TITLE PAGE

SHARP: Single-cell RNA-seq Hyper-fast and Accurate Processing via Ensemble Random Projection

Shibiao Wan[1,2,3], Junil Kim[1,2,4,5] and Kyoung Jae Won[1,2,4,5*]

[1]Institute for Diabetes, Obesity and Metabolism, Perelman School of Medicine, University of Pennsylvania, Philadelphia, 19104, Pennsylvania, USA

[2]Department of Genetics, University of Pennsylvania, Philadelphia, 19104, Pennsylvania, USA

[3]Center for Applied Bioinformatics, St. Jude Children's Research Hospital, Memphis, TN 38105

[4]Biotech Research and Innovation Centre (BRIC), University of Copenhagen, Ole Maaløes Vej 5. 2200 Copenhagen N, Denmark

[5]Novo Nordisk Foundation Center for Stem Cell Biology, DanStem, Faculty of Health and Medical Sciences, University of Copenhagen, Ole Maaløes Vej 5. 2200 Copenhagen N, Denmark

Email: Shibiao Wan (shibiaowan.work@gmail.com), Junil Kim (neocaleb@gmail.com), Kyoung Jae Won (kyoung.won@bric.ku.dk)

[*]Corresponding author. Kyoung Jae Won (kyoung.won@bric.ku.dk)

Running title:

SHARP: single cell RNA-seq clustering

Keywords

Single cell analysis; 10 million single cells; ensemble clustering; random projection; dimension reduction; scRNA-seq

ABSTRACT

To process large-scale single-cell RNA-sequencing (scRNA-seq) data effectively without excessive distortion during dimension reduction, we present SHARP, an ensemble random projection-based algorithm which is scalable to clustering 10 million cells. Comprehensive benchmarking tests on 17 public scRNA-seq datasets demonstrate that SHARP outperforms existing methods in terms of speed and accuracy. Particularly, for large-size datasets (>40,000 cells), SHARP's running speed far excels other competitors while maintaining high clustering accuracy and robustness. To the best of our knowledge, SHARP is the only R-based tool that is scalable to clustering scRNA-seq data with 10 million cells.

INTRODUCTION

By enabling transcriptomic profiling at the individual-cell level, scRNA-seq has been widely applied in various domains of biology and medicine to characterize novel cell types and detect intra-population heterogeneity (Potter 2018). The amount of scRNA-seq data in public domain has increased explosively due to technological development and the efforts to obtain large-scale transcriptomic profiling of cells (Han et al. 2018). Computational algorithms to process and analyze large-scale high-dimensional single-cell data are essential. To cluster high dimensional scRNA-seq data, dimension reduction algorithms such as principal component analysis (PCA) (Joliffe and Morgan 1992) or independent component analysis (ICA) (Hyvarinen and Oja 2000) has been successfully applied to process and to visualize high dimensional scRNA-seq data. However, it requires considerable time to obtain principal or independent components as the number of cells increases. It is also notable that dimension reduction decreases processing time at the cost of losing original cell-to-cell distances. For instance, t-distributed stochastic neighbor embedding (tSNE) (van der Maaten 2014) effectively visualizes multi-dimensional data into a reduced-dimensional space. However, tSNE distorts the distance between cells for its visualization. Besides, tSNE requires considerable time for large-

scale scRNA-seq data visualization and clustering.

Random projection (RP) (Bingham and Mannila 2001) has been suggested as a powerful dimension reduction method. Based on the Johnson-Lindenstrauss lemma (Johnson and Lindenstrauss 1984), RP reduces the dimension while the distances between the points are approximately preserved (Frankl and Maehara 1988). Theoretically, RP is very fast because it does not require calculation of pairwise cell-to-cell distances or principle components.

To effectively handle very large-scale scRNA-seq data without excessive distortion of cell-to-cell distances, we developed SHARP (Supplemental Code), a hyper-fast clustering algorithm based on ensemble random projection (RP) (Fig. 1a and Methods). RP (Bingham and Mannila 2001) projects the original D-dimensional data into a d-dimensional ($d << D$) subspace, using a $d \times D$- dimensional random matrix **R** whose elements conform to a distribution with zero mean and unit variance. Notably, RP preserves cell-to-cell distances even in a much lower-dimensional space and is robust to missing values, which provides a well-suited condition for clustering high-dimensional scRNA-seq data. SHARP dramatically reduced the running cost for clustering while guaranteeing far enhanced clustering performance especially for large-size scRNA-seq datasets. SHARP requires the time complexity of only $O(N\log(N)\sqrt{D})$ for scRNA-seq data with N cells and D genes. Compared with it, a simple hierarchical clustering algorithm requires $O(N^2D)$ (Murtagh and Legendre 2014) to calculate distance between cells. tSNE combined with the k-means algorithm requires $O(DN\log(N))$ (van der Maaten 2014) and a simple PCA requires $O(ND \cdot \min(N, D))$ for data reduction (Bingham and Mannila 2001).

There have been previous ensemble-based approaches for RP (Fern and Brodley 2003; Bertoni and Valentini 2006). Compared with them, we developed a strategy specifically for handling scRNA-seq data by using (1) a divide-and-conquer approach for very large-scale scRNA-seq data clustering; (2) a two-layer meta-clustering approach for robust clustering; and (3) a very sparse random projection embedded into

ensemble clustering for hyper-fast clustering.

We showed that SHARP preserves the cell-to-cell distance during dimension reduction and performs clustering much faster than other competitors while the clustering performance is guaranteed. We also demonstrated that SHARP is robust to the dropouts. Benchmarking various clustering algorithms, we found that SHARP is the only R-based algorithm to perform clustering 1.3 million cells (10x Genomics 2017) and can handle clustering even up to 10 million cells.

RESULTS

SHARP employed RP for ultra-fast scRNA-seq clustering

SHARP employed a divide-and-conquer strategy followed by RP to accommodate effective processing of large-scale scRNA-seq data (Fig. 1a and Methods). SHARP processes scRNA-seq data in 4 interconnected steps: (1) data partition, (2) RP based clustering, (3) weighted ensemble clustering and (4) similarity-based meta-clustering. During data partition, the scRNA-seq data is divided into small blocks (random size). It is noted that currently R lacks 64-bit integers support and a scRNA-seq data matrix with >1 million cells (so that the number of elements is usually significantly larger than $2^{31} - 1$) cannot be directly loaded into R. The divide-and-conquer strategy enables SHARP to upload and process more than 1 million cells. The divided data blocks are further processed by RP followed by a hierarchical clustering algorithm. Because the performance of an individual RP-based clustering is volatile, ensemble of several runs of RPs is used. A weighted-ensemble clustering (i.e., wMetaC) algorithm merges individual RP-based clustering results. Finally, a similarity-based ensemble clustering (i.e., sMetaC) approach is to integrate clustering results of each block (Fig. 1a and Methods).

