Cell BLAST: Searching large-scale scRNA-seq database via unbiased cell embedding

Zhi-Jie Cao, Lin Wei, Shen Lu, De-Chang Yang, Ge Gao*

[1] Biomedical Pioneering Innovation Center (BIOPIC), Beijing Advanced Innovation Center for Genomics (ICG), Center for Bioinformatics (CBI), State Key Laboratory of Protein and Plant Gene Research, School of Life Sciences, Peking University, Beijing, 100871, China
[2] College of Life Sciences, Beijing Normal University, Beijing, 100875, China

* To whom correspondence should be addressed. Tel: +86-010-62755206; Email: gaog@mail.cbi.pku.edu.cn

Abstract

Large amount of single-cell RNA sequencing data produced by various technologies is accumulating rapidly. An efficient cell querying method facilitates integrating existing data and annotating new data. Here we present a novel cell querying method Cell BLAST based on deep generative modeling, together with a well-curated reference database and a user-friendly Web interface at http://cblast.gao-lab.org, as an accurate and robust solution to large-scale cell querying.

Main Text

Technological advances during the past decade have led to rapid accumulation of large-scale single-cell RNA sequencing (scRNA-seq) data. Analogous to biological sequence analysis[1], identifying expression similarity to well-curated references via a cell querying algorithm is becoming the first step of annotating newly sequenced cells. Tools have been developed to identify similar cells using approximate cosine distance[2] or LSH Hamming distance[3, 4] calculated from a subset of carefully selected genes. Such intuitive approach is efficient, especially for large-scale data, but may suffer from non-biological variation across datasets, i.e. batch effect[5, 6]. Meanwhile, multiple data harmonization methods have been proposed to remove such confounding factors during alignment, for example, via warping canonical correlation vectors[7] or matching mutual nearest neighbors across batches[6].While these methods can be applied to align multiple reference datasets, computation-intensive realignment is required for mapping query cells to the (pre-)aligned reference data space.

Here we introduce a new customized deep generative model together with a cell-to-cell similarity metric specifically designed for cell querying to address these challenges (Figure 1A, Method). Differing from canonical variational autoencoder (VAE) models[8-11], adversarial batch alignment is applied to correct batch effect during low-dimensional embedding of reference datasets. Query cells can be readily mapped to the batch-corrected reference space due to parametric nature of neural networks. Such design also enables a special “online tuning” mode which is able to handle batch effect between query and reference when necessary. Moreover, by exploiting the model’s universal approximator posterior to model uncertainty in latent space, we implement a distribution-based metric for measuring cell-to-cell similarity. Last but not least, we also provide a well-curated multi-species single-cell transcriptomics database (ACA) and an easy-to-use Web interface for convenient exploratory analysis.

To assess our model’s capability to capture biological similarity in the low-dimensional latent space, we first benchmarked against several popular dimension reduction tools[8, 12, 13] using real-world data (Supplementary Table 1), and found that our model is overall among the best performing methods (Supplementary Figure 1-2). We further compared batch effect correction performance using combinations of multiple datasets with overlapping cell types

48 profiled (Supplementary Table 1). Our model achieves significantly better dataset mixing
49 (Figure 1B) while maintaining comparable cell type resolution (Figure 1C). Latent space
50 visualization also demonstrates that our model is able to effectively remove batch effect for
51 multiple datasets with considerable difference in cell type distribution (Supplementary Figure
52 3). Of note, we found that the correction of inter-dataset batch effect does not automatically
53 generalize to that within each dataset, which is most evident in the pancreatic datasets
54 (Supplementary Figure 3C-D, Supplementary Figure 4A-C). For such complex scenarios, our
55 model is flexible in removing multiple levels of batch effect simultaneously (Supplementary
56 Figure 4D-H).
57
58 While the unbiased latent space embedding derived by nonlinear deep neural network
59 effectively removes confounding factors, the network's random components and nonconvex
60 optimization procedure also lead to serious challenges, especially false positive hits when
61 cells outside reference types are provided as query. Thus, we propose a novel probabilistic
62 cell-to-cell similarity metric in latent space based on posterior distribution of each cell, which
63 we term "normalized projection distance" (NPD). Distance metric ROC analysis (Method)
64 shows that our posterior NPD metric is more accurate and robust than Euclidean distance,
65 which is commonly used in other neural network-based embedding tools (Figure 1D).
66 Additionally, we exploit stability of query-hit distance across multiple models to improve
67 specificity (Methods, Supplementary Figure 4L). An empirical p-value is computed for each
68 query hit as a measure of "confidence", by comparing posterior distance to the empirical
69 NULL distribution obtained from randomly selected pairs of cells in the queried database.
70
71 The high specificity of Cell BLAST is especially important for discovering novel cell types.
72 Two recent studies ("Montoro"[14] and "Plasschaert"[15]) independently reported a rare tracheal
73 cell type named pulmonary ionocyte. We artificially removed ionocytes from the "Montoro"
74 dataset, and used it as reference to annotate query cells from the "Plasschaert" dataset. In
75 addition to accurately annotating 95.9% of query cells, Cell BLAST correctly rejects 12 out
76 of 19 "Plasschaert" ionocytes (Figure 1E). Moreover, it highlights the existence of a putative
77 novel cell type as a well-defined cluster with large p-values among all 156 rejected cells,
78 which corresponds to ionocytes (Figure 1F-G, Supplementary Figure 6A, also see
79 Supplementary Figure 5 for more detailed analysis on the remaining 7 cells). In contrary,
80 scmap-cell[2] only rejected 7 "Plasschaert" ionocytes despite higher overall rejection number
81 of 401 (i.e. more false negatives, Supplementary Figure 6B-E).

82

83 We further systematically compared the performance of query-based cell typing with scmap-
84 cell[2] and CellFishing.jl[4] using four groups of datasets, each including both positive control
85 and negative control queries (first 4 groups in Supplementary Table 2). Of interest, while Cell
86 BLAST shows superior performance than scmap-cell and CellFishing.jl under the default
87 setting (Supplementary Figure 7A-C, 8-10), detailed ROC analysis reveals that performance
88 of scmap-cell could be further improved to a level comparative to Cell BLAST by employing
89 higher thresholds, while ROC and optimal thresholds of CellFishing.jl show large variation
90 across different datasets (Supplementary Figure 7D). Cell BLAST presents the most robust
91 performance with default threshold (p-value < 0.05) across multiple datasets, which largely
92 benefits real-world application. Additionally, we also assessed their scalability using
93 reference data varying from 1,000 to 1,000,000 cells. Both Cell BLAST and CellFishing.jl
94 scale well with increasing reference size, while scmap-cell's querying time rises dramatically
95 for larger reference dataset with more than 10,000 cells (Supplementary Figure 7E).
96

97 Moreover, our deep generative model combined with posterior-based latent-space similarity
98 metric enables Cell BLAST to model continuous spectrum of cell states accurately. We
99 demonstrate this using a study profiling mouse hematopoietic progenitor cells ("Tusi"[16]), in
100 which computationally inferred cell fate distributions are available. For the purpose of
101 evaluation, cell fate distributions inferred by authors are recognized as ground truth. We
102 selected cells from one sequencing run as query and the other as reference to test if we can
103 accurately transfer continuous cell fate between experimental batches (Figure 2A-B). Jensen-
104 Shannon divergence between prediction and ground truth shows that our prediction is again
105 more accurate than scmap (Figure 2C).
106

107 Besides batch effect among multiple reference datasets, *bona fide* biological similarity could
108 also be confounded by large, undesirable bias between query and reference data. Taking
109 advantage of the dedicated adversarial batch alignment component, we implemented a
110 particular "online tuning" mode to handle such often-neglected confounding factor. Briefly,
111 the combination of reference and query data is used to fine-tune the existing reference-based
112 model, with query-reference batch effect added as an additional component to be removed by
113 adversarial batch alignment (Method). Using this strategy, we successfully transferred cell
114 fate from the above "Tusi" dataset to an independent human hematopoietic progenitor dataset
115 ("Velten"[17]) (Figure 2D). Expression of known cell lineage markers validates the rationality

116 of transferred cell fates (Supplementary Figure 11A-F). In contrary, scmap-cell incorrectly
117 assigned most cells to monocyte and granulocyte lineages (Supplementary Figure 11G). As
118 another example, we applied "online tuning" to *Tabula Muris*[18] spleen data, which exhibits
119 significant batch effect between 10x and Smart-seq2 processed cells. ROC of Cell BLAST
120 improves significantly after "online tuning", achieving high specificity, sensitivity and
121 Cohen's κ at the default cutoff (Supplementary Figure 11H, last group in Supplementary
122 Table 2).
123
124 A comprehensive and well-curated reference database is crucial for practical application of
125 Cell BLAST. Based on public scRNA-seq datasets, we developed a comprehensive Animal
126 Cell Atlas (ACA). With 986,305 cells in total, ACA currently covers 27 distinct organs
127 across 8 species, offering the most comprehensive compendium for diverse species and
128 organs (Figure 2E, Supplementary Figure 12A-B, Supplementary Table 3). To ensure unified
129 and high-resolution cell type description, all records in ACA are collected and annotated with
130 a standard procedure (Method), with 95.4% of datasets manually curated with Cell Ontology,
131 a structured controlled vocabulary for cell types.
132
133 A user-friendly Web server is publicly accessible at http://cblast.gao-lab.org, with all curated
134 datasets and pretrained models available, providing "off-the-shelf" querying service. Of note,
135 we found that our model works well on all ACA datasets with minimal hyperparameter
136 tuning (latent space visualization available on the website). Based on this wealth of resources,
137 users can obtain querying hits and visualize cell type predictions with minimal effort
138 (Supplementary Figure 12C-D). For advanced users, a well-documented python package
139 implementing the Cell BLAST toolkit is also available, which enables model training on
140 custom references and diverse downstream analyses.
141
142 By explicitly modeling multi-level batch effect as well as uncertainty in cell-to-cell similarity
143 estimation, Cell BLAST is an accurate and robust querying algorithm for heterogeneous
144 single-cell transcriptome datasets. In combination with a comprehensive, well-annotated
145 database and an easy-to-use Web interface, Cell BLAST provides a one-stop solution for
146 both bench biologists and bioinformaticians

Software availability

The full package of Cell BLAST is available at http://cblast.gao-lab.org

Acknowledgements

The authors thank Drs. Zemin Zhang, Cheng Li, Letian Tao, Jian Lu and Liping Wei at Peking University for their helpful comments and suggestions during the study.
This work was supported by funds from the National Key Research and Development Program (2016YFC0901603), the China 863 Program (2015AA020108), as well as the State Key Laboratory of Protein and Plant Gene Research and the Beijing Advanced Innovation Center for Genomics (ICG) at Peking University. The research of G.G. was supported in part by the National Program for Support of Top-notch Young Professionals.
Part of the analysis was performed on the Computing Platform of the Center for Life Sciences of Peking University, and supported by the High-performance Computing Platform of Peking University.

Author contributions

G.G. conceived the study and supervised the research; Z.J.C. and L.W. contributed to the computational framework and data curation; S. L., Z.J.C., and D.C.Y designed, implemented and deployed the website; Z.J.C. and G.G. wrote the manuscript with comments and inputs from all co-authors.

