, J. Adam: a method for stochastic optimization. In *International Conference on Learning Representations* (ICLR, 2015).
62. Ruder, S. An overview of gradient descent optimization algorithms. Preprint at https://arxiv.org/abs/1609.04747 (2016).
63. Luo, H. et al. Characterizing dedifferentiation of thyroid cancer by integrated analysis. *Sci. Adv.* **7**, eabf3657 (2021).
64. Brodersen, K. H., Ong, C. S., Stephan, K. E. & Buhmann, J. M. The balanced accuracy and its posterior distribution. In *2010 20th international conference on pattern recognition*, 3121–3124 (IEEE, 2010).
65. Kelleher, J. D., Mac Namee, B. & D'arcy, A.*Fundamentals of machine learning for predictive data analytics: algorithms, worked examples, and case studies* (MIT press, 2020).
66. Bredikhin, D., Kats, I. & Stegle, O. Muon: multimodal omics analysis framework. *Genome Biol.* **23**, 42 (2022).
67. Vinh, N. X., Epps, J. & Bailey, J. Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. *J. Mach. Learn. Res.* **11**, 2837–2854 (2010).
68. Rand, W. M. Objective criteria for the evaluation of clustering methods. *J. Am. Stat. Assoc.* **66**, 846–850 (1971).
69. Rosenberg, A. & Hirschberg, J. V-measure: A conditional entropy-based external cluster evaluation measure. In *Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL)*, 410–420 (Association for Computational Linguistics, 2007).
70. Soneson, C. & Robinson, M. D. Bias, robustness and scalability in single-cell differential expression analysis. *Nat. Methods* **15**, 255–261 (2018).
71. Waskom, M. L. seaborn: statistical data visualization. *J. Open Source Softw.* **6**, 3021 (2021).
72. Virtanen, P. et al. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nat. Methods* **17**, 261–272 (2020).
73. Li, Y. et al. Metaq: fast, scalable and accurate metacell inference via single-cell quantization, https://doi.org/10.5281/zenodo.14271480 (2024).

Acknowledgements

This work was supported in part by the following grants: National Natural Science Foundation of China under Grant 62176171 (P), U21B2040 (P), 623B2075 (L), 82103031 (L), and 82272933 (L); Fundamental Research Funds for the Central Universities under Grant CJ202303 (P); Sichuan Science and Technology Planning Project under Grant 24NSFTD0130 (P); Sichuan Science and Technology Program 2023YFS0098 (L) and 2023YFG0278 (L); Clinical Research Incubation Project, West China Hospital, Sichuan University under Grant 22HXFH019 (L).

Author contributions

X.P. and Yunfan L. conceived the study and designed the MetaQ algorithm. Yunfan L. implemented the MetaQ algorithm. Yunfan L., Yijie L., D.P., and P.H. evaluated the baseline methods. Hancong L., D.Z., L.C., and Han L. preprocessed the data and analyzed the results. X.L. and J.X. participated in the paper revision. All authors participated in writing the manuscript.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s41467-025-56424-6.

Correspondence and requests for materials should be addressed to Xi Peng.

Peer review information *Nature Communications* thanks Shengquan Chen, and the other, anonymous, reviewer(s) for their contribution to the peer review of this work. A peer review file is available.

Reprints and permissions information is available at http://www.nature.com/reprints

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.