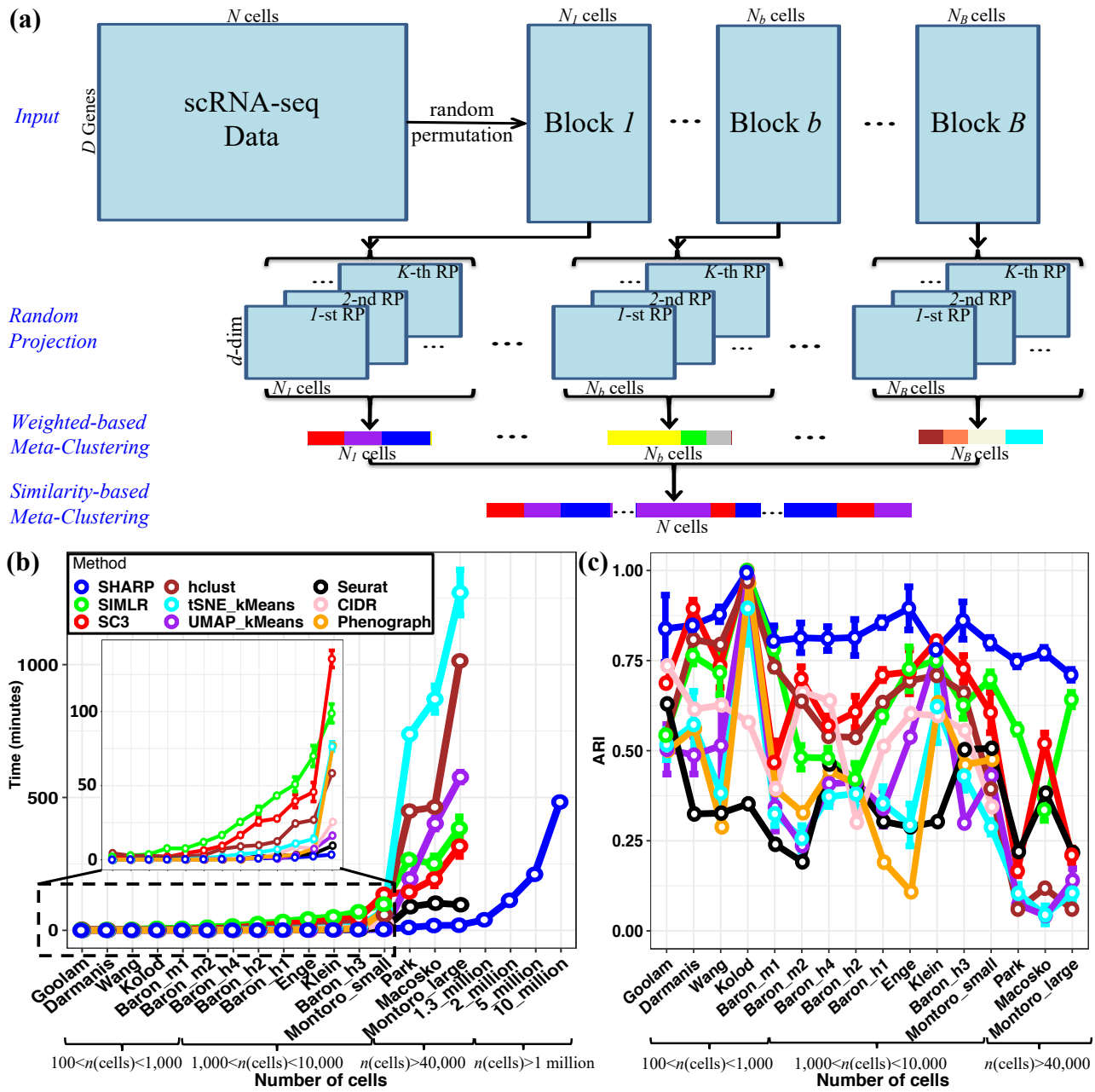

Figure 1 The framework of SHARP. (a) SHARP has 4 steps for clustering: divide-and-conquer, random projection (RP), weighted-based meta clustering, and similarity-based meta-clustering. (b) Running time and (c) clustering performance based on ARI (Hubert and Arabie 1985) of SHARP in 20 single-cell RNA-seq datasets with numbers of single cells ranging from 124 to 10 million (where datasets with 2 million, 5 million and 10 million cells were generated by randomly oversampling the dataset with 1.3 million single cells). For the datasets with >1 million cells, only SHARP can run and only the running time was provided due to lack

106 of the ground-truth clustering labels. All of the results for SHARP were based on 100 runs of SHARP on each
107 dataset. All the tests except for the larger-than-1-million-cell datasets were performed using a single core on
108 an Intel Xeon CPU E5-2699 v4 @ 2.20GHz system with 500GB memory. To run datasets with > 1 million
109 cells, we used 16 cores on the same system. CIDR and Phenograph were unable to produce clustering results
110 for those datasets with number of cells larger than 40,000 (i.e., Park, Macosko and Montoro_large).
111
112 **SHARP is faster than other predictors and is scalable to 1.3 million cells and even up to 10 million cells.**
113 We performed comprehensive benchmarking of SHARP against existing scRNA-seq clustering algorithms
114 including SC3 (Kiselev et al. 2017), SIMLR (Wang et al. 2017), hierarchical clustering and tSNE combined
115 with k-means, UMAP (Becht et al. 2018) with k-means, Seurat (Butler et al. 2018), CIDR (Lin et al. 2017)
116 and Phenograph (Levine et al. 2015) (Fig. 1b-c and Supplemental Methods) using 17 publicly available
117 scRNA-seq datasets whose cell number ranges from 124 to 1.3 million cells (Darmanis et al. 2015; Klein et
118 al. 2015; Kolodziejczyk et al. 2015; Macosko et al. 2015; Baron et al. 2016; Goolam et al. 2016; Wang et al.
119 2016; 10x Genomics 2017; Enge et al. 2017; Montoro et al. 2018; Park et al. 2018) (Supplemental Table S1).
120
121 The benchmarking tests demonstrated dramatic cost reduction by SHARP (Fig. 1b). Reflecting the theoretical
122 running costs, the two classical algorithms (tSNE + kMeans, hierarchical clustering) manifested exponential
123 increase in their processing time as the number of cells increased (Fig. 1b). SC3 (Kiselev et al. 2017) and
124 SIMLR (Wang et al. 2017) showed better performance than the classical clustering approaches, but they still
125 required a considerable amount of time for clustering. The computing cost of SHARP was substantially lower
126 than other clustering algorithms. Remarkably, the required computing cost of SHARP rose roughly linearly
127 even with the very large size of the datasets. For the cells with larger than 40,000, SHARP ran at least 20
128 times faster than SC3 (Kiselev et al. 2017) and SIMLR (Wang et al. 2017) . While CIDR and Phenograph
129 were reported to perform very fast and robustly (Levine et al. 2015; Lin et al. 2017), they were unable to
130 produce clustering results for those datasets with more than 40,000 single cells such as Park (Park et al. 2018),

131 Macosko (Macosko et al. 2015) and Montoro_large (Montoro et al. 2018) datasets (Fig. 1b).

132

133 Notably, SHARP clustered the scRNA-seq with 1.3 million cells in 42 minutes when using a multi-core
134 system (Fig. 1b). Due to the data loading problem (and potential exhaustive memory use), we could not show
135 the running time of other approaches for 1.3 million cells. When using a multi-core system (16 cores) on the
136 Montoro_large dataset (Montoro et al. 2018) with 66,265 cells, SHARP ran remarkably faster than SC3 and
137 SIMLR more than 40 times (Supplemental Fig S1 and Supplemental Methods). The running time of SHARP
138 for 1.3 million cells is even 2 times (42 mins vs 96 mins) faster than that of Seurat for 66,255 cells. We expect
139 far superior performance of SHARP against its competitors in case data loading is feasible.

140

141 To evaluate the performance of SHARP exhaustively and demonstrate the scalability of SHARP, we
142 performed random over-sampling of the mouse brain dataset of 1.3 million cells (10x Genomics 2017) so that
143 we were able to construct even larger sizes of scRNA-seq datasets. For this simulation, we tested up to 10
144 million cells. The running time of SHARP was simply linearly increased with the increasing of cell numbers
145 from 1 million to 10 million. In our system using 16 cores, SHARP needed around 8 hours (i.e., 482.8 minutes)
146 to cluster 10 million cells into 1175 clusters (Fig 1b).