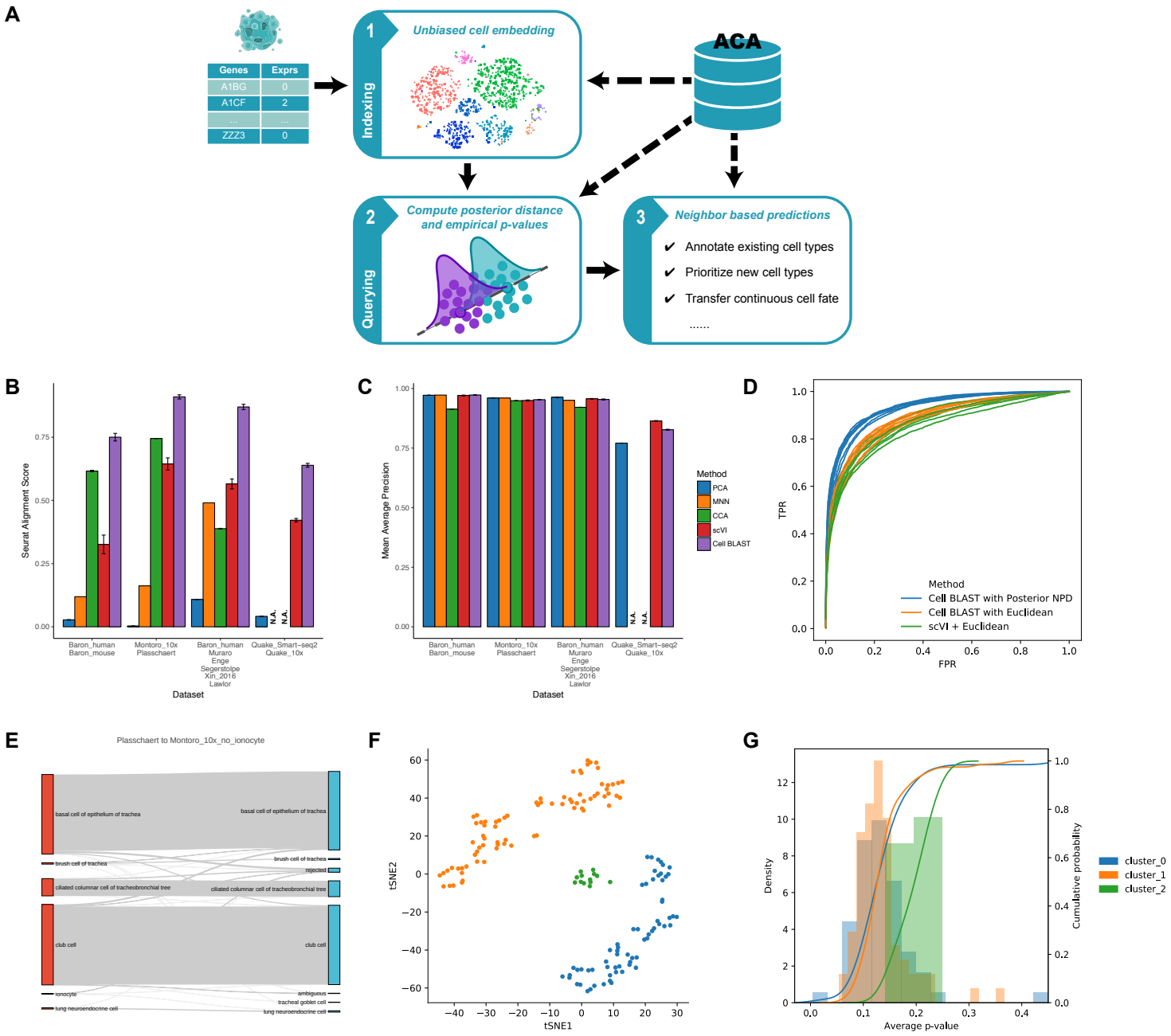
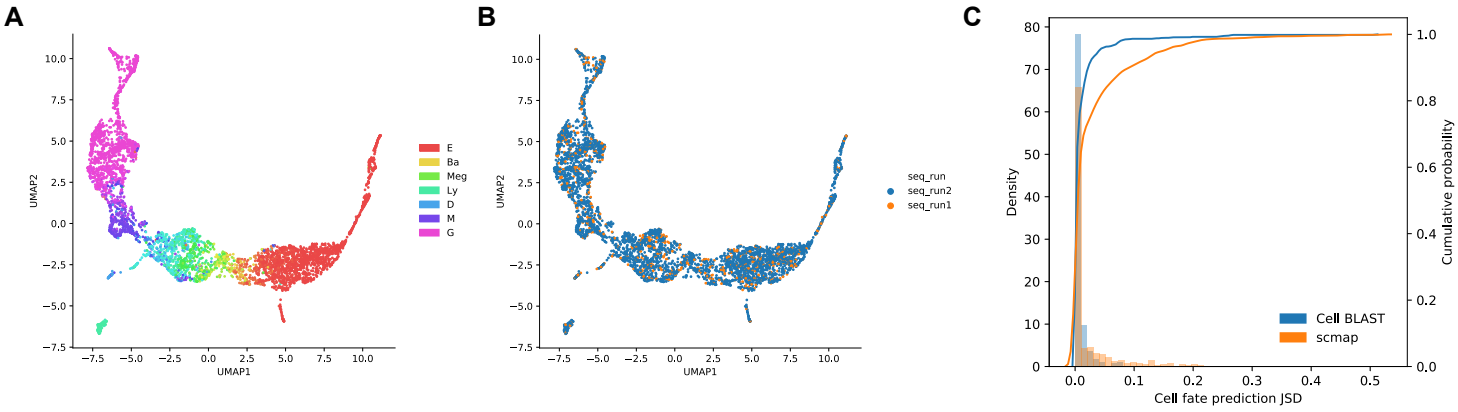
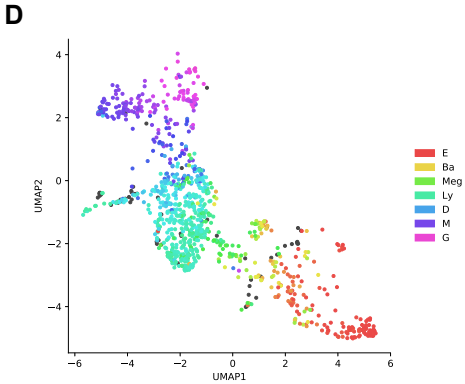


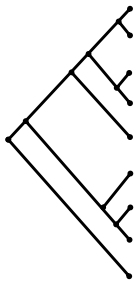
Figure. 1 Cell BLAST benchmarking and application to trachea datasets.

(A) Overall Cell BLAST workflow. (B) Extent of dataset mixing after batch effect correction in four groups of datasets, quantified by Seurat alignment score. High Seurat alignment score indicates that local neighborhoods consist of cells from different datasets uniformly rather than the same dataset only. Methods that did not finish under 2-hour time limit are marked as N.A. (C) Cell type resolution after batch effect correction, quantified by cell type mean average precision (MAP). MAP can be thought of as a generalization to nearest neighbor accuracy, with larger values indicating higher cell type resolution, thus more suitable for cell querying. Methods that did not finish under 2-hour time limit are marked as N.A. (D) ROC curve of different distance metrics in discriminating cell pairs with the same cell type from cell pairs with different cell types. (E) Sankey plot comparing Cell BLAST predictions and original cell type annotations for the “Plasschaert” dataset. (F) t-SNE visualization of Cell BLAST rejected cells, colored by unsupervised clustering.





E



	ACA	Single Cell Portal	Hemberg	SCPortalen	scRNASeqDB
Human	**9**	6	4	5	5
Mouse	**22**	10	6	6	-
Lizard	**1**	-	-	-	-
Turtle	**1**	-	-	-	-
Zebrafish	**4**	1	-	-	-
Planaria	**1**	-	-	-	-
Drosophila	**1**	**1**	-	-	-
Nematode	**1**	-	-	-	-
Hydra	-	**1**	-	-	-

Figure. 2 Application to hematopoietic progenitor datasets.

(A, B) UMAP visualization of latent space learned on the "Tusi" dataset, colored by sequencing run (A) and cell fate (B). Model is trained solely on cells from run 2 and used to project cells from run 1. Each one of the seven terminal cell fates (E, erythroid; Ba, basophilic or mast; Meg, megakaryocytic; Ly, lymphocytic; D, dendritic; M, monocytic; G, granulocytic neutrophil) are assigned a distinct color. Color of each single cell is then determined by the linear combination of these seven colors in hue space, weighed by cell fate distribution among these terminal fates. (C) Distribution of Jensen-Shannon divergence between predicted cell fate distributions and author provided "ground truth". (D) UMAP visualization of the "Velten" dataset, colored by Cell BLAST predicted cell fates. (E) Number of organs covered in each species for different single-cell transcriptomics databases, including Single Cell Portal (https://portals.broadinstitute.org/single_cell); Hemberg collection[2]; SCPortalen[19]; scRNASeqDB[20].

Methods

The deep generative model

The model we used is based on adversarial autoencoder (AAE)[21]. Denote the gene expression profile of a cell as $\boldsymbol{x} \in \mathbb{R}^G$. The data generative process is modeled by a continuous latent variable $\boldsymbol{z} \in \mathbb{R}^D$ $(D \ll G)$ with Gaussian prior $\boldsymbol{z} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_D)$, as well as a one-hot latent variable $\boldsymbol{c} \in \{0,1\}^K$, $\boldsymbol{c}^T\boldsymbol{c} = 1$ with categorical prior $\boldsymbol{c} \sim Cat(K)$, which aims at modeling cell type clusters. A unified latent vector is then determined by $\boldsymbol{l} = \boldsymbol{z} + H\boldsymbol{c}$, where $H \in \mathbb{R}^{D\times K}$. A neural network (decoder, denoted by Dec below) maps the cell embedding vector $\boldsymbol{l}$ to two parameters of the negative binomial distribution $\boldsymbol{\mu}, \boldsymbol{\theta} = \mathrm{Dec}(\boldsymbol{l})$ that models the distribution of expression profile $\boldsymbol{x}$:

$$p(\boldsymbol{x}|\boldsymbol{z}, \boldsymbol{c}; \mathrm{Dec}) = p(\boldsymbol{x}|\boldsymbol{\mu}, \boldsymbol{\theta}) = \prod_{j=1}^{G} p(x_j|\mu_j, \theta_j) \tag{1}$$

$$p(x_j|\mu_j, \theta_j) = \frac{\Gamma(x_j + \theta_j)}{\Gamma(\theta_j)\Gamma(x_j + 1)} \left(\frac{\mu_j}{\theta_j + \mu_j}\right)^{x_j} \left(\frac{\theta_j}{\theta_j + \mu_j}\right)^{\theta_j} \tag{2}$$

Where $\boldsymbol{\mu}$ and $\boldsymbol{\theta}$ are mean and dispersion of the negative binomial distribution. Theoretically, the negative binomial model should be fitted on raw count data[8, 13, 22]. However, for the purpose of cell querying, datasets have to be normalized to minimize the influence of capture efficiency and sequencing depth. We empirically found that on normalized data, negative binomial still produced better results than alternative distributions like log-normal. To prevent numerical instability caused by normalization breaking the mean-variance relation of negative binomial, we additionally include variance of the dispersion parameter as a regularization term.

Training objectives for the adversarial autoencoder are:

$$\begin{aligned}\min_{\mathrm{Dec,Enc}} -\mathbb{E}_{\boldsymbol{x}\sim p_{data}(\boldsymbol{x})}\big[\mathbb{E}_{\boldsymbol{z}\sim q(\boldsymbol{z}|\boldsymbol{x};\mathrm{Enc}),\boldsymbol{c}\sim q(\boldsymbol{c}|\boldsymbol{x};\mathrm{Enc})} \log p(\boldsymbol{x}|\boldsymbol{z}, \boldsymbol{c}; \mathrm{Dec}) + \lambda_z \\ \cdot \mathbb{E}_{\boldsymbol{z}\sim q(\boldsymbol{z}|\boldsymbol{x};\mathrm{Enc})} \log \mathrm{D_z}(\boldsymbol{z}) + \lambda_c \cdot \mathbb{E}_{\boldsymbol{c}\sim q(\boldsymbol{c}|\boldsymbol{x};\mathrm{Enc})} \log \mathrm{D_c}(\boldsymbol{c})\big]\end{aligned} \tag{3}$$

$$\max_{\mathrm{D_z}} \lambda_z \cdot \left(\mathbb{E}_{\boldsymbol{z}\sim p(\boldsymbol{z})}[\log \mathrm{D_z}(\boldsymbol{z})] + \mathbb{E}_{\boldsymbol{x}\sim p_{data}(\boldsymbol{x})}\mathbb{E}_{\boldsymbol{z}\sim q(\boldsymbol{z}|\boldsymbol{x};\mathrm{Enc})}\big[\log\big(1 - \mathrm{D_z}(\boldsymbol{z})\big)\big]\right) \tag{4}$$

$$\max_{\mathrm{D_c}} \lambda_c \cdot \left(\mathbb{E}_{\boldsymbol{c}\sim p(\boldsymbol{c})}[\log \mathrm{D_c}(\boldsymbol{c})] + \mathbb{E}_{\boldsymbol{x}\sim p_{data}(\boldsymbol{x})}\mathbb{E}_{\boldsymbol{c}\sim q(\boldsymbol{c}|\boldsymbol{x};\mathrm{Enc})}\big[\log\big(1 - \mathrm{D_c}(\boldsymbol{c})\big)\big]\right) \tag{5}$$

$q(\boldsymbol{z}|\boldsymbol{x}; \mathrm{Enc})$ and $q(\boldsymbol{c}|\boldsymbol{x}; \mathrm{Enc})$ are "universal approximator posteriors" parameterized by another neural network (encoder, denoted by Enc). Expectations with regard to $q(\boldsymbol{z}|\boldsymbol{x}; \mathrm{Enc})$

and $q(\boldsymbol{c}|\boldsymbol{x};\mathrm{Enc})$ are approximated by sampling $\boldsymbol{x}' \sim Poisson(\boldsymbol{x})$ and feeding to the deterministic encoder network. D_z and D_c are discriminator networks for $\boldsymbol{z}$ and $\boldsymbol{c}$ which output the probability that a latent variable sample is from the prior rather than the posterior. Effectively, adversarial training between encoder (Enc) and discriminators (D_z and D_c) drives encoder output to match prior distributions of latent variables $p(\boldsymbol{z})$ and $p(\boldsymbol{c})$. λ_z and λ_c are hyperparameters that control prior matching strength. The model is much easier and more stable to train compared with canonical GANs because of low dimensionality and simple distribution of $\boldsymbol{z}$ and $\boldsymbol{c}$.

At convergence, the encoder learns to map data distribution to latent variables that follow their respective prior distributions and the decoder learns to map latent variables from prior distributions back to data distribution. The key element we use for cell querying is vector $\boldsymbol{l}$ on the decoding path, as it defines a unified latent space in which biological similarities are well captured. The model also works if no categorical latent variable is used, in which case $\boldsymbol{l} = \boldsymbol{z}$ directly.

Some architectural designs are learned from scVI[8], including logarithm transformation before encoder input, and softmax output scaled by library size when computing $\boldsymbol{\mu}$. Stochastic gradient descent with minibatches is applied to optimize the loss functions. Specifically, we use the "RMSProp" optimization algorithm with no momentum term to ensure stability of adversarial training. The model is implemented using Tensorflow[23] python library.

