Visualization and analysis of single-cell RNA-seq data by kernel-based similarity learning

Bo Wang[1], Junjie Zhu[2], Emma Pierson[1], Daniele Ramazzotti[1,3] & Serafim Batzoglou[1]

We present single-cell interpretation via multikernel learning (SIMLR), an analytic framework and software which learns a similarity measure from single-cell RNA-seq data in order to perform dimension reduction, clustering and visualization. On seven published data sets, we benchmark SIMLR against state-of-the-art methods. We show that SIMLR is scalable and greatly enhances clustering performance while improving the visualization and interpretability of single-cell sequencing data.

Single-cell RNA sequencing (scRNA-seq) has revealed previously unknown heterogeneity and functional diversity among cell populations[1]. Recent studies demonstrate that *de novo* cell type discovery of functionally distinct cell subpopulations is possible via the unbiased analysis of scRNA-seq data[2–4]. However, many of these analyses employ computational methods that were developed to analyze traditional bulk RNA-seq data, in which gene expression measurements are averaged over a population of cells. These methods fail to adequately address underlying challenges such as outlier cell populations, transcript amplification noise and dropout events, where expression measurements of zero occur on account of random sampling of transcripts[5]. Recent platforms such as DropSeq[6] and GemCode[7] have enabled a dramatic increase in throughput to thousands of cells. These platforms, however, typically produce sparse data sets for which 95% of measurements are zeros. A key problem with unsupervised methods for dimension reduction, clustering and visualization of these data is that they usually rely on similarity metrics which may not generalize across platforms and biological experiments. To address these problems, we introduce SIMLR, a framework that learns an appropriate cell-to-cell similarity metric from the input single-cell data (see Online Methods, **Fig. 1**, **Supplementary Software**, and https://github.com/BatzoglouLabSU/SIMLR).

SIMLR offers three main advantages over previous methods. First, it learns a distance metric that best fits the structure of the data by combining multiple kernels[8] (see Online Methods). The diverse statistical characteristics of single-cell data do not easily fit specific statistical assumptions made by standard dimension reduction or clustering algorithms. Multiple kernels have been shown to correspond to different informative representations of the data and often are more flexible than a single kernel[9]. Second, SIMLR addresses the challenge of high levels of dropout events by employing a rank constraint in the learned cell-to-cell similarity and graph diffusion[10]. The rank constraint enforces block structures in the similarity matrix, and the graph diffusion improves weak similarity measures (see Online Methods). Third, the similarities learned by SIMLR can be efficiently adapted into multiple downstream steps, such as prioritizing enes by ranking their concordance with the similarity and creating low-dimensional representations of cells by transforming the input into a stochastic neighbor embedding framework (see Online Methods).

In comparing SIMLR with conventional methods, we first show that SIMLR learns a similarity metric which outperforms standard similarity measures by analyzing four published single-cell data sets[2–4,11] spanning a variety of cell types (see Online Methods and **Supplementary Table 1**). Cell types in each data set were known a priori and were validated in the respective studies, providing a reliable gold standard. SIMLR takes the inputs of the raw gene expressions and the number of cell types—but no information about the individual cell identities (true labels)—and learned a matrix of similarities between cells. SIMLR's learned similarities corresponded better to the gold-standard labels than did standard similarity measures like correlation or Euclidian distance (**Fig. 2**). Because true labels of the Buettner data set[11] are well-studied cell-cycle states, we conducted additional analysis on this data set using SIMLR's gene prioritization feature (see Online Methods and **Fig. 1**). SIMLR implicated genes enriched in strongly connected gene–gene interaction networks and in highly significant biological processes related to cell cycle, translation and metabolic processes (**Supplementary Figs. 1** and **2**; **Supplementary Note 1**).