147

148 **The outperforming clustering performance of SHARP is not highly affected by the number of cells**

149 In parallel, we compared the clustering performance using the pre-defined cell types for each dataset
150 (Supplemental Table S1). To evaluate clustering performance, we used adjusted Rand index (ARI) (Hubert
151 and Arabie 1985). ARI (Supplemental Methods) is a similarity metric to measure how accurately a prediction
152 of clustering is made in the unsupervised learning scenarios, which is similar to the accuracy measurement in
153 supervised classification problems. Generally, the larger ARI, the better the predicted clustering is, with +1
154 indicating the predicted clustering is perfectly consistent with the reference, whereas 0 (or negative value)
155 indicating that the predicted clustering is as good as (or worse than) random guess.

156

157 For almost all datasets we tested, SHARP showed better performances (Fig. 1c). It is notable that the
158 performance of other algorithms became generally worse for large datasets (>40,000 single cells). In contrast,
159 SHARP showed an ARI larger than 0.7 regardless of the size of the datasets, demonstrating its robustness.
160 The ARI of Seurat was in general poor even for the small sized datasets. Seurat showed relatively faster speed
161 (despite with worse clustering performance) even with its implementation of tSNE compared to other existing
162 methods except SHARP. This is because Seurat only uses genes with high variations in their expressions,
163 which could affect the clustering performance (Fig. 1c).

164

165 For robust assessment of the clustered results, we also used artificial datasets by mixing cells with known cell
166 types obtained from Tabula Muris (Schaum et al. 2018) (Supplemental Fig S2). SHARP again outperformed
167 other methods in terms of both clustering performance (Supplemental Fig S3) and running time (Supplemental
168 Table S2).

169

170 **SHARP preserves cell-to-cell distance**

171 To explain the robust clustering performance of SHARP, we investigated the degree of distortion caused by
172 dimension reduction and compared the correlation of cell-to-cell distances after reducing dimension using
173 SHARP, PCA and tSNE, respectively. For this, we calculated the pairwise Pearson correlation between each
174 pairs of cells for the original scRNA-seq data and the dimension-reduced data. Reflecting the property of RP,
175 SHARP showed almost perfect similarities in cell-to-cell distance with correlation coefficient > 0.94 even in
176 a dimensional space which is 74 times lower (from 20862 to 279) than the original one (Fig. 2a, Supplemental
177 Fig S4 and Supplemental Methods). Cell-to-cell distances were distorted when dimension reduction was
178 performed to the same number of dimensions using PCA (Fig. 2a). tSNE, an algorithm to visualize high-
179 dimensional data into 2- or 3-dimensional space, showed a lower correlation as expected (Fig. 2a).

180

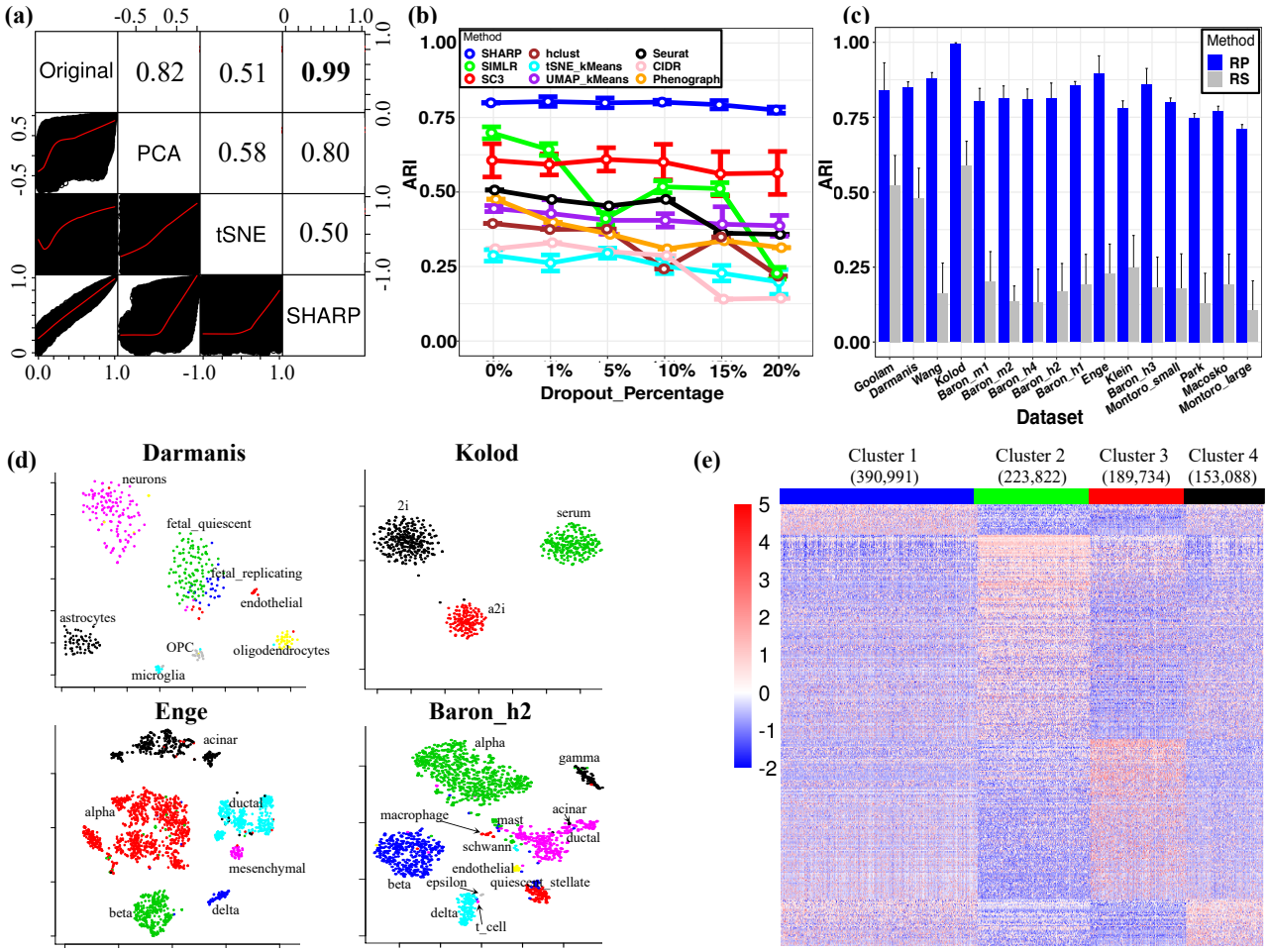

Figure 2. The properties of SHARP. (a) Cell-to-cell distance preservation in SHARP space comparing with that in tSNE and PCA for the Enge (Enge et al. 2017) dataset. The lower triangular part shows the scatter plots of the cell-to-cell distances, whereas the upper triangular part shows the Pearson correlation coefficient (PCC) of the corresponding two spaces. (b) SHARP is robust to the additional dropout events on the Montoro_small (Montoro et al. 2018) dataset. (c) Comparing RP (SHARP uses RP) with random gene selection (RS) in 16 single-cell RNA-seq datasets (Darmanis et al. 2015; Klein et al. 2015; Kolodziejczyk et al. 2015; Macosko et al. 2015; Baron et al. 2016; Goolam et al. 2016; Wang et al. 2016; Enge et al. 2017; Montoro et al. 2018; Park et al. 2018) with the number of single cells ranging from 124 to 66,265. (d) Visualization capabilities of SHARP in the Darmanis (Darmanis et al. 2015), Kolod (Kolodziejczyk et al. 2015), Enge (Enge et al. 2017) and Baron_h2 (Baron et al. 2016) datasets. (e) Cluster-specific marker gene expressions of the top 4 major clusters for the 1.3 million single cells (10x Genomics 2017) by SHARP. The