Adversarial batch alignment

As a natural extension to the prior matching adversarial training strategy described in the previous section, also following recent works in domain adaptation[24-26], we propose the adversarial batch alignment strategy to align the latent space distribution of different batches. Below, we extend derivation in the original GAN paper[27] to show that adversarial batch alignment converges when embedding space distributions of different batches are aligned. In the case of multiple batches, we assume that marginal data distribution can be factorized as below:

$$p_{data}(\boldsymbol{x}, \boldsymbol{b}) = p(\boldsymbol{b})p(\boldsymbol{x}|\boldsymbol{b}) \tag{6}$$

$\boldsymbol{b} \in \{0,1\}^B, \boldsymbol{b}^T\boldsymbol{b} = 1$ denotes a one-hot batch indicator, and batch distribution $p(\boldsymbol{b})$ is categorical:

$$p(b_i = 1) = w_i, \qquad \sum_{i=0}^{B} w_i = 1 \tag{7}$$

Adversarial batch alignment introduces an additional loss:

$$\min_{\mathrm{Dec,Enc}} \mathbb{E}_{\boldsymbol{b}\sim p(\boldsymbol{b}),\boldsymbol{x}\sim p(\boldsymbol{x}|\boldsymbol{b})} \mathbb{E}_{\boldsymbol{l}\sim\delta(\boldsymbol{z}+\boldsymbol{c}H-\boldsymbol{l})q(\boldsymbol{z}|\boldsymbol{x})q(\boldsymbol{c}|\boldsymbol{x})}[\mathcal{L}_{base} + \lambda_b \cdot \boldsymbol{b}^T \log \mathrm{D_b}(\boldsymbol{l})] \tag{8}$$

$$\max_{\mathrm{D_b}} \mathbb{E}_{\boldsymbol{b}\sim p(\boldsymbol{b}),\boldsymbol{x}\sim p(\boldsymbol{x}|\boldsymbol{b})} \mathbb{E}_{\boldsymbol{l}\sim\delta(\boldsymbol{z}+\boldsymbol{c}H-\boldsymbol{l})q(\boldsymbol{z}|\boldsymbol{x})q(\boldsymbol{c}|\boldsymbol{x})}[\lambda_b \cdot \boldsymbol{b}^T \log \mathrm{D_b}(\boldsymbol{l})] \tag{9}$$

$\mathcal{L}_{base}$ denotes the loss function in (3). D_b is a multi-class batch discriminator network that outputs the probability distribution of batch membership based on embedding vector $\boldsymbol{l}$. λ_b is a hyperparameter controlling batch alignment strength. Additionally, the generative distribution is extended to condition on $\boldsymbol{b}$ as well:

$$\begin{gathered} p(\boldsymbol{x}|\boldsymbol{z},\boldsymbol{c},\boldsymbol{b};\mathrm{Dec}) = p(\boldsymbol{x}|\boldsymbol{\mu},\boldsymbol{\theta}) \\ \boldsymbol{\mu},\boldsymbol{\theta} = \mathrm{Dec}(\boldsymbol{l},\boldsymbol{b}) \\ \boldsymbol{l} = \boldsymbol{z} + \boldsymbol{c}H \end{gathered} \tag{10}$$

Now, we focus on batch alignment and discard the first $\mathcal{L}_{base}$ term and scaling parameter λ_b. To simplify notation, we fuse data distribution and encoder transformation, and replace the minimization over encoder to minimization over batch embedding distributions:

$$\min_{p(\boldsymbol{l})} \sum_{i=1}^{B} w_i \mathbb{E}_{\boldsymbol{l}\sim p_i(\boldsymbol{l})} \log \mathrm{D_b}_i(\boldsymbol{l}) \tag{11}$$

$$\max_{\mathrm{D_b}} \sum_{i=1}^{B} w_i \mathbb{E}_{\boldsymbol{l}\sim p_i(\boldsymbol{l})} \log \mathrm{D_b}_i(\boldsymbol{l}) \tag{12}$$

Here $\mathrm{D}_{b_i}(\boldsymbol{l})$ denotes the ith dimension of discriminator output, i.e. probability that the discriminator "thinks" a cell is from the ith batch. D_b is assumed to have sufficient capacity, which is generally reasonable in the case of neural networks. Global optimum of (12) is reached when D_b outputs optimal batch membership distribution at every $\boldsymbol{l}$:

$$\max_{\mathrm{D_b}_i(\boldsymbol{l})} w_i p_i(\boldsymbol{l}) \log \mathrm{D_b}_i(\boldsymbol{l}), \qquad s.t. \sum_{i=1}^{B} \mathrm{D_b}_i(\boldsymbol{l}) = 1 \tag{13}$$

Solution to the above maximization is given by:

$$\mathrm{D_b}^*{}_i(\boldsymbol{l}) = \frac{w_i p_i(\boldsymbol{l})}{\sum_{i=1}^{B} w_i p_i(\boldsymbol{l})} \tag{14}$$

Substituting $\mathrm{D}_b{}^*(l)$ back into (11), we obtain:

$$
\begin{aligned}
&\sum_{i=1}^{B} w_i \mathbb{E}_{\boldsymbol{l}\sim p_i(\boldsymbol{l})} \log \frac{w_i p_i(\boldsymbol{l})}{\sum_{i=1}^{B} w_i p_i(\boldsymbol{l})} \\
&\qquad = \sum_{i=1}^{B} w_i \mathbb{E}_{\boldsymbol{l}\sim p_i(\boldsymbol{l})} \log \frac{p_i(\boldsymbol{l})}{\sum_{i=1}^{B} w_i p_i(\boldsymbol{l})} + \sum_{i=1}^{B} w_i \mathbb{E}_{\boldsymbol{l}\sim p_i(\boldsymbol{l})} \log w_i \\
&\qquad = \sum_{i=1}^{B} w_i \cdot \mathrm{KL}\left(p_i(\boldsymbol{l}) \parallel \sum_{i=1}^{B} w_i p_i(\boldsymbol{l}) \right) + \sum_{i=1}^{B} w_i \log w_i \\
&\qquad \geq \sum_{i=1}^{B} w_i \log w_i
\end{aligned}
\tag{155}
$$

Thus $\sum_{i=1}^{B} w_i \log w_i$ is the global minimum, reached if and only if $p_i(\boldsymbol{l}) = p_j(\boldsymbol{l}), \forall i, j$. The minimization of (11) is equivalent to minimizing a form of generalized Jensen-Shannon divergence among multiple batch embedding distributions.

Note that in practice, model training looks for a balance between $\mathcal{L}_{base}$ and pure batch alignment. Aligning cells of the same type induces minimal cost in $\mathcal{L}_{base}$, while cells of different types could cause $\mathcal{L}_{base}$ to rise dramatically if improperly aligned. During training, gradient from both batch discriminators and decoder provide fine-grain guidance to align different batches, leading to better results than "hand-crafted" alignment strategies like CCA[7] and MNN[6]. Empirically, given proper values for λ_b, the adversarial approach correctly handles difference in cell type distribution among batches. If multiple levels of batch effect exist, e.g. within-dataset and cross-dataset, we use an independent batch discriminator for each component, providing extra flexibility.

Data preprocessing for benchmarks

Most informative genes were selected using the Seurat[7] function "FindVariableGenes". We set the argument "binning.method" to "equal_frequency" and left other arguments as default. If within dataset batch effect exists, genes are selected independently for each batch and then pooled together. By default, a gene is retained if it is selected in at least 50% of batches. Downstream benchmarks were all performed using this gene set, except for scmap and CellFishing.jl which provide their own gene selection method. GNU parallel[28] was used to parallelize and manage jobs throughout the benchmarking and data processing pipeline.

271 **Benchmarking dimension reduction**

272 PCA was performed using the R package irlba[29] (v2.3.2). ZIFA[12] was downloaded from its
273 Github repository, and hard coded random seeds were removed to reveal actual stability.
274 ZINB-WaVE[13] (v1.0.0) was performed using the R package zinbwave. scVI[8] (v0.2.3) was
275 downloaded from its Github repository, and minor changes were made to the original code to
276 address PyTorch[30] compatibility issues. Our modified versions of ZIFA and scVI are
277 available upon request.
278 For PCA and ZIFA, data was logarithm transformed after normalization and adding a
279 pseudocount of 1. Hyperparameters of all methods above were left as default. For our model,
280 we used the same set of hyperparameters throughout all benchmarks. λ_z and λ_c are both set to
281 0.001. All neural networks (encoder, decoder and discriminators) use a single layer of 128
282 hidden units. Learning rate of RMSProp optimizer is set to 0.001, and minibatches of size
283 128 are used. For comparability, target dimensionality of each method was set to 10. All
284 benchmarked methods were repeated multiple times with different random seeds. Run time
285 was limited to 2 hours, after which the job would be terminated.
286 Cell type nearest neighbor mean average precision (MAP) was computed with K nearest
287 neighbors of each cell based on low dimensional space Euclidean distance. Denote cell type
288 of a cell as y, and cell type of its ordered nearest neighbors as $y_1, y_2, \dots y_k$. Average precision
289 (AP) for that cell is defined as:

$$\mathrm{AP} = \frac{\sum_{k=1}^{K} 1_{y=y_k} \cdot \frac{\sum_{k'=1}^{k} 1_{y=y_{k'}}}{k}}{\sum_{k=1}^{K} 1_{y=y_k}} \tag{16}$$

290 Mean average precision is then given by:

$$\mathrm{MAP} = \frac{1}{N} \sum_{i=1}^{N} \mathrm{AP}_i \tag{17}$$

291 Note that when $K = 1$, MAP reduces to nearest neighbor accuracy. We set K to 1% of total
292 cell number throughout all benchmarks.

293 **Benchmarking batch effect correction**

294 We merge multiple datasets according to shared gene names. If datasets to be merged are
295 from different species, Ensembl ortholog[31] information is used to map them to ortholog
296 groups. To obtain informative genes in merged datasets, we take the union of informative

genes from each dataset, and then intersect the union with the intersection of detected genes from each dataset.

CCA[7] and MNN[6] alignments were performed using R packages Seurat[7] (v2.3.3) and scran[32] (v1.6.9) respectively. Hard coded random seeds in Seurat were removed to reveal actual stability. The modified version of Seurat is available upon request. For comparability, we evaluated cell type resolution and batch mixing in a 10-dimensional latent space. For MNN alignment, we set argument "cos.norm.out" to false, and left other arguments as default. PCA was applied to reduce dimensionality to 10 after obtaining the MNN corrected expression matrix. For CCA alignment, we used the first 10 canonical correlation vectors. Run time was limited to 2 hours, after which the job would be terminated. Seurat alignment score was computed exactly as described in the CCA alignment paper[7].

Cell querying based on posterior distributions

We evaluate cell-to-cell similarity based on posterior distribution distance. Like in the training phase, we obtain samples from the "universal approximator posterior" by sampling $\boldsymbol{x}' \sim Poisson(\boldsymbol{x})$ and feeding to the encoder network. In order to obtain robust estimation of distribution distance with a small number of posterior samples, we project posterior samples of two cells onto the line connecting their posterior point estimates in the latent space, and use projected scalar distribution distance to approximate true distribution distance. Wasserstein distance is computed on normalized projections to account for non-uniform density across the embedding space:

$$NPD(p,q) = \frac{1}{2} \cdot \left(W_1\left(z_p(p), z_p(q)\right) + W_1\left(z_q(p), z_q(q)\right)\right) \tag{18}$$

Where

$$W_1(u,v) = \inf_{\pi \in \Gamma(u,v)} \int |x-y| d\pi(x,y)$$

$$z_u(v) = \frac{v - \mathbb{E}(u)}{\sqrt{var(u)}}$$

We term this distance metric normalized projection distance (NPD). By default, 50 samples from the posterior are used to compute NPD, which produces sufficiently accurate results (Supplementary Figure 4I-J). The definition of NPD does not imply an efficient nearest neighbor searching algorithm. To increase speed, we first use Euclidean distance-based nearest neighbor searching, which is highly efficient in the low dimensional latent space, and then compute posterior distances only for these nearest neighbors. Empirical distribution of

326 posterior NPD for a dataset is obtained by computing posterior NPD on randomly selected
327 pairs of cells in the reference dataset. Empirical p-values of query hits are computed by
328 comparing posterior NPD of a query hit to this empirical distribution.
329 We note that even with the querying strategy described above, querying with single models
330 still occasionally leads to many false positive hits when cell types that the model has not been
331 trained on are provided as query. This is because embeddings of such untrained cell types are
332 mostly random, and they could localize close to reference cells by chance. We reason that
333 embedding randomness of untrained cell types could be utilized to identify and correctly
334 reject them. Practically, we train multiple models with different starting points (as determined
335 by random seeds), and compute query hit significance for each model. A query hit is
336 considered significant only if it is consistently significant across multiple models. To acquire
337 predictions based on significant hits, we use majority voting for discrete variables, e.g. cell
338 type, or averaging for continuous variables, e.g. cell fate distribution.

339 **Distance metric ROC analysis**

340 Our model and scVI[8] were fitted on reference datasets and applied to positive and negative
341 control query datasets in the pancreas group of Supplementary Table 2. We then randomly
342 selected 10,000 query-reference cell pairs. A query-reference pair is defined as "positive" if
343 the query cell and reference cell are of the same cell type, and "negative" otherwise. Each
344 benchmarked similarity metric was then computed on all sampled query-reference pairs, and
345 used as predictors for "positive" / "negative" pairs. AUROC values were computed for each
346 benchmarked similarity metric. In addition to Euclidean distance, we also computed posterior
347 distribution distances for scVI (Supplementary Figure 4K). NPD is computed as described in
348 (18), based on samples from the posterior Gaussian. JSD is computed in the original latent
349 space (without projection).

350 **Benchmarking cell querying**

351 For each of the four querying groups, three types of datasets were used, namely reference,
352 positive query and negative query (see Supplementary Table 2 for a detailed list of datasets
353 used in each querying group). For each querying method, cell type predictions for query cells
354 are obtained based on reference hits with a minimal similarity cutoff. Cell ontology
355 annotations in ACA were used as ground truth. Cells with no cell ontology annotations were
356 excluded in the analysis. Predictions are considered correct if and only if it exactly matches
357 the ground truth, i.e. no flexibility based on cell type similarity. This prevents unnecessary

358 bias introduced in the selection of cell type similarity measure. Cells were inversely weighed
359 by size of the corresponding dataset when computing sensitivity, specificity and Cohen's κ.
360 AUROC was computed using linear interpolation. For scmap[2], we varied minimal cosine
361 similarity requirement for nearest neighbors. For Cell BLAST, we varied maximal p-value
362 cutoff used in filtering hits. For CellFishing.jl[4], the original implementation does not include
363 a dedicated cell type prediction function, so we used the same strategy as for our own method
364 (majority voting after distance filtering) to acquire final predictions, in which we varied the
365 Hamming distance cutoff used in distance filtering. Lastly, 4 random seeds were tested for
366 each cutoff and each method to reflect stability. Several other cell querying tools
367 (CellAtlasSearch[3], scQuery[33], scMCA[34]) were not included in our benchmark because they
368 do not support custom reference datasets.