To analyze how well SIMLR applies to dimension reduction, we performed extensive comparisons of SIMLR with eight other dimension reduction methods, including principal component analysis (PCA)[12]; t-distributed stochastic neighbor embedding (t-SNE)[13]; and zero-inflated factor analysis (ZIFA)[5], which was shown to outperform many other model-based methods. Under six different performance metrics (see Online Methods and **Supplementary Note 2**), we found that SIMLR consistently outperforms the existing alternatives on the four data sets; the differences between SIMLR and the second-best method are often large (**Supplementary Note 3**; **Supplementary Tables 2** and **3**; **Supplementary Figs. 3–6**).

We also applied SIMLR's dimension reduction to visualize differences between cell populations (see Online Methods, **Fig. 3**,

[1]Department of Computer Science, Stanford University, Stanford, California, USA. [2]Department of Electrical Engineering, Stanford University, Stanford, California, USA. [3]Department of Pathology, Stanford University, Stanford, California, USA. Correspondence should be addressed to B.W. (bowang87@stanford.edu) or S.B. (serafim@cs.stanford.edu).

RECEIVED 9 JUNE 2016; ACCEPTED 2 FEBRUARY 2017; PUBLISHED ONLINE 6 MARCH 2017; DOI:10.1038/NMETH.4207

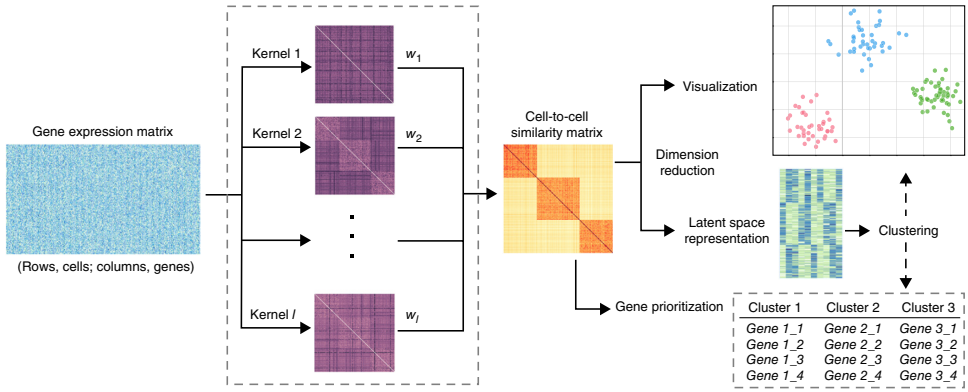

Figure 1 | Overview of SIMLR. Given a gene expression matrix as input, SIMLR learns proper weights for multiple kernels, which are different measures of cell-to-cell distances, and constructs a symmetric similarity matrix. SIMLR assumes that if C separable populations exist among the cells, then the similarity matrix should have an approximate block-diagonal structure with C blocks whereby cells within the same subpopulation are more similar. SIMLR uses the learned similarity to visualize the cells, reduce the dimension of the data, and cluster the cells into subgroups, as well as to prioritize the most variable genes that explain the differences across the populations. Solid arrows represent the information used compute the similarity matrix, and immediate applications of the similarity matrix. Dotted arrows indicate that clustering results can optionally be used in combination with visualization or gene prioritization. Each point in the 2D scatter plot represents a cell, and the corresponding colors represent the group ID assigned by clustering.

and **Supplementary Fig. 7**). In addition to producing 2D embeddings consistent with the true labels on each data set, SIMLR can identify subpopulation structures even when the selection of the rank-constraint parameter does not account for additional structured heterogeneity within each subpopulation. From the similarity structure learned by SIMLR in the Kolodziejczyk data set[4], we observed that each of the three validated groups could be further divided into subgroups (**Figs. 2** and **3**), consistent with the original study by Kolodziejczyk. While these subgroups were identified using preselected genes in the original study, SIMLR preserved these substructures without requiring gene preselection.

SIMLR can also be used to cluster cells, either by applying affinity propagation (AP)[14] directly to the learned similarity matrix, or