193 total number of clusters predicted by SHARP is 244. The number in brackets represents the number of single
194 cells in the corresponding cluster.
195
196 **SHARP is robust to dropouts**
197 scRNA-seq suffers a high frequency of dropouts where many of the true expressions are not captured. To
198 evaluate the robustness of SHARP against dropouts, we tested SHARP while increasing dropout rates in the
199 Montoro_small (Montoro et al. 2018) dataset (Fig. 2b). It is of note that we applied additional random
200 dropouts to the current scRNA-seq with a certain level of dropouts. For instance, originally the Montoro_small
201 dataset has the dropout rate of 79.6% for the top 8000 genes in terms of non-zero gene expressions. Based on
202 the original high dropout rate, additional artificial dropouts were imposed which provide more harsh
203 conditions for clustering. Our test results showed that additional 20% of dropout was enough to evaluate the
204 robustness of the clustering algorithms.
205
206 We found that both SHARP and SC3 were robust to the added dropouts (Methods), while we observed poorer
207 performance for the added dropouts in general (Fig 2b). The performance of SIMLR, even though it was
208 better than SC3 when there were no added dropouts, became worse when the added dropout rates were
209 increased over 5%.
210
211 **Cell-to-cell distance is important for clustering results**
212 To evaluate the contribution of RP for clustering, we performed clustering after replaying RP with random
213 selection (RS) of genes while other procedures are unchanged. This violates the condition for RP (a matrix
214 with zero mean and unit variance) and therefore, rough preservation of cell-to-cell distance is no longer
215 guaranteed. Random selection of genes severely undermined the performance (Fig. 2c), suggesting that RP is
216 a major component contributing to the performance of SHARP.
217

218 Furthermore, to demonstrate the effectiveness of wMetaC, we compared wMetaC with the method of
219 averaging gene expressions after multiple runs of RP (we named it as "RP_avg") while other configurations
220 are unchanged. We observed that wMetaC performs better than simply averaging the expression profiles after
221 random projection across all datasets (Supplemental Fig S5). The superiority of SHARP was more evident
222 for those large-size scRNA-seq datasets when the number of cells is larger than 40,000 (e.g., Park, Macosko,
223 and Montoro_large).
224
225 **SHARP is equipped with visualization**
226 SHARP is equipped with tSNE (Fig. 2d and Supplemental Fig S6) and heatmaps (Supplemental Fig S7) to
227 visualize its clustering results (Methods). For instance, the heat map for the Enge dataset (Enge et al. 2017)
228 clearly showed the cell types in pancreas including α (GCG), β (INS), acinar (PRSS1) and δ (SST) cells
229 (Supplemental Fig S7).
230
231 **Clustering 1.3 million cell data using SHARP**
232 Of note, SHARP provides an opportunity to study the million-cell-level dataset. Previous analysis on the
233 scRNA-seq data with 1,306,127 cells from embryonic mice brains (10x Genomics 2017) was performed using
234 the *k*-means and a graph clustering (equivalent to kernel k-means) algorithms (10x Genomics 2017). However,
235 *k*-means cannot identify the optimal number of clusters and it depends on the initial seeds for clustering. Using
236 SHARP, we identified a total of 244 clusters from this dataset (17 clusters with more than 1,000 cells)
237 (Supplemental Table S3). The top 4 clusters among them were found to have clear different expression
238 patterns (Fig. 2e). Gene Ontology (GO) analysis (Supplemental Table S3) show that Cluster 2 is associated
239 with dendrites and Cluster 3 is with axon. We also identified a cluster (Cluster 8) enriched for the genes
240 associated with "non-motile cilium assembly", which is important for brain development and function
241 (Guemez-Gamboa et al. 2014) and immune cells with high IL4 expression (Cluster 14).
242

DISCUSSION

The size of scRNA-seq data is increasing exponentially in recent years. Besides the 1.3 million brain cell dataset we used, we expect larger sizes of datasets to be generated. There is an urgent need for a computational approach to handle large size of datasets efficiently. Unfortunately, we found the majority of the computational algorithms cannot deal with very large size of the data (Fig 1b). Furthermore, the clustering performance became worse as the number of cells increases (Fig 1c).

To address these problems, we developed SHARP based on RP, which, to the best of our knowledge, has not been introduced for scRNA-seq data analysis. RP preserves the distance of the data points even in a lower dimension. Reflecting this, the cell-to-cell distance was greatly well preserved after running SHARP (Fig 2a). The performance comparison demonstrated that SHARP is a very powerful clustering software which far excels other competitors in terms of computation cost and the clustering performance (Fig 1b,c).

To further demonstrate the scalability of SHARP, we have extended our comparison of SHARP against some scalable scRNA-seq analysis methods including bigSCale (Iacono et al. 2018), Geometric Sketching (GeoSketch) (Hie et al. 2019b) and Scanorama (Hie et al. 2019a). As they were designed for different purposes, we only evaluated their scalability by running them with scRNA-seq datasets with different cell sizes and measured the running costs. Results (Supplemental Fig S8) suggested that the computing time for bigScale and Scanorama increased exponentially while SHARP and GeoSketched exerted more powerful scalability. For datasets with cell numbers larger than 200,000, bigScale and Scanorama did not work. These results demonstrate that SHARP's scalability is comparable with other state-of-the-art algorithms.

SHARP is composed of many indispensable components. To evaluate the contribution of RP, we replaced RP with random selection, so that the Johnson-Lindenstrauss lemma (Johnson and Lindenstrauss 1984) is no longer valid. We observed the performance become severely worse by random selection, suggesting that the

268 theoretical background by RP is a major contributor towards the performance of SHARP. Also, SHARP
269 employed an ensemble clustering method to combine the results of several runs of RP. We found that the
270 ensemble strategy provides the robust performance in clustering (Supplemental Fig S9 and Supplemental
271 Methods). SHARP was robust to the ensemble size when the size is larger than 5 (Supplemental Fig S10,
272 Supplemental Fig S11 and Supplemental Methods). Moreover, SHARP's performance was not highly
273 affected by the size of the block when the size is larger than 1,000 cells (Supplemental Fig S12 and
274 Supplemental Methods). SHARP is also roughly insensitive to the degree of dimension reduction
275 (Supplemental Fig S13 and Supplemental Methods).
276
277 It should be noted that the superior performance SHARP is not due to that a single run of RP outperforms
278 traditional dimension reduction methods such as PCA. On the contrary, the performance of a single run of RP
279 is volatile (Supplemental Fig S9). Instead, the superior performance of SHARP is due to the following
280 reasons: (1) multiple runs of RP generated diversity and then they were combined with the weighted-ensemble
281 clustering approach for robust clustering and (2) RP's dimension reduction with the property of preserving
282 cell-to-cell distance in the lower-dimensional space. Our meta-clustering strategies were designed to
283 effectively handle tiny clusters generated during the data partition stage (Supplemental Fig S14 and
284 Supplemental Note 7).
285
286 To assess the robustness of SHARP, we performed various tests. (1) We ran 100 times of SHARP and reported
287 the mean +/- standard deviation of ARI and computational time for almost all scRNA-seq datasets in this
288 paper; (2) We tested SHARP on a variety of scRNA-seq datasets with different numbers of cells ranging from
289 several hundred to 10 millions; (3) We used SHARP to process scRNA-seq with various additional dropout
290 rates. In these series of tests, SHARP demonstrated robustness across different cases.
291

Randomness has been widely used in multiple different domains like scalable dimensionality reduction and matrix decomposition approximation (Halko et al. 2011). The randomized method like the Gaussian random matrix, which also belongs to one kind of random projections, was used for low-rank matrix decomposition approximation (Halko et al. 2011), whereas we used random projection for dimension reduction and clustering in large-scale scRNA-seq data. Besides, the random matrix we used for SHARP was highly sparse whereas the one used in (Halko et al. 2011) was not.