369 **Benchmarking querying speed**

370 To evaluate scalability of querying methods, we constructed reference data of varying sizes
371 by subsampling from the 1M mouse brain dataset[35]. For query data, the "Marques" dataset[36]
372 was used. For all methods, only querying time was recorded, not including time consumed to
373 build reference indices.

374 **Application to trachea datasets**

375 We first removed cells labeled as "ionocytes" in the "Montoro_10x"[14] dataset, and used
376 "FindVariableGenes" from Seurat to select informative genes using the remaining cells. Four
377 models with different starting points were trained on the tampered "Montoro_10x" dataset.
378 We computed posterior distribution distance p-values as mentioned before, and used a cutoff
379 of p-value > 0.1 to reject query cells from the "Plasschaert"[15] dataset as potential novel cell
380 types. We clustered rejected cells using spectral clustering (Scikit-learn[37] v0.20.1) after
381 applying t-SNE[38] to latent space coordinates. Average p-value for a query cell was computed
382 as the geometric mean of p-values across all hits.

383 **Online tuning**

384 When significant batch effect exists between reference and query, we support further aligning
385 query data with reference data in an online-learning manner. All components in the
386 pretrained model, including encoder, decoder, prior discriminators and batch discriminators
387 are retained. The reference-query batch effect is added as an extra component to be removed
388 using adversarial batch alignment. Specifically, a new discriminator dedicated to reference-

query batch effect is added, and the decoder is expanded to accept an extra one-hot indicator for reference and query. The expanded model is then fine-tuned using the combination of reference and query data. Two precautions are taken to prevent decrease in specificity caused by over-alignment. Firstly, adversarial alignment loss is constrained to cells that have mutual nearest neighbors[6] between reference and query data in each SGD minibatch. Secondly, we penalize the deviation of fine-tuned model weights from original weights.

Application to hematopoietic progenitor datasets

For within- "Tusi"[16] query, we trained four models using only cells from sequencing run 2, and cells from sequencing run 1 were used as query cells. PBA inferred cell fate distributions, which are 7-dimensional categorical distributions across 7 terminal cell fates, were used as ground truth. We took the average cell fate distributions of significant querying hits (p-value < 0.05) as predictions for query cells. As for scmap-cell, we filtered nearest neighbors according to default cosine similarity cutoff of 0.5. Jensen-Shannon divergence (JSD) between true and predicted cell fate distribution was computed as below:

$$JSD(\boldsymbol{p} \parallel \boldsymbol{q}) = \frac{1}{2} \cdot \sum_{i \in \{E,Ba,Meg,Ly,D,M,G\}} p_i \log \frac{p_i}{\frac{p_i + q_i}{2}} + q_i \log \frac{q_i}{\frac{p_i + q_i}{2}} \tag{19}$$

For cross-species querying between "Tusi" and "Velten"[17], we mapped both mouse and human genes to ortholog groups. Online tuning with 200 epochs was used to increase sensitivity and accuracy. Latent space visualization was performed with UMAP[39, 40].

ACA database construction

We searched Gene Expression Omnibus (GEO)[41] using the following search term:

```
(
   "expression profiling by high throughput sequencing"[DataSet Type] OR
   "expression profiling by high throughput sequencing"[Filter] OR
   "high throughput sequencing"[Platform Technology Type]
) AND
"gse"[Entry Type] AND
(
"single cell"[Title] OR
"single-cell"[Title]
) AND
("2013"[Publication Date] : "3000"[Publication Date]) AND
"supplementary"[Filter]
```

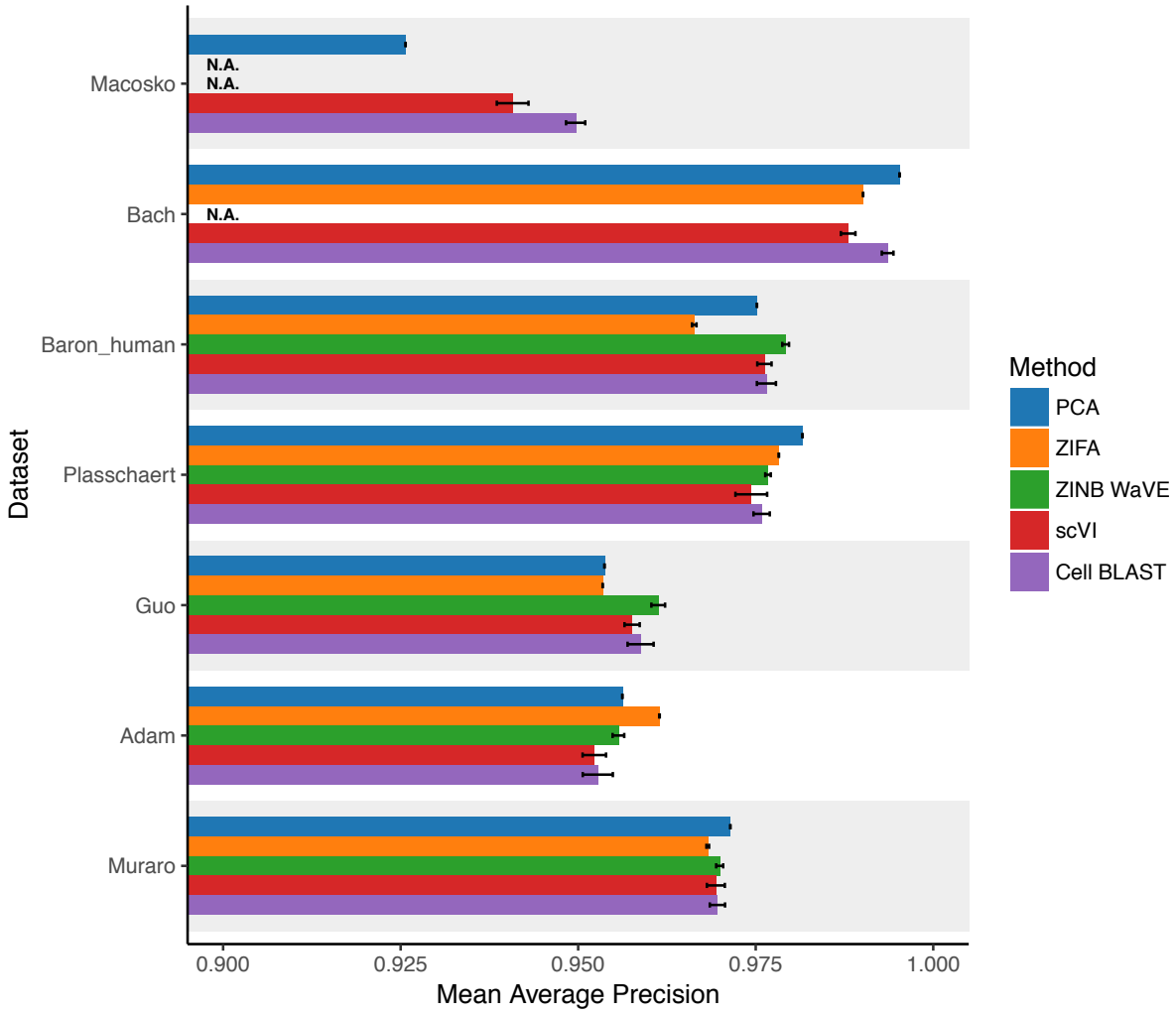
420 Datasets in the Hemberg collection (https://hemberg-lab.github.io/scRNA.seq.datasets/) were
421 merged into this list. Only animal single-cell transcriptomic datasets profiling samples of
422 normal condition were selected. We also manually filtered small scale or low-quality data.
423 Additionally, several other high-quality datasets missing in the previous list were included for
424 comprehensiveness.
425 The expression matrices and metadata of selected datasets were retrieved from GEO,
426 supplementary files of the publication or by directly contacting the authors. Metadata were
427 further manually curated by adding additional description in the paper to acquire the most
428 detailed information of each cell. We unified raw cell type annotation by Cell Ontology[42], a
429 structured controlled vocabulary for cell types. Closest Cell Ontology terms were manually
430 assigned based on Cell Ontology description and context of the study.

431 **Building reference panels for ACA database**

432 Two types of searchable reference panels are built for the ACA database. The first consists of
433 individual datasets with dedicated models trained on each of them, while the second consists
434 of datasets grouped by organ and species, with models trained to align multiple datasets
435 profiling the same species and the same organ.
436 Data preprocessing follows the same procedure as in previous benchmarks. Both cross-
437 dataset batch effect and within-dataset batch effect are manually examined and removed
438 when necessary. For the first type of reference panels, datasets too small (typically < 1,000
439 cells sequenced) are excluded because of insufficient training data. These datasets are still
440 included in the second type of panels, where they are trained jointly with other datasets
441 profiling the same organ in the same species. For each reference panel, four models with
442 different starting points are trained. Latent space visualization, self-projection coverage and
443 accuracy on all reference panels are available on our website.

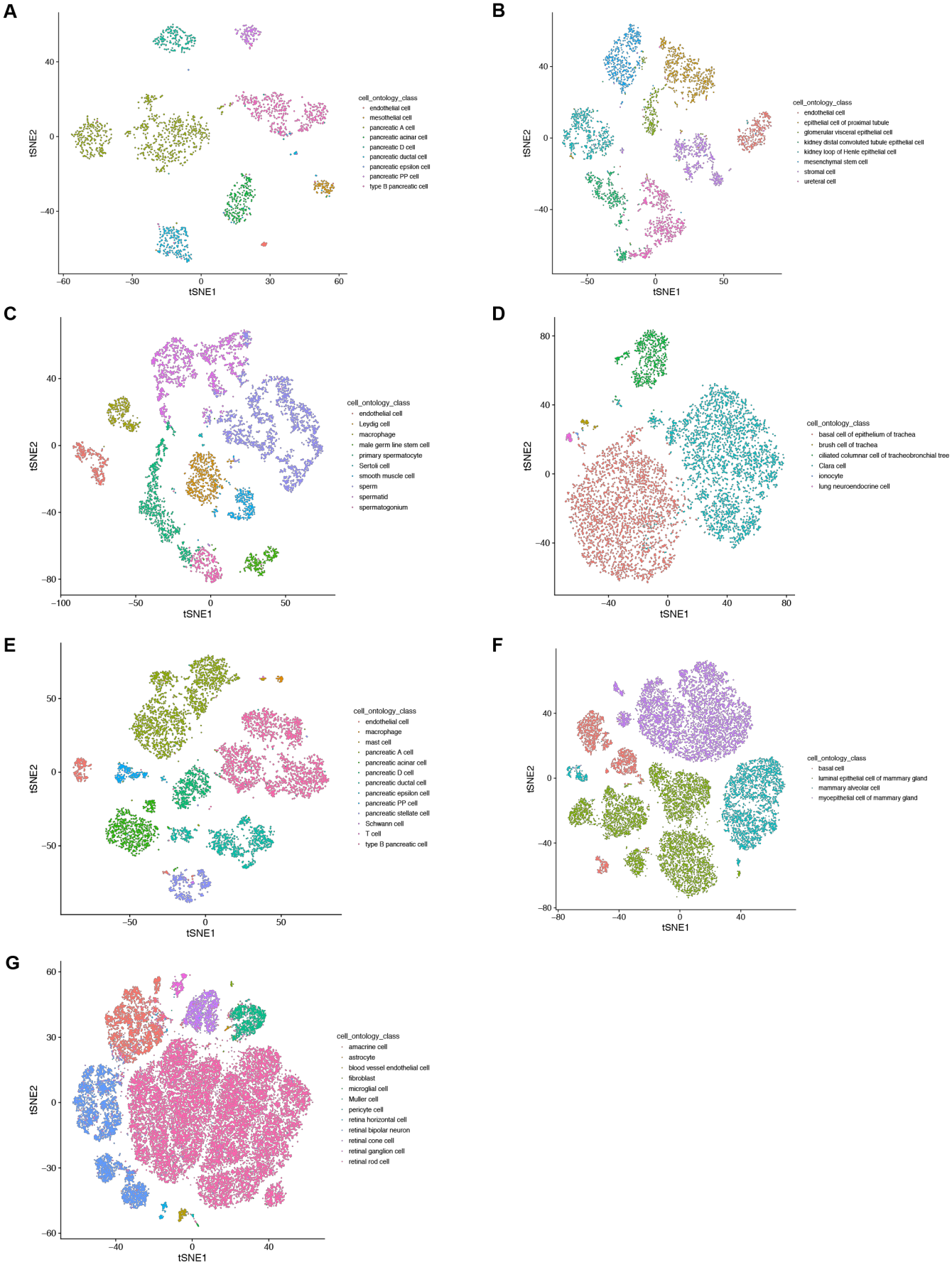
444 **Web interface**

445 For conveniently performing and visualizing Cell BLAST analysis, we built a one-stop web
446 interface. The client-side was made from Vue.js, a the single-page application Javascript
447 framework, and D3.js for cell ontology visualization. We used Koa2, a web framework for
448 Node.js, as the server-side. The Cell BLAST web portal with all accessible curated datasets is
449 deployed on Huawei Cloud.