Clustering involves extensive use of computational resources in calculating distances and/or dimension reduction. We demonstrated that SHARP is scalable to processing even 10 million cells. Currently, SHARP is not designed to detect rare cell populations, which could be a further application of RP. Besides clustering, the property of RP to preserve cell-to-cell distance in the reduced dimension will be useful for other applications for scRNA-seq data.

METHODS

The framework of SHARP. SHARP accepts gene expression data arranged in a matrix $\boldsymbol{M} \in \mathcal{R}^{D\times N}$, where each of the D rows corresponds to a gene (or transcript) and each of the N columns corresponds to a single cell. The type of input data can be either fragments/reads per kilo base per million mapped reads (FPKM/RPKM), counts per million mapped reads (CPM), transcripts per million (TPM) or unique molecule identifiers (UMI). For consistency, FPKM/RPKM values are converted into TPM values and UMI values are converted into CPM values.

Data partition. For a large-scale dataset SHARP performs data partition using a divide-and-conquer strategy. SHARP divides scRNA-seq data $\boldsymbol{M} \in \mathcal{R}^{D\times N}$ into B blocks, where each block may contain different numbers

of cells (i.e., $N_1, \dots, N_B$, where $\sum_{i=1}^{B} N_i = N$). To avoid bias during data partition, we randomly permuted the original single-cell data before partitioning. In practice, SHARP roughly equally divides $\boldsymbol{M}$ and allows users to assign the base number of single cells in each block (e.g., n). In this case, $B = \lceil N/n \rceil$, where $\lceil x \rceil$ is the minimum integer no less than x. The numbers of single cells $\{N_b\}_{b=1}^{B}$ in each block are as follows:

1. If $B = 1$, $N_b = N$, where $b = B = 1$;
2. If $B = 2$, $N_b = \begin{cases} \left\lfloor \frac{N}{2} \right\rfloor, \text{where } b = \{B-1\} \\ \left\lceil \frac{N}{2} \right\rceil, \text{where } b = \{B\} \end{cases}$;
3. If $B \geq 3$, $N_b = \begin{cases} n, & \text{where } b = \{1, \dots, B-2\} \\ \left\lfloor \frac{N-n(B-2)}{2} \right\rfloor, & \text{where } b = \{B-1\} \\ \left\lceil \frac{N-n(B-2)}{2} \right\rceil, & \text{where } b = \{B\} \end{cases}$

This enables SHARP to maximize the usage of local computational resources and avoid memory overflow while minimizing the negative impact from imbalanced numbers of data for each block.

Random Projection (RP). RP is a group of simple yet powerful dimension-reduction technique. It is based on the Johnson-Lindenstrauss lemma (Johnson and Lindenstrauss 1984) (Supplemental Methods). Specifically, the original *D*-dimensional data is projected onto a *d*-dimensional subspace, using a random matrix whose column are unit length, i.e.,

$$\boldsymbol{P} = \frac{1}{\sqrt{\boldsymbol{d}}} \boldsymbol{R}\boldsymbol{M} \in \mathcal{R}^{d \times N}, \boldsymbol{M} \in \mathcal{R}^{D \times N}, \boldsymbol{R} \in \mathcal{R}^{d \times D}$$

As long as the elements of **R** conforms to any distributions with zero mean and unit variance, **R** gives a mapping that satisfies the Johnson-Lindenstrauss lemma.

Choice of random matrix R. To reduce the computational complexity, we adopted a very sparse RP proposed in (Li et al. 2006) where the elements of **R** (i.e., $r_{i,j}$) is defined as:

338 $$r_{i,j} = \sqrt{s}\begin{cases} 1 & \text{with probability } \frac{1}{2s}, \\ 0 & \text{with probability } 1-\frac{1}{s}, \\ -1 & \text{with probability } \frac{1}{2s}, \end{cases} \quad \text{where } i = \{1, \dots, d\}, j = \{1, \dots, D\}.$$

339 As suggested in (Li et al. 2006), we selected $s = \sqrt{D}$.

340

341 **Choice of the subspace dimension *d*.** To balance between maintaining robust performance and yielding a
342 solution as parsimonious as possible, we selected $d = \log_2(N)/\epsilon^2$, where $\epsilon \in (0,1]$ as suggested by Johnson-
343 Lindenstrauss lemma.

344

345 **Ensemble RP.** After RP, pairwise correlation coefficients between each pair of single cells were calculated
346 using the dimension-reduced feature matrix. An agglomerative hierarchical clustering (hclust) with the
347 "ward.D" (Ward Jr 1963) method was used to cluster the correlation-based distance matrix. We first applied
348 RP *K* times to obtain *K* RP-based dimension-reduced feature matrices and then further *K* distance matrices.
349 Each of the K matrices was clustered by a "ward.D"-based hclust. As a result, *K* different clustering results
350 were obtained, each from a RP-based distance matrix, which would be combined by a weighted-based meta-
351 clustering (wMetaC) algorithm (Ren et al. 2017) detailed in the next step.

352

353 **Weighted-based meta-clustering (wMetaC).** Compared to the traditional cluster-based similarity
354 partitioning algorithm (CSPA) (Strehl and Ghosh 2002) which treat each instance and each cluster equally
355 important, wMetaC assigns different weights to different instances (or instance pairs) and different clusters
356 to improve the clustering performance. wMetaC includes four steps: (1) calculating cell weights; (2)
357 calculating weighted cluster-to-cluster pairwise similarity; (3) clustering on weighted cluster-based similarity
358 matrix and (4) determining final results by a voting scheme. Note that wMetaC was applied to each block of
359 single cells. The flowchart of the wMeaC ensemble clustering method is shown in Supplemental Fig S15.
360 Specifically, for calculating cell weights, similar to the first several steps in CSPA, we first converted the

individual RP-based clustering results into a co-location similarity matrix **S**, whose element $s_{i,j}$ represents the similarity between the *i*-th and *j*-th single cells. Then, based on the idea that the weight for each pair of single cells is determined by the degree of consistency of the co-location clustering results of these two single cells, we converted the similarity matrix **S** to the weight matrix **W** according to the following equation:

$$w_{i,j} = s_{i,j}(1 - s_{i,j}),$$

where $w_{i,j}$ is the element in the *i*-th row and the *j*-th column of W. It is easy to see that when $s_{i,j} = 1$ (i.e., the *i*-th cell and the *j*-th cell are with 100% probability in the same cluster) or $s_{i,j} = 0$ (i.e., the *i*-th cell and the *j*-th cell are with 0% probability in the same cluster), $w_{i,j}$ reaches the minimum at 0; when $s_{i,j} = 0.5$ (i.e., the co-location probability of the *i*-th cell and the *j*-th cell in the same cluster is 0.5 whereas the probability of them in different clusters is also 0.5, which means this is the most difficult-to-cluster case), $w_{i,j}$ reaches the maximum at 0.25. In other words, zero weight is assigned to those most "easy-to-cluster" pairs of single cells and the highest cell-to-cell weight is assigned for the most "difficult-to-cluster" pairs. Then, a weight associated with each cell was calculated as the accumulation of all the cell-to-cell weights related with the corresponding cell. To calculate the weighted cluster-to-cluster similarity, we first noted that the size of the similarity matrix is $|C| \times |C|$, where *C* is the union set of all the clusters obtained in each individual RP-based clustering results in the previous step, and $|\cdot|$ is the cardinality of a set. Then, for any two clusters, their similarity is determined by the sum of weights of their overlapped elements (i.e., cells) divided by that of their combined ones. Specifically, given two clusters C_u and C_v, the cluster-to-cluster similarity in wMetaC is defined as:

$$S_{wMetaC} = \frac{\sum_{t \in C_u \cap C_v} \sum_{j=1}^{N} w_{t,j} + \delta}{\sum_{t \in C_u \cup C_v} \sum_{j=1}^{N} w_{t,j} + \delta},$$

where $w_{t,j}$ is the co-location weight for the *t*-th cell and *j*-th cell derived above, *N* is the number of cells, and δ is a very small positive number (by default, we used $\delta = 0.01$) to avoid the denominator to be 0. We can treat $\sum_{j=1}^{N} w_{t,j}$ as the overall co-location weights for the *t*-th cell because it sums up all the possible co-location weights between the *t*-th cell and all cells. Thus, in the equation, the numerator represents the sum of

385 the co-location weights of the cells which are found in both Cluster C_u and Cluster C_v, whereas the
386 denominator represents the sum of the co-location weights of the cells which are found in either Cluster C_u
387 or Cluster C_v.
388
389 Then, in the third step (i.e., meta-clustering), we used a hierarchical clustering with "ward.D" to cluster the
390 obtained similarity matrix. After clustering, we understood which cluster in the 1-st RP-based clustering
391 corresponds to which cluster(s) in the 2-nd, 3-rd,…, *K*-th RP-based clustering. Then, in the final step, we
392 reorganized the *K* RP-based clustering results according to the result in the third step, and then we used a
393 voting scheme (see below) to determine the final clustering results. These procedures were repeated for each
394 of the *B* blocks.
395
396 **Voting scheme.** Given *K* runs of RPs, we obtained *K* different individual clustering results by using
397 hierarchical clustering on each of the *K* RP-based dimension-reduced feature matrices. Suppose that the *k*-th
398 ($k \in \{1, \dots, K\}$) individual clustering results in $|C^k|$ clusters, i.e., $\{C_1^k, C_2^k, \dots, C_{|C_k|}^k\}$. Then, by using the first
399 three steps of wMetaC, we obtained a weighted-based cluster-wise similarity matrix and then we used
400 hierarchical clustering again for this meta-clustering problem. After meta-clustering, assume that the number
401 of clusters is $|G|$, i.e., $\{G_1, \dots, G_g, \dots, G_{|G|}\}$, where $|G| \leq \sum_{k=1}^{K} |C^k|$. Thus, G_g corresponds to one or more
402 clusters from the individual clustering set. Finally, a single cell was assigned to the meta-cluster to which it
403 belongs with the highest ratio. Ties were broken randomly. For example, given $K = 5$, suppose for a single
404 cell, its 5 RP-based individual clustering results are $(C_3^1, C_1^2, C_6^3, C_3^4, C_2^5)$, i.e., it belongs to the 3-rd cluster in
405 the 1-st RP clustering results, the 1-st cluster in the 2-nd RP, the 6-th cluster in the 3-rd RP, the 3-rd cluster
406 in the 4-th RP and the 2-nd cluster in the 5-th RP (Note that the individual clusters are not necessarily
407 consistent with each other and that's why meta-clustering method like wMetaC is required). After meta-
408 clustering, suppose that C_3^1, C_6^3 and C_2^5 belong to the same meta-cluster G_3, while C_1^2 and C_3^4 belong to G_1 and
409 G_2, respectively. Because this single cell is predicted to belong to Cluster G_3 with the highest ratio (i.e., 3/5),

410 the final predicted cluster of this cell is Cluster G_3.

411

412 **Similarity-based meta-clustering (sMetaC).** To integrate the clustering results of the *B* blocks obtained by
413 wMetaC, we proposed a similarity-based meta-clustering (sMetaC) approach which is similar to wMetaC.
414 The major differences between wMetaC and sMetaC: (1) the cluster-to-cluster pairwise similarity of the
415 former is calculated based on co-location weights of single cells in each cluster, whereas that of the latter is
416 calculated based on the mean of the cell-to-cell correlation coefficients; (2) the individual clustering results
417 of the former actually correspond to the same block of single cells but in different lower-dimensional space,
418 whereas those of the latter correspond to different blocks of single cells; (3) the former requires a voting
419 scheme to integrate K individual clustering results, whereas the latter does not and it just needs to reorganize
420 the clusters to make clusters consistent across blocks.

421

422 The cluster-to-cluster pairwise similarity of wMetaC is calculated based on co-location weights of single cells
423 in each cluster, whereas that of the sMetaC is calculated based on the Pearson correlation coefficient of the
424 mean cluster-wise feature vectors after ensemble random projection. Specifically, in sMetaC, given two
425 clusters G^u and G^v which were obtained by wMetaC (note that these two clusters may or may not belong to
426 the same block of single cells), we calculated the similarity between these two clusters in sMetaC as follows

427 $S_{sMetaC}(G^u, G^v) = \mathbf{cor}\left(\frac{1}{|G^u|}\sum_{i\in G^u}\boldsymbol{p}_i^u, \frac{1}{|G^v|}\sum_{j\in G^v}\boldsymbol{p}_j^v\right)$,

428 where $\mathbf{cor}(\cdot,\cdot)$ is the Pearson correlation coefficient of two vectors, $\boldsymbol{p}_i^u$ and $\boldsymbol{p}_j^v$ are the dimension-reduced
429 feature vectors after ensemble random projection for the *i*-th cell in Cluster G^u and the *j*-th cell in Cluster
430 G^v, and $|G^u|$ and $|G^v|$ are the numbers of cells in Cluster G^u and Cluster G^v, respectively.

431

432 **Determining the optimal number of clusters.** SHARP determines the optimal number of clusters by using
433 three criteria which are based on internal evaluations of the clustering results (Supplemental Methods).

434

435 **Time complexity analysis. S**HARP includes 4 steps for clustering: (1) data partition; (2) RP; (3) wMetaC;
436 (4) sMetaC. For a scRNA-seq data matrix $\boldsymbol{M} \in \mathcal{R}^{D\times N}$, SHARP first divides the data into B blocks, the b-th
437 block with N_b single cells. According to the "data partition" section, $N_b \leq n$, where n is a fixed user-defined
438 parameter enabling that one application of RP-based clustering runs sufficiently fast. Our analysis
439 (Supplemental Fig S12) shows that $n = 1500$ or 2000 is a good balance between performance and speed in
440 our case. Then, for each block, one run of RP requires time complexity of $O(nd\sqrt{D})$ (Johnson and
441 Lindenstrauss 1984), where $d = \lceil \log(N)/\epsilon^2 \rceil \ll D$. Note here, d is calculated based on N rather than n for
442 dimension-reduction consistency across blocks. Practically, in the 13 reported scRNA-seq datasets, the
443 dimension can be reduced by 42~238 times (i.e., *D/d*), depending on the number of single cells and the number
444 of genes. SHARP requires several (i.e., K) runs of RPs (with complexity of $O(Knd\sqrt{D})$). Subsequently,
445 SHARP uses a hierarchical clustering (hclust) with "ward.D" for each of the K RPs, thus with overall time
446 complexity of $O(K(nd\sqrt{D} + n^2))$ (Note that the time complexity of hclust in R package is $O(n^2)$ (Murtagh
447 and Legendre 2014)). Later, wMetaC, essentially a hclust for the individual predicted clusters (also without
448 loss of generality, suppose the number of clusters for each RP-based clustering is equally C_1, where $C_1 \ll n$),
449 was applied to each block (in total, time complexity of $O(K(nd\sqrt{D} + n^2) + (KC_1)^2)$). Finally, SHARP
450 integrated the results of all blocks by proposing a method called sMetaC, whose time complexity is similar to
451 wMetaC except the number of instances is different (similarly, we can suppose the number of clusters in each
452 block is equally C_2, where $C_2 \ll n$). In this case, the total time complexity is $O(B[K(nd\sqrt{D} + n^2) +$
453 $(KC_1)^2] + (BC_2)^2)$. Practically, K, B, C_1 and C_2 are very small, thus the time complexity of SHARP can be
454 written as $O(KN(d\sqrt{D} + n))$. Because n is fixed across different datasets, $d = \lceil \log(N)/\epsilon^2 \rceil$ and D is usually
455 larger than 10,000, thus $d\sqrt{D} > n$ and therefore the time complexity of SHARP is essentially
456 $O(N\log(N)\sqrt{D})$.
457 On the other hand, among the compared state-of-the-art methods, tSNE plus k-means is arguably the fastest.
458 Theoretically, tSNE requires $O(DN\log(N))$ for dimension reduction to 2D or 3D space (van der Maaten