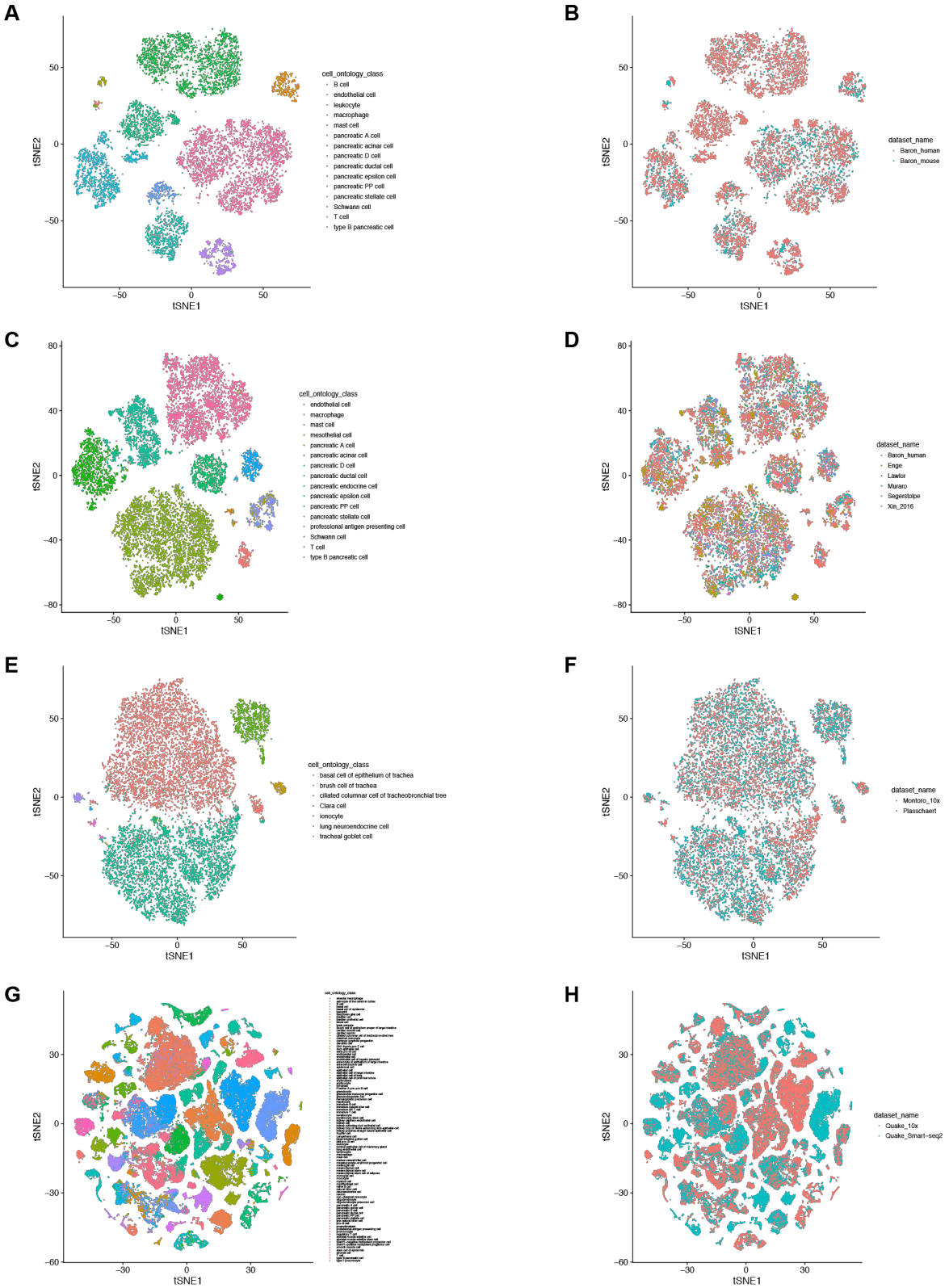
Supplementary Fig. 1 Comparing low dimensional space cell type resolution of different dimension reduction methods.

Nearest neighbor cell type mean average precision (MAP) is used to evaluate how well biological similarity is captured. MAP can be thought of as a generalization to nearest neighbor accuracy, with larger values indicating higher cell type resolution, thus more suitable for cell querying. Methods that did not finish under 2-hour time limit are marked as N.A.



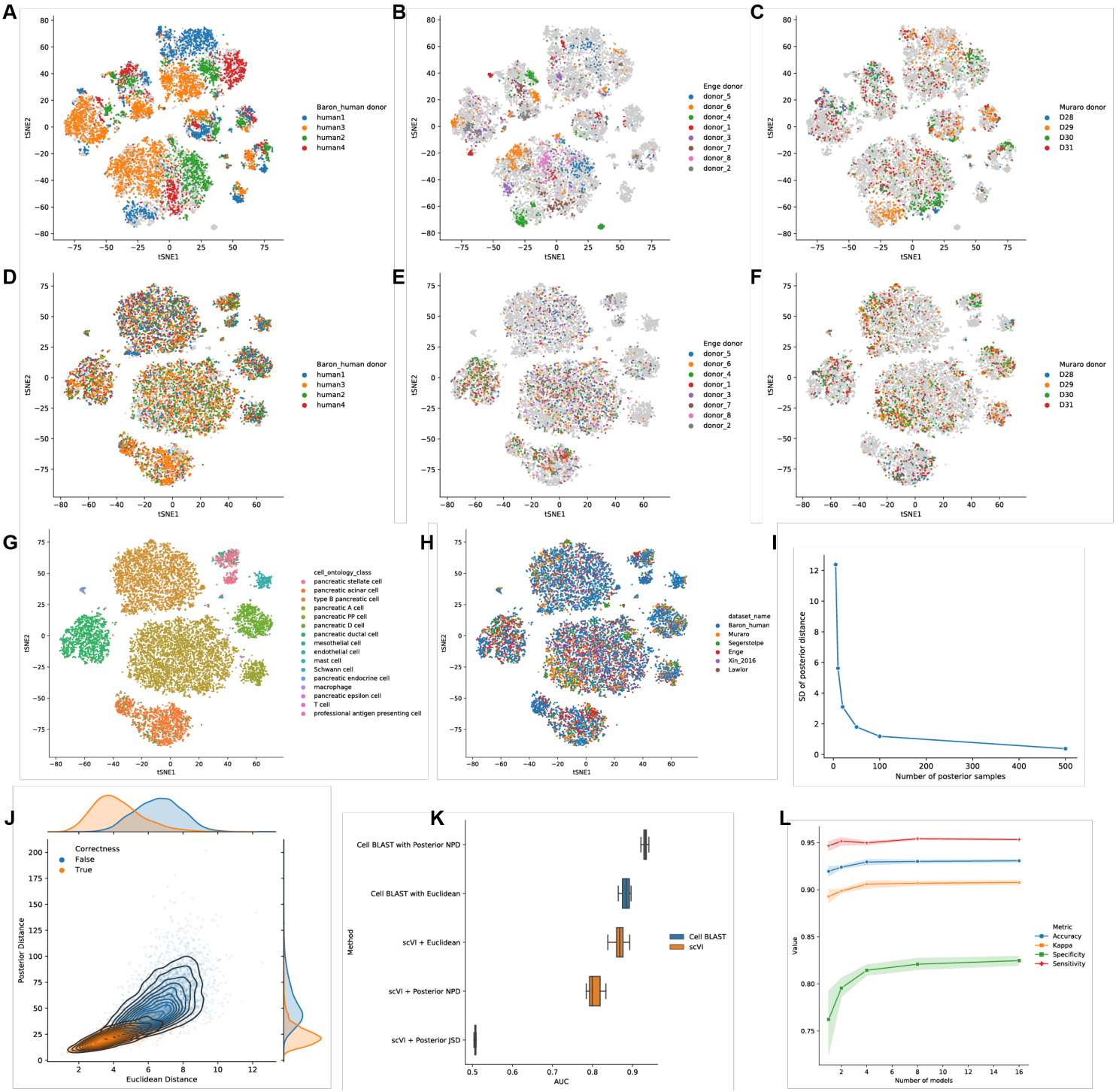
Supplementary Fig. 2 t-SNE visualization of latent spaces learned by our model.

(A) "Muraro"[43], (B) "Adam"[44], (C) "Guo"[45], (D) "Plasschaert"[15], (E) "Baron_human"[46], (F) "Bach"[47], (G) "Macosko"[48].



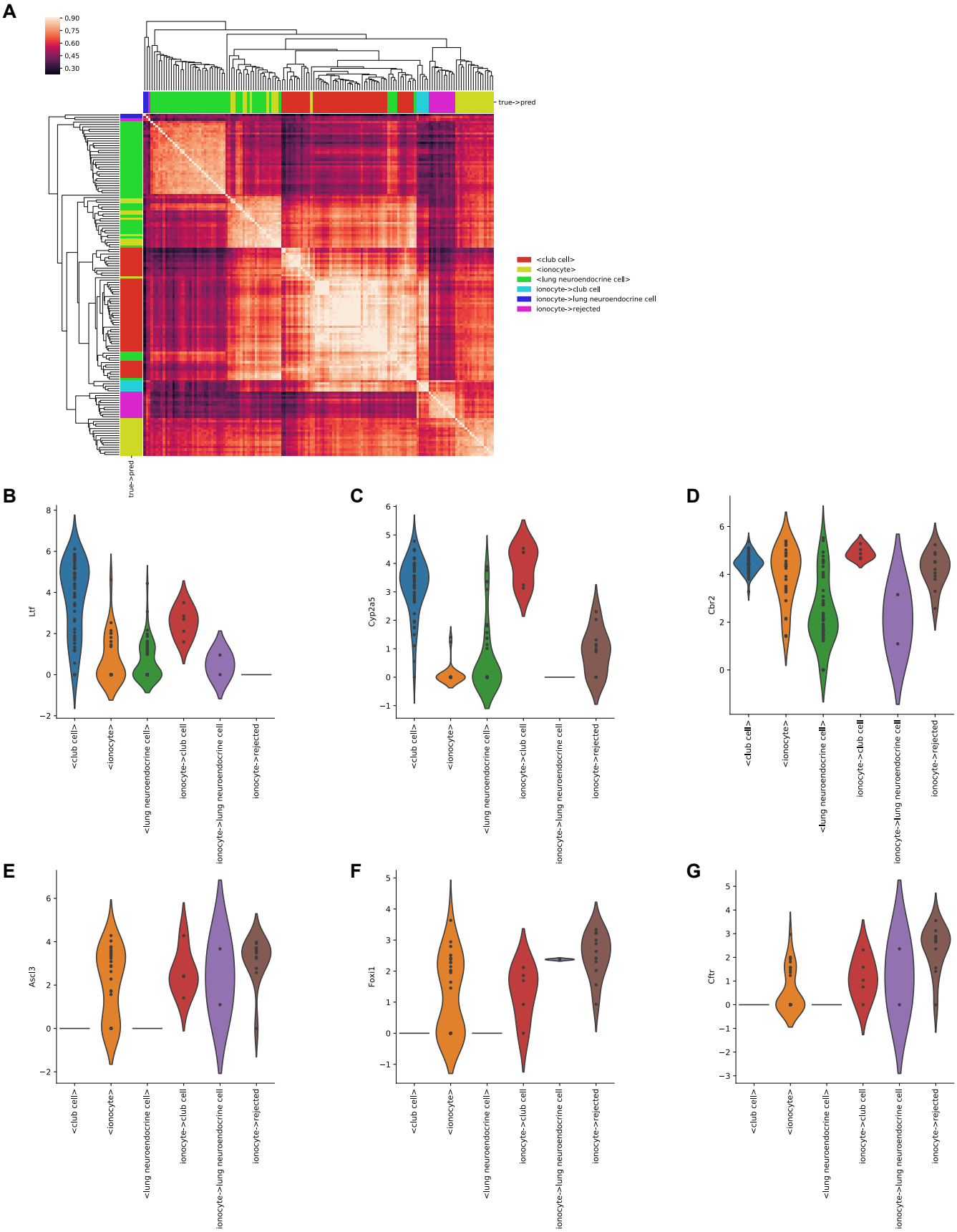
Supplementary Fig. 3 t-SNE visualization of latent spaces learned by our model on combinations of multiple datasets, with batch effect corrected.

Figures in the left column color cells by their cell types, while figures in the right column color cells by dataset. (A-B) "Baron_human"[46] and "Baron_mouse"[46]; (C-D) "Baron_human"[46], "Muraro"[43], "Enge"[49], "Segerstolpe"[50], "Xin_2016"[51] and "Lawlor"[52]; (E-F) "Montoro_10x"[14] and "Plasschaert"[15]; (G-H) "Quake_Smart-seq2"[18] and "Quake_10x"[18].