459 2014). For tSNE plus k-means for clustering, the time complexity is $O(DN\log(N) + 2Nki)$, where k is the
460 number of clusters and i is the number of iterations. Thus, the total time complexity for tSNE + kMeans is
461 $O(DN\log(N))$.
462 All the tests except for the 1.3-million-cell dataset were performed using a single core on an Intel Xeon CPU
463 E5-2699 v4 @ 2.20GHz system with 500GB memory. To run 1.3 million cells, we used 16 cores on the same
464 system.
465
466 **Visualization.** For visualization, SHARP uses a weighted combination of the dimension-reduced feature
467 matrix and the cell-to-cluster matrix derived from the clustering results. The former matrix is obtained by
468 three steps: 1) applying *K* runs of RPs for each block of the large-scale scRNA-seq data; 2) combining these
469 block-wise matrices to obtain *K* RP-based dimension-reduced matrices and 3) averaging these *K* matrices into
470 one ensemble matrix. For the latter matrix, we constructed a $N \times pC$ matrix, where *N* is the number of single
471 cells and *pC* is the predicted number of clusters. If the *i*-the single cell is predicted to be in the *j*-th cluster,
472 then the element of the *i*-th row and *j*-th column is 1, otherwise 0. Subsequently, these two matrices were
473 combined with different weights to formulate the visualization matrix, which is the input matrix of tSNE for
474 visualization.
475
476 First, both the dimension-reduced feature matrix $\boldsymbol{A} \in \mathcal{R}^{N\times d}$ (where *N* is the number of cells and *d* is the
477 number of dimensions after dimension reduction) and the cell-to-cluster matrix $\boldsymbol{B} \in \mathcal{R}^{N\times pC}$ (where *pC* is the
478 predicted number of clusters) are centered and scaled along each dimension across cells, and we notate the
479 results as $\overline{\boldsymbol{A}}$ and $\overline{\boldsymbol{B}}$. Then, these two matrices are combined as follows: $\overline{\boldsymbol{V}} = [w\overline{\boldsymbol{A}}, \overline{\boldsymbol{B}}]$, where $\overline{\boldsymbol{V}}$ is the input
480 matrix to tSNE for visualization and $w > 0$ is the weight ratio of the dimension-reduction feature matrix over
481 the cell-to-cluster matrix. If $0<w<1$, more weight will be given to cell-to-cluster matrix, suggesting that the
482 clustering results are believed to be better for visualization; if $w>1$, more weight will be given to the
483 dimension-reduced feature matrix, which indicates that the data in dimension-reduced space can be more

484 suitable for visualization. While it is possible to employ some algorithms to optimize *w*, we adopted an
485 empirical value (i.e., $w = 2$) by default, which is robust for better visualization across different scRNA-seq
486 datasets. For flexibility, we also provided an extra option to allow users to define by their own *w*.
487
488 Based on the clustering results, SHARP can further detect genes for each cluster. We adopted a method similar
489 to SC3 except three points: (1) Besides p-value and area under receiver operating curve (AUROC), SHARP
490 uses two more criteria to select marker genes, namely cluster-mean fold change (FC) and expression sparsity
491 (i.e., the percentage of expressions across all cells); (2) SHARP uses an adaptive threshold instead of a hard-
492 threshold (i.e., p-value < 0.01 and AUROC > 0.85); (3) SHARP uses a parallelization way to calculate all of
493 the criteria mentioned above.
494
495 **Simulated data.** In this study, we have generated three simulated scRNA-seq data from the Tabula Muris
496 (Schaum et al. 2018). Specifically, we only selected the data from the microfluidic droplet-based method and
497 three simulated data were generated, i.e., mdata3, mdata6 and mdata8. mdata3 was generated by mixing
498 scRNA-seq data from three organs, including heart and aorta, thymus and liver; mdata6 was generated from
499 six organs, including heart and aorta, thymus, liver, bladder, kidney and tongue; and mdata8 was generated
500 from eight organs, including, heart and aorta, thymus, liver, bladder, kidney, tongue, spleen and trachea. Note
501 that only those filtered data by the original paper were used. The total numbers of single cells for mdata3,
502 mdata6 and mdata8 are 3,898, 16,717 and 37,538, respectively.
503
504 **Adding dropouts**. To further demonstrate the robustness of SHARP against scRNA-seq dropout events, we
505 artificially added dropouts to a benchmarking dataset (e.g., Montoro_small). Specifically, we randomly
506 selected a percentage (e.g., 1%, 5%, 10%, 15% and 20%) of non-zero expressions from the Montoro_small
507 dataset and then set them to be 0. In other words, the dropout percentage here refers to the added dropout
508 percentage.

Public datasets used in this study. Single cell RNA-seq data were obtained from the GEO accession numbers provided by their respective original publications. For the 1.3 million single-cell dataset, we downloaded from the 10x Genomics website https://support.10xgenomics.com/single-cell-gene-expression/datasets/1.3.0/1M_neurons. The scRNA-seq data from the Tabula Muris was downloaded from the website https://figshare.com/projects/Tabula_Muris_Transcriptomic_characterization_of_20_organs_and_tissues_from_Mus_musculus_at_single_cell_resolution/27733.

Software Availability

The source code for SHARP has been submitted to Github (https://github.com) under the project name SHARP (https://github.com/shibiaowan/SHARP).

DISCLOSURE DECLARATION

The authors declare no competing financial interests.

ACKNOWLEDGEMENTS

This work is supported by NIH grant (R01 DK106027) and the Novo Nordisk Foundation grant number NNF17CC0027852.

AUTHORS' CONTRIBUTIONS

SW and KJW conceived and designed the study. SW developed the SHARP algorithm, performed the experiments and analyzed the data. SW implemented the SHARP package. SW, JK and KJW participated in writing the paper. The manuscript was approved by all authors.

534

535 **REFERENCES**

536 10x Genomics. 2017. Transcriptional Profiling of 1.3 Million Brain Cells with the Chromium Single Cell 3'
537 Solution.
538 Baron M, Veres A, Wolock SL, Faust AL, Gaujoux R, Vetere A, Ryu JH, Wagner BK, Shen-Orr SS, Klein
539 AM et al. 2016. A Single-Cell Transcriptomic Map of the Human and Mouse Pancreas Reveals Inter-
540 and Intra-cell Population Structure. *Cell Syst* **3**: 346-360.
541 Becht E, Dutertre C-A, Kwok IW, Ng LG, Ginhoux F, Newell EW. 2018. Evaluation of UMAP as an
542 alternative to t-SNE for single-cell data. *bioRxiv*: 298430.
543 Bertoni A, Valentini G. 2006. Ensembles based on random projections to improve the accuracy of clustering
544 algorithms. *Lect Notes Comput Sc* **3931**: 31-37.
545 Bingham E, Mannila H. 2001. Random projection in dimensionality reduction: applications to image and text
546 data. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery*
547 *and data mining*, pp. 245-250. ACM.
548 Butler A, Hoffman P, Smibert P, Papalexi E, Satija R. 2018. Integrating single-cell transcriptomic data across
549 different conditions, technologies, and species. *Nat Biotechnol* **36**: 411-420.
550 Darmanis S, Sloan SA, Zhang Y, Enge M, Caneda C, Shuer LM, Gephart MGH, Barres BA, Quake SR. 2015.
551 A survey of human brain transcriptome diversity at the single cell level. *P Natl Acad Sci USA* **112**:
552 7285-7290.
553 Enge M, Arda E, Mignardi M, Beausang J, Bottino R, Kim SK, Quake SR. 2017. Single-Cell Analysis of
554 Human Pancreas Reveals Transcriptional Signatures of Aging and Somatic Mutation Patterns. *Cell*
555 **171**: 321-330.
556 Fern XZ, Brodley CE. 2003. Random projection for high dimensional data clustering: A cluster ensemble
557 approach. In *Proceedings of the 20th international conference on machine learning (ICML-03)*, pp.
558 186-193.
559 Frankl P, Maehara H. 1988. The Johnson-Lindenstrauss Lemma and the Sphericity of Some Graphs. *J Comb*
560 *Theory B* **44**: 355-362.
561 Goolam M, Scialdone A, Graham SJL, Macaulay IC, Jedrusik A, Hupalowska A, Voet T, Marioni JC,
562 Zernicka-Goetz M. 2016. Heterogeneity in Oct4 and Sox2 Targets Biases Cell Fate in 4-Cell Mouse
563 Embryos EDITORIAL COMMENT. *Obstet Gynecol Surv* **71**: 411-412.
564 Guemez-Gamboa A, Coufal NG, Gleeson JG. 2014. Primary Cilia in the Developing and Mature Brain.
565 *Neuron* **82**: 511-521.
566 Halko N, Martinsson PG, Tropp JA. 2011. Finding Structure with Randomness: Probabilistic Algorithms for
567 Constructing Approximate Matrix Decompositions. *Siam Rev* **53**: 217-288.
568 Han X, Wang R, Zhou Y, Fei L, Sun H, Lai S, Saadatpour A, Zhou Z, Chen H, Ye F et al. 2018. Mapping the
569 Mouse Cell Atlas by Microwell-Seq. *Cell* **172**: 1091-1107 e1017.
570 Hie B, Bryson B, Berger B. 2019a. Efficient integration of heterogeneous single-cell transcriptomes using
571 Scanorama. *Nat Biotechnol* **37**: 685-691.
572 Hie B, Cho H, DeMeo B, Bryson B, Berger B. 2019b. Geometric Sketching Compactly Summarizes the
573 Single-Cell Transcriptomic Landscape. *Cell Syst* **8**: 483-+.
574 Hubert L, Arabie P. 1985. Comparing Partitions. *Journal of Classification* **2**: 193-218.
575 Hyvarinen A, Oja E. 2000. Independent component analysis: algorithms and applications. *Neural Networks*
576 **13**: 411-430.
577 Iacono G, Mereu E, Guillaumet-Adkins A, Corominas R, Cusco I, Rodriguez-Esteban G, Gut M, Perez-Jurado
578 LA, Gut I, Heyn H. 2018. bigSCale: an analytical framework for big-scale single-cell data. *Genome*
579 *Res* **28**: 878-890.
580 Johnson WB, Lindenstrauss J. 1984. Extensions of Lipschitz mappings into a Hilbert space. *Contemporary*
581 *mathematics* **26**: 1.

582 Joliffe IT, Morgan BJ. 1992. Principal component analysis and exploratory factor analysis. *Stat Methods Med*
583 *Res* **1**: 69-95.
584 Kiselev VY, Kirschner K, Schaub MT, Andrews T, Yiu A, Chandra T, Natarajan KN, Reik W, Barahona M,
585 Green AR et al. 2017. SC3: consensus clustering of single-cell RNA-seq data. *Nat Methods* **14**: 483-
586 486.
587 Klein AM, Mazutis L, Akartuna I, Tallapragada N, Veres A, Li V, Peshkin L, Weitz DA, Kirschner MW.
588 2015. Droplet Barcoding for Single-Cell Transcriptomics Applied to Embryonic Stem Cells. *Cell* **161**:
589 1187-1201.
590 Kolodziejczyk AA, Kim JK, Tsang JCH, Ilicic T, Henriksson J, Natarajan KN, Tuck AC, Gao XF, Buhler M,
591 Liu PT et al. 2015. Single Cell RNA-Sequencing of Pluripotent States Unlocks Modular
592 Transcriptional Variation. *Cell Stem Cell* **17**: 471-485.
593 Levine JH, Simonds EF, Bendall SC, Davis KL, Amir EAD, Tadmor MD, Litvin O, Fienberg HG, Jager A,
594 Zunder ER et al. 2015. Data-Driven Phenotypic Dissection of AML Reveals Progenitor-like Cells
595 that Correlate with Prognosis. *Cell* **162**: 184-197.
596 Li P, Hastie TJ, Church KW. 2006. Very sparse random projections. In *Proceedings of the 12th ACM SIGKDD*
597 *international conference on Knowledge discovery and data mining*, pp. 287-296. ACM.
598 Lin PJ, Troup M, Ho JWK. 2017. CIDR: Ultrafast and accurate clustering through imputation for single-cell
599 RNA-seq data. *Genome Biol* **18**.
600 Macosko EZ, Basu A, Satija R, Nemesh J, Shekhar K, Goldman M, Tirosh I, Bialas AR, Kamitaki N,
601 Martersteck EM et al. 2015. Highly Parallel Genome-wide Expression Profiling of Individual Cells
602 Using Nanoliter Droplets. *Cell* **161**: 1202-1214.
603 Montoro DT, Haber AL, Biton M, Vinarsky V, Lin B, Birket SE, Yuan F, Chen SJ, Leung HM, Villoria J et
604 al. 2018. A revised airway epithelial hierarchy includes CFTR-expressing ionocytes. *Nature* **560**:
605 319-324.
606 Murtagh F, Legendre P. 2014. Ward's hierarchical agglomerative clustering method: which algorithms
607 implement Ward's criterion? *Journal of classification* **31**: 274-295.
608 Park J, Shrestha R, Qiu CX, Kondo A, Huang SZ, Werth M, Li MY, Barasch J, Susztak K. 2018. Single-cell
609 transcriptomics of the mouse kidney reveals potential cellular targets of kidney disease. *Science* **360**:
610 758-763.
611 Potter SS. 2018. Single-cell RNA sequencing for the study of development, physiology and disease. *Nat Rev*
612 *Nephrol* **14**: 479-492.
613 Ren YZ, Domeniconi C, Zhang GJ, Yu GX. 2017. Weighted-object ensemble clustering: methods and
614 analysis. *Knowl Inf Syst* **51**: 661-689.
615 Schaum N Karkanias J Neff NF May AP Quake SR Wyss-Coray T Darmanis S Batson J Botvinnik O Chen
616 MB et al. 2018. Single-cell transcriptomics of 20 mouse organs creates a Tabula Muris. *Nature* **562**:
617 367-372.
618 Strehl A, Ghosh J. 2002. Cluster ensembles---a knowledge reuse framework for combining multiple partitions.
619 *J Mach Learn Res* **3**: 583-617.
620 van der Maaten L. 2014. Accelerating t-SNE using Tree-Based Algorithms. *J Mach Learn Res* **15**: 3221-3245.
621 Wang B, Zhu JJ, Pierson E, Ramazzotti D, Batzoglou S. 2017. Visualization and analysis of single-cell RNA-
622 seq data by kernel-based similarity learning. *Nat Methods* **14**: 414-416.
623 Wang YJ, Schug J, Won KJ, Liu CY, Naji A, Avrahami D, Golson ML, Kaestner KH. 2016. Single-Cell
624 Transcriptomics of the Human Endocrine Pancreas. *Diabetes* **65**: 3028-3038.
625 Ward Jr JH. 1963. Hierarchical grouping to optimize an objective function. *J Am Stat Assoc* **58**:
626 236-244.
627