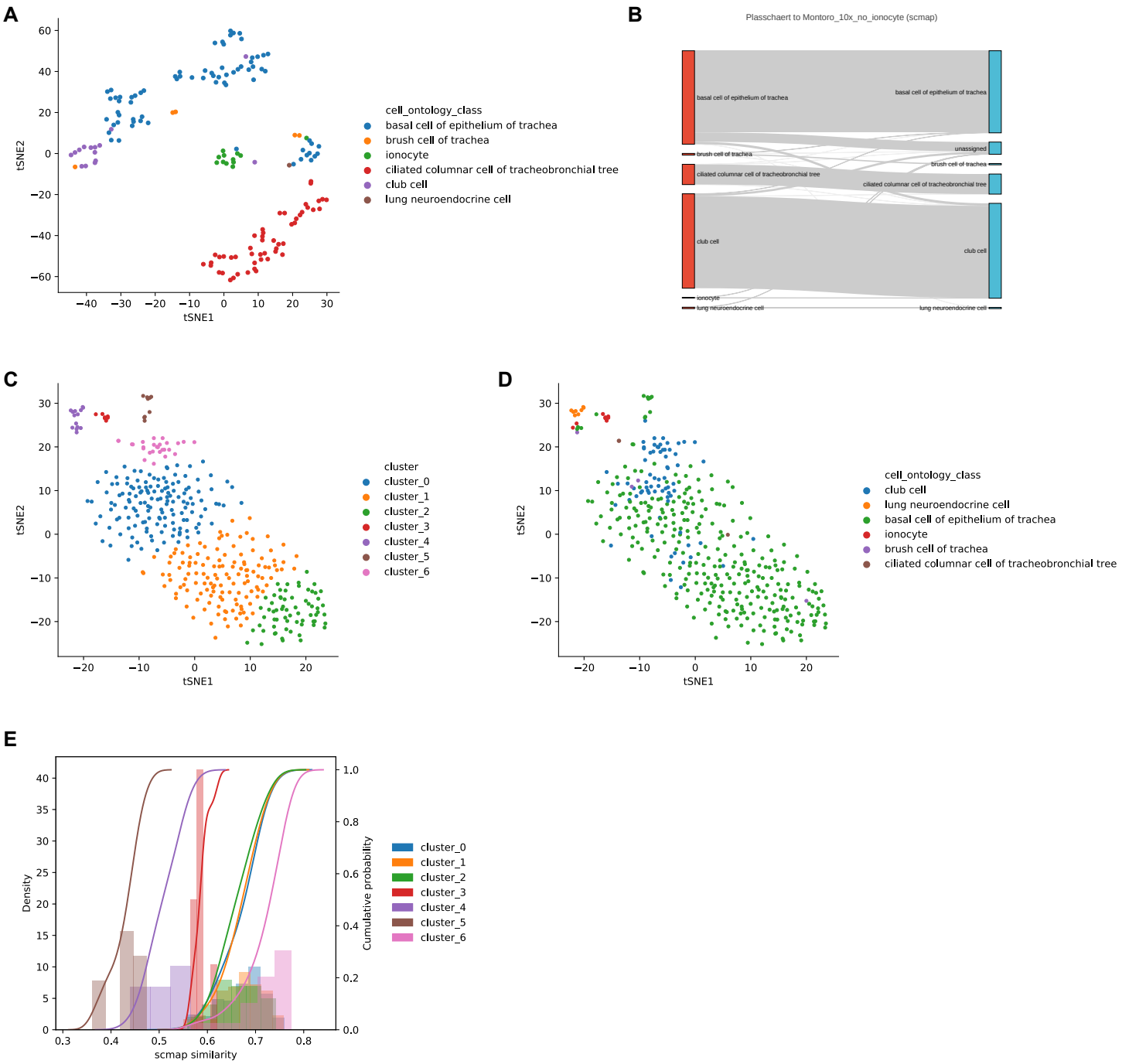
Supplementary Fig. 4 Multi-level batch correction and Cell BLAST strategy optimization.

(A-C) Latent space learned only with cross-dataset batch correction, colored by (A) donor in "Baron_human"[46], (B) donor in "Enge"[49], (C) donor in "Muraro"[43], respectively. (D-H) Latent space learned with both cross-dataset batch correction and within-dataset batch correction, colored by (D) donor in "Baron_human"[46], (E) donor in "Enge"[49], (F) donor in "Muraro"[43], (G) cell type, (H) dataset, respectively. (I) Standard deviation decreases as number of samples from the posterior increases. (J) Relation between Euclidean distance and posterior distance on "Baron_human"[46] data. Orange points represent cell pairs that are of the same cell type ("positive pairs"), while blue points represent cell pairs of different cell types ("negative pairs"). (K) AUROC of different distance metrics in discriminating cell pairs with the same cell type from cell pairs with different cell types. Note that posterior distribution distances for scVI only lead to decrease in performance, possibly due to improper Gaussian assumption in the posterior. (L) Accuracy, Cohen's κ, specificity and sensitivity all increase as the number of models used for cell querying increases, among which improvement of specificity is most significant.



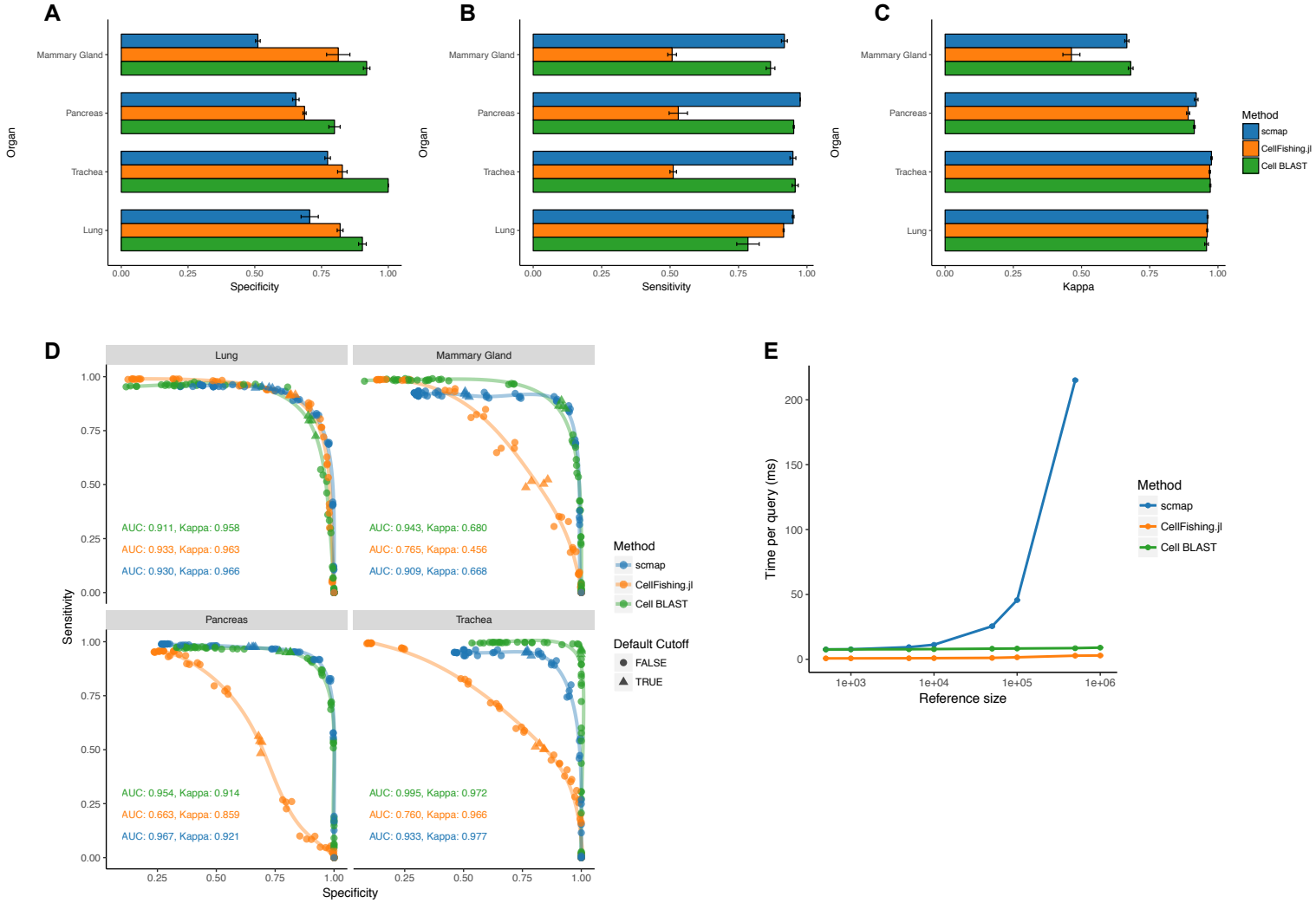
Supplementary Fig. 5 Ionocytes predicted to be club cells are potentially doublets or of intermediate cell state.

(A) Cell-cell correlation heatmap for several cell types of interest. Cells labeled as "<X>" are reference cells in the "Montoro"[14] dataset. Cells labeled as "X->Y" are cells annotated as "X" in the original "Plasschaert"[15] dataset but predicted to be "Y". (B-D) Expression levels of several club cell markers in cell groups of interest. (E-G) Expression levels of several ionocyte markers in cell groups of interest.



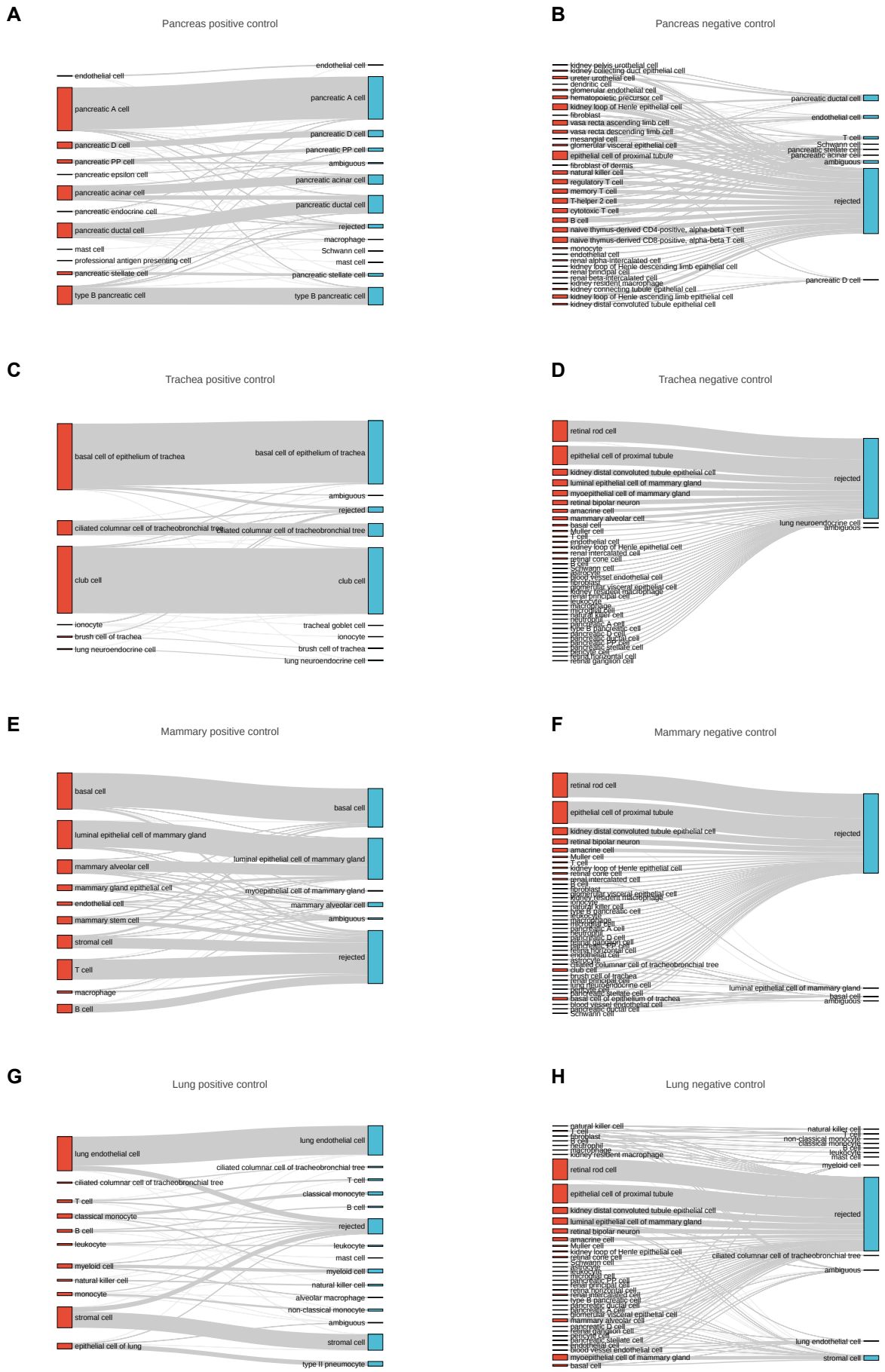
484 **Supplementary Fig. 6 Rejected cells in the "Montoro" - "Plasschaert" query.**

485 (A) t-SNE visualization of Cell BLAST rejected cells, colored by cell type. (B) Sankey plot of scmap prediction.
486 (C, D) t-SNE visualization of scmap rejected cells, colored by unsupervised clustering (C) and cell type (D). (E)
487 scmap similarity distribution in each cluster of scmap rejected cells. The rejected ionocytes do not have the
488 lowest cosine similarity scores to draw enough attention.

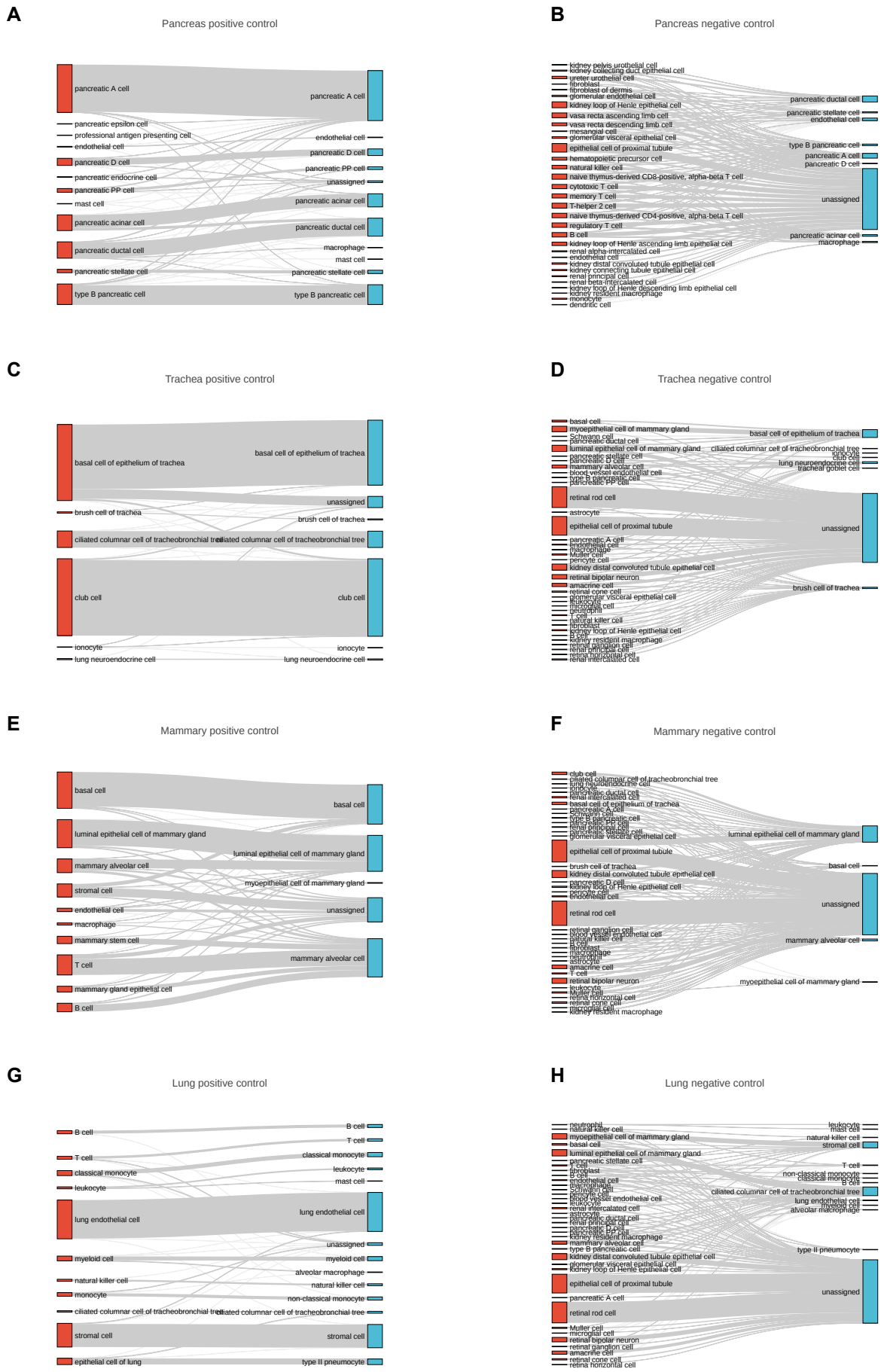


Supplementary Fig. 7 Benchmarking cell querying.

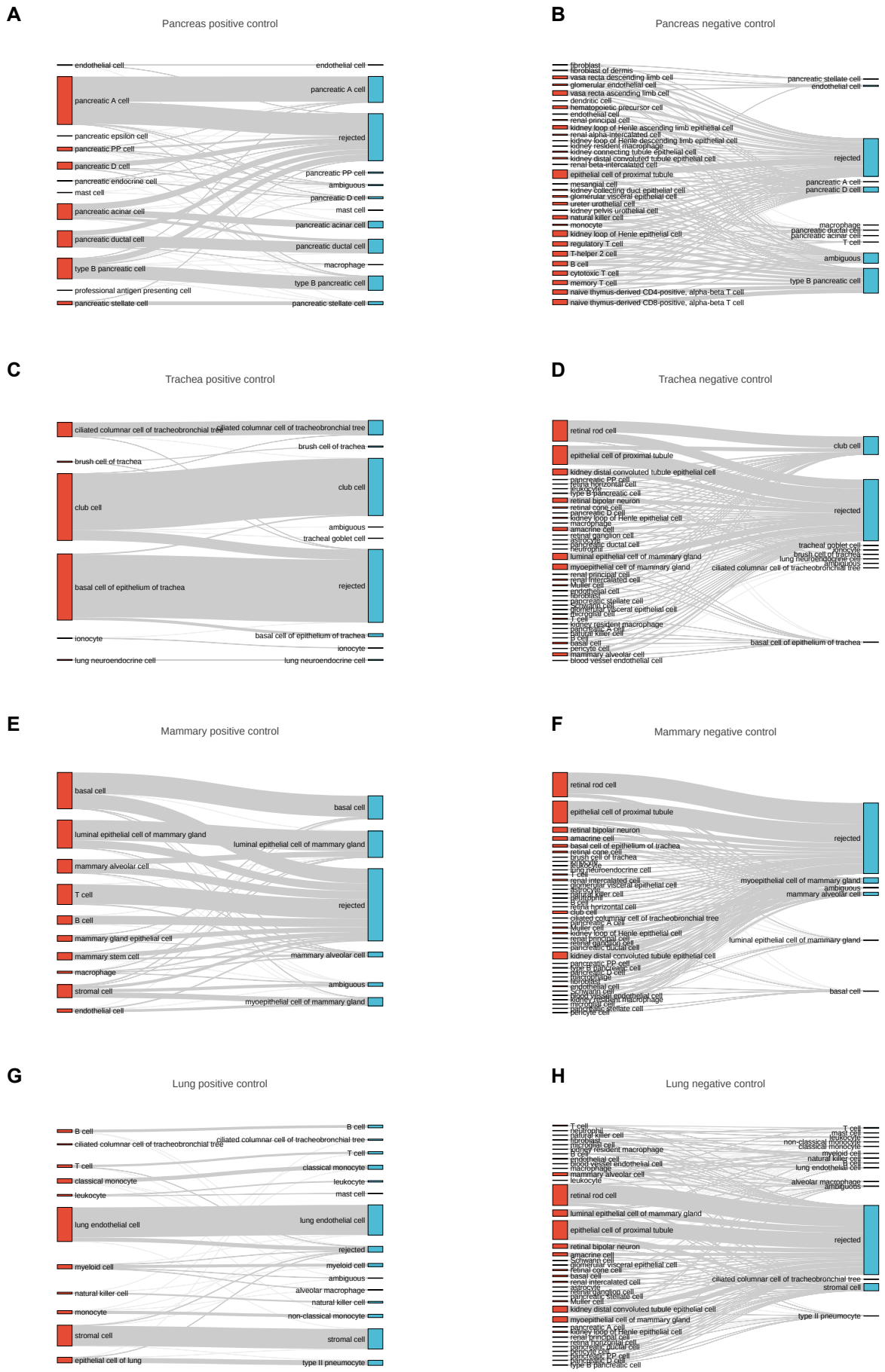
(A-C) Querying specificity (A), sensitivity (B) and Cohen's κ (C) for different methods under the default setting. (D) ROC curve of cell querying in four different groups of test datasets. Cohen's κ values in bottom left of each subpanel correspond to the optimal point on ROC curve. Points corresponding to each method's default cutoff (scmap: cosine distance = 0.5, CellFishing.jl: Hamming distance = 110, Cell BLAST: p-value = 0.05) are marked as triangles. Note that CellFishing.jl does not provide a default cutoff, so we chose Hamming distance of 110 which the closest to balancing sensitivity and specificity, but it's still far from being stable across different datasets. (E) Querying speed on references of different sizes subsampled from the 1M mouse brain dataset[35].



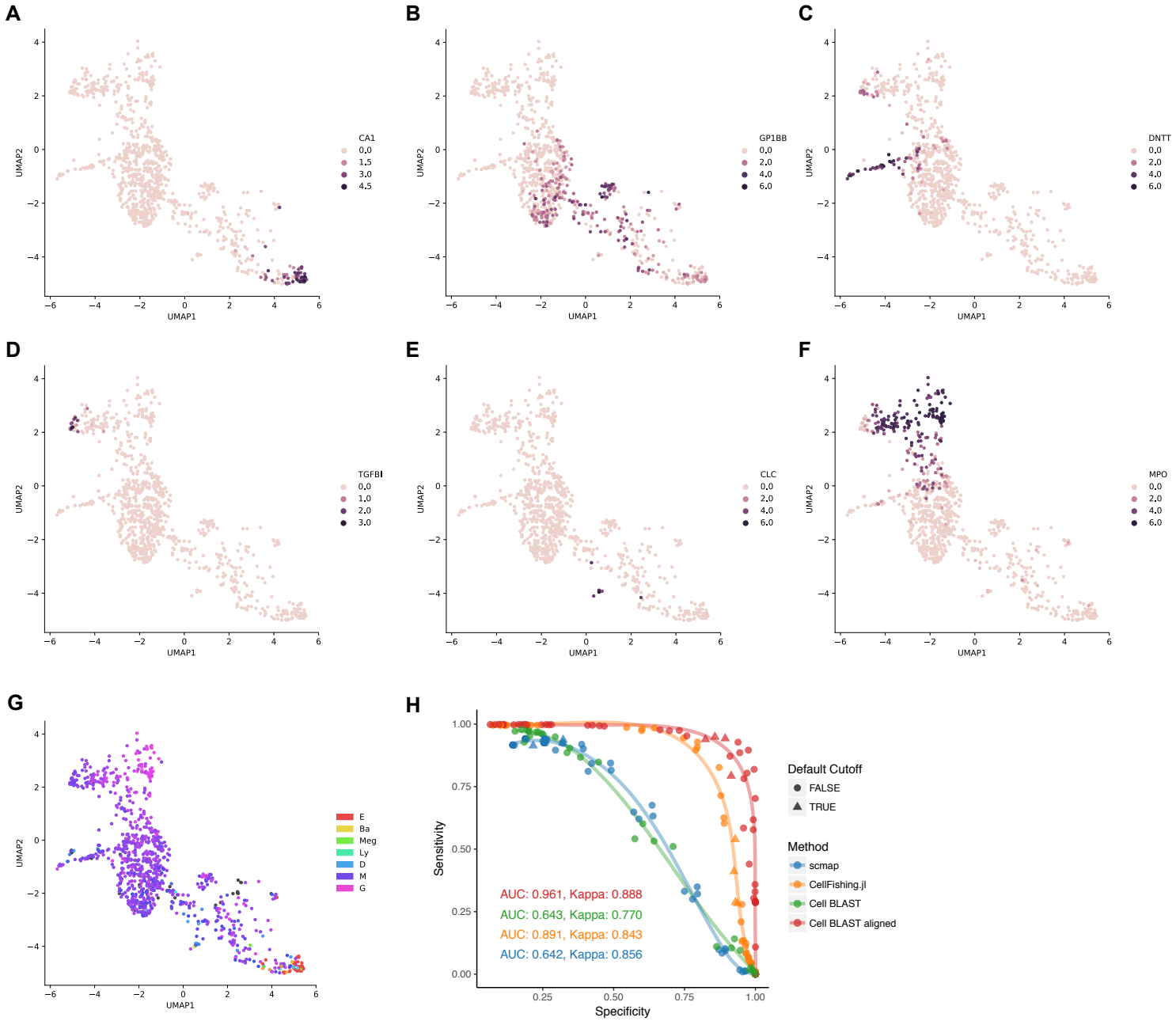
498 **Supplementary Fig. 8 Sankey plots for Cell BLAST in the cell querying benchmark.**



499 **Supplementary Fig. 9 Sankey plots for scmap in the cell querying benchmark.**



500 **Supplementary Fig. 10 Sankey plots for CellFishing.jl in the cell querying benchmark.**



Supplementary Fig. 11 Using "online tuning" in hematopoietic progenitor and *Tabula Muris*[18] spleen data.

UMAP visualization of the "Velten"[17] dataset, colored by expression of lineage markers, including CA1 for erythrocyte lineage (A), GP1BB for megakaryocyte lineage (B), DNTT for B cell lineage, TGFBI for monocyte, dendritic cell lineage (D), CLC for eosinophil, basophil, mast cell lineage (E), MPO for neutrophil lineage (F), and scmap predicted cell fate distribution (G). (H) ROC curve of cell querying in *Tabula Muris*[18] spleen data. Cohen's κ values in bottom left of each subpanel correspond to the optimal point on ROC curve. Points corresponding to each method's default cutoff (scmap: cosine distance = 0.5, CellFishing.jl: Hamming distance = 110, Cell BLAST: p-value = 0.05) are marked as triangles.

510
511 **Supplementary Fig. 12 ACA database and Cell BLAST Web portal.**

512 (A) Comparison of cell number in different single-cell transcriptomics databases. (B) Composition of different
513 single-cell sequencing platforms in ACA. (C) Home page of the Web interface. (D) Web interface showing
514 results of an example query.

Dataset Name	Organism	Organ Profiled	Experimental Platform	Used In
Guo[45]	Human	Testis	10x[53]	DR
Muraro[43]		Pancreas	CEL-Seq2[54]	DR, BC
Xin_2016[51]			SMARTer[55]	BC
Lawlor[52]			SMARTer[55]	BC
Segerstolpe[50]			Smart-seq2[56]	BC
Enge[49]			Smart-seq2[56]	BC
Baron_human[46]			inDrop[57]	DR, BC
Baron_mouse[46]	Mouse		inDrop[57]	BC
Adam[44]		Kidney	Drop-seq[48]	DR
Plasschaert[15]		Trachea	inDrop[57]	DR, BC
Montoro_10x[14]			10x[53]	BC
Macosko[48]		Retina	Drop-seq[48]	DR
Bach[47]		Mammary Gland	10x[53]	DR
Quake_Smart-seq2[18]		20 Organs	Smart-seq2[56]	BC
Quake_10x[18]		12 Organs	10x[53]	BC

Supplementary Table. 1 Datasets used in dimensionality reduction and batch effect correction benchmarking.

DR, dimension reduction benchmarking; BC, batch effect correction benchmarking

Group	Role	Dataset Name	Organism	Organ Profiled	Experimental Platform
Pancreas	Reference	Baron_human[46]	Human	Pancreas	inDrop[57]
		Xin_2016[51]			SMARTer[55]
		Lawlor[52]			SMARTer[55]
	Positive control query	Muraro[43]			CEL-Seq2[54]
		Segerstolpe[50]			Smart-seq2[56]
		Enge[49]			Smart-seq2[56]
	Negative control query	Wu_human[58]		Kidney	10x[53]
		Zheng[53]		PBMC	10x[53]
		Philippeos[59]		Skin	Smart-seq2[56]
Trachea	Reference	Montoro_10x[14]	Mouse	Trachea	10x[53]
	Positive control query	Plasschaert[15]			inDrop[57]
	Negative control query	Baron_mouse[46]		Pancreas	inDrop[57]
		Park[60]		Kidney	10x[53]
		Bach[47]		Mammary Gland	10x[53]
		Macosko[48]		Retina	Drop-seq[48]
Mammary Gland	Reference	Bach[47]	Mouse	Mammary Gland	10x[53]
	Positive control query	Giraddi_10x[61]			10x[53]
		Quake_Smart-seq2_ Mammary_Gland[18]			Smart-seq2[56]
		Quake_10x_ Mammary_Gland[18]			10x[53]
	Negative control query	Baron_mouse[46]		Pancreas	inDrop[57]
		Park[60]		Kidney	10x[53]
		Plasschaert[15]		Trachea	inDrop[57]

		Macosko[48]		Retina	Drop-seq[48]
Lung	Reference	Quake_10x_ Lung[18]	Mouse	Lung	10x[53]
	Positive control query	Quake-Smart-seq2_ Lung[18]			Smart-seq2[56]
	Negative control query	Baron_mouse[46]		Pancreas	inDrop[57]
		Park[60]		Kidney	10x[53]
		Bach[47]		Mammary Gland	10x[53]
		Plasschaert[15]		Trachea	inDrop[57]
Spleen	Reference	Quake_10x_ Spleen[18]	Mouse	Spleen	10x[53]
	Positive control query	Quake_Smart-seq2_ Spleen[18]			Smart-seq2[56]
	Negative control query	Baron_mouse[46]		Pancreas	inDrop[57]
		Park[60]		Kidney	10x[53]
		Macosko[48]		Retina	Drop-seq[48]
		Bach[47]		Mammary Gland	10x[53]

Supplementary Table. 2 Datasets used in cell query benchmarking.

Supplementary Table. 3 Raw data of benchmarking results.

Supplementary Table. 4 Datasets in ACA.

525 **Main text references**

526 1. Altschul, S.F., Gish, W., Miller, W., Myers, E.W. & Lipman, D.J. Basic local
527 alignment search tool. *J Mol Biol* **215**, 403-410 (1990).
528 2. Kiselev, V.Y., Yiu, A. & Hemberg, M. scmap: projection of single-cell RNA-seq data
529 across data sets. *Nat Methods* **15**, 359-362 (2018).
530 3. Srivastava, D., Iyer, A., Kumar, V. & Sengupta, D. CellAtlasSearch: a scalable search
531 engine for single cells. *Nucleic Acids Res* **46**, W141-W147 (2018).
532 4. Sato, K., Tsuyuzaki, K., Shimizu, K. & Nikaido, I. CellFishing.jl: an ultrafast and
533 scalable cell search method for single-cell RNA sequencing. *Genome Biol* **20**, 31
534 (2019).
535 5. Tung, P.Y. et al. Batch effects and the effective design of single-cell gene expression
536 studies. *Sci Rep* **7**, 39921 (2017).
537 6. Haghverdi, L., Lun, A.T.L., Morgan, M.D. & Marioni, J.C. Batch effects in single-
538 cell RNA-sequencing data are corrected by matching mutual nearest neighbors. *Nat*
539 *Biotechnol* **36**, 421-427 (2018).
540 7. Butler, A., Hoffman, P., Smibert, P., Papalexi, E. & Satija, R. Integrating single-cell
541 transcriptomic data across different conditions, technologies, and species. *Nat*
542 *Biotechnol* **36**, 411-420 (2018).
543 8. Lopez, R., Regier, J., Cole, M.B., Jordan, M.I. & Yosef, N. Deep generative modeling
544 for single-cell transcriptomics. *Nat Methods* **15**, 1053-1058 (2018).
545 9. Ding, J., Condon, A. & Shah, S.P. Interpretable dimensionality reduction of single
546 cell transcriptome data with deep generative models. *Nat Commun* **9**, 2002 (2018).
547 10. Grønbech, C.H. et al. scVAE: Variational auto-encoders for single-cell gene
548 expression data. *bioRxiv preprint*, 318295 (2019).
549 11. Wang, D. & Gu, J. VASC: Dimension Reduction and Visualization of Single-cell
550 RNA-seq Data by Deep Variational Autoencoder. *Genomics, proteomics*
551 *bioinformatics* (2018).
552 12. Pierson, E. & Yau, C. ZIFA: Dimensionality reduction for zero-inflated single-cell
553 gene expression analysis. *Genome Biol* **16**, 241 (2015).
554 13. Risso, D., Perraudeau, F., Gribkova, S., Dudoit, S. & Vert, J.P. A general and flexible
555 method for signal extraction from single-cell RNA-seq data. *Nat Commun* **9**, 284
556 (2018).
557 14. Montoro, D.T. et al. A revised airway epithelial hierarchy includes CFTR-expressing
558 ionocytes. *Nature* **560**, 319-324 (2018).
559 15. Plasschaert, L.W. et al. A single-cell atlas of the airway epithelium reveals the CFTR-
560 rich pulmonary ionocyte. *Nature* **560**, 377-381 (2018).
561 16. Tusi, B.K. et al. Population snapshots predict early haematopoietic and erythroid
562 hierarchies. *Nature* **555**, 54-60 (2018).
563 17. Velten, L. et al. Human haematopoietic stem cell lineage commitment is a continuous
564 process. *Nat Cell Biol* **19**, 271-281 (2017).
565 18. Tabula Muris, C. et al. Single-cell transcriptomics of 20 mouse organs creates a
566 Tabula Muris. *Nature* **562**, 367-372 (2018).
567 19. Abugessaisa, I. et al. SCPortalen: human and mouse single-cell centric database.
568 *Nucleic Acids Res* **46**, D781-D787 (2018).
569 20. Cao, Y., Zhu, J., Jia, P. & Zhao, Z. scRNASeqDB: A Database for RNA-Seq Based
570 Gene Expression Profiles in Human Single Cells. *Genes (Basel)* **8** (2017).

571 **Supplementary references**

572 21. Makhzani, A., Shlens, J., Jaitly, N., Goodfellow, I. & Frey, B. Adversarial
573 autoencoders. *arXiv preprint* (2015).
574 22. Love, M.I., Huber, W. & Anders, S. Moderated estimation of fold change and
575 dispersion for RNA-seq data with DESeq2. *Genome Biol* **15**, 550 (2014).
576 23. Abadi, M. et al. in 12th USENIX Symposium on Operating Systems Design and
577 Implementation 265-283 (2016).
578 24. Ganin, Y. & Lempitsky, V. Unsupervised domain adaptation by backpropagation.
579 *arXiv preprint* (2014).
580 25. Xie, Q., Dai, Z., Du, Y., Hovy, E. & Neubig, G. in Advances in Neural Information
581 Processing Systems 585-596 (2017).
582 26. Tzeng, E., Hoffman, J., Saenko, K. & Darrell, T. in Proceedings of the IEEE
583 Conference on Computer Vision and Pattern Recognition 7167-7176 (2017).
584 27. Goodfellow, I. et al. in Advances in neural information processing systems 2672-2680
585 (2014).
586 28. Tange, O. Gnu parallel 2018. (2018).
587 29. Baglama, J., Reichel, L. & Lewis, B.J.R.p.v. irlba: Fast truncated singular value
588 decomposition and principal components analysis for large dense and sparse matrices.
589 **2** (2017).
590 30. Paszke, A. et al. Automatic differentiation in pytorch. (2017).
591 31. Herrero, J. et al. Ensembl comparative genomics resources. *Database (Oxford)* **2016**
592 (2016).
593 32. Lun, A.T., McCarthy, D.J. & Marioni, J.C. A step-by-step workflow for low-level
594 analysis of single-cell RNA-seq data with Bioconductor. *F1000Res* **5**, 2122 (2016).
595 33. Alavi, A., Ruffalo, M., Parvangada, A., Huang, Z. & Bar-Joseph, Z. A web server for
596 comparative analysis of single-cell RNA-seq data. *Nat Commun* **9**, 4768 (2018).
597 34. Han, X. et al. Mapping the Mouse Cell Atlas by Microwell-Seq. *Cell* **172**, 1091-1107
598 e1017 (2018).
599 35. 10x-Genomics in 1.3 Million Brain Cells from E18 Mice (2017).
600 36. Marques, S. et al. Oligodendrocyte heterogeneity in the mouse juvenile and adult
601 central nervous system. *Science* **352**, 1326-1329 (2016).
602 37. Pedregosa, F. et al. Scikit-learn: Machine learning in Python. **12**, 2825-2830 (2011).
603 38. Maaten, L.v.d. & Hinton, G. Visualizing data using t-SNE. *Journal of machine*
604 *learning research* **9**, 2579-2605 (2008).
605 39. McInnes, L., Healy, J. & Melville, J. Umap: Uniform manifold approximation and
606 projection for dimension reduction. *arXiv preprint* (2018).
607 40. Becht, E. et al. Dimensionality reduction for visualizing single-cell data using UMAP.
608 *Nat Biotechnol* (2018).
609 41. Barrett, T. et al. NCBI GEO: archive for functional genomics data sets--update.
610 *Nucleic Acids Res* **41**, D991-995 (2013).
611 42. Diehl, A.D. et al. The Cell Ontology 2016: enhanced content, modularization, and
612 ontology interoperability. *J Biomed Semantics* **7**, 44 (2016).
613 43. Muraro, M.J. et al. A Single-Cell Transcriptome Atlas of the Human Pancreas. *Cell*
614 *Syst* **3**, 385-394 e383 (2016).
615 44. Adam, M., Potter, A.S. & Potter, S.S. Psychrophilic proteases dramatically reduce
616 single-cell RNA-seq artifacts: a molecular atlas of kidney development. *Development*
617 **144**, 3625-3632 (2017).
618 45. Guo, J. et al. The adult human testis transcriptional cell atlas. *Cell Res* **28**, 1141-1157
619 (2018).

620 46. Baron, M. et al. A Single-Cell Transcriptomic Map of the Human and Mouse
621 Pancreas Reveals Inter- and Intra-cell Population Structure. *Cell Syst* **3**, 346-360 e344
622 (2016).
623 47. Bach, K. et al. Differentiation dynamics of mammary epithelial cells revealed by
624 single-cell RNA sequencing. *Nat Commun* **8**, 2128 (2017).
625 48. Macosko, E.Z. et al. Highly Parallel Genome-wide Expression Profiling of Individual
626 Cells Using Nanoliter Droplets. *Cell* **161**, 1202-1214 (2015).
627 49. Enge, M. et al. Single-Cell Analysis of Human Pancreas Reveals Transcriptional
628 Signatures of Aging and Somatic Mutation Patterns. *Cell* **171**, 321-330 e314 (2017).
629 50. Segerstolpe, A. et al. Single-Cell Transcriptome Profiling of Human Pancreatic Islets
630 in Health and Type 2 Diabetes. *Cell Metab* **24**, 593-607 (2016).
631 51. Xin, Y. et al. RNA Sequencing of Single Human Islet Cells Reveals Type 2 Diabetes
632 Genes. *Cell Metab* **24**, 608-615 (2016).
633 52. Lawlor, N. et al. Single-cell transcriptomes identify human islet cell signatures and
634 reveal cell-type-specific expression changes in type 2 diabetes. *Genome Res* **27**, 208-
635 222 (2017).
636 53. Zheng, G.X. et al. Massively parallel digital transcriptional profiling of single cells.
637 *Nat Commun* **8**, 14049 (2017).
638 54. Hashimshony, T. et al. CEL-Seq2: sensitive highly-multiplexed single-cell RNA-Seq.
639 *Genome Biol* **17**, 77 (2016).
640 55. Verboom, K. et al. SMARTer single cell total RNA sequencing. *bioRxiv preprint*,
641 430090 (2018).
642 56. Picelli, S. et al. Full-length RNA-seq from single cells using Smart-seq2. *Nat Protoc*
643 **9**, 171-181 (2014).
644 57. Klein, A.M. et al. Droplet barcoding for single-cell transcriptomics applied to
645 embryonic stem cells. *Cell* **161**, 1187-1201 (2015).
646 58. Wu, H. et al. Comparative Analysis and Refinement of Human PSC-Derived Kidney
647 Organoid Differentiation with Single-Cell Transcriptomics. *Cell Stem Cell* **23**, 869-
648 881 e868 (2018).
649 59. Philippeos, C. et al. Spatial and Single-Cell Transcriptional Profiling Identifies
650 Functionally Distinct Human Dermal Fibroblast Subpopulations. *J Invest Dermatol*
651 **138**, 811-825 (2018).
652 60. Park, J. et al. Single-cell transcriptomics of the mouse kidney reveals potential
653 cellular targets of kidney disease. *Science* **360**, 758-763 (2018).
654 61. Giraddi, R.R. et al. Single-Cell Transcriptomes Distinguish Stem Cell State Changes
655 and Lineage Specification Programs in Early Mammary Gland Development. *Cell*
656 *Rep* **24**, 1653-1666 e1657 (2018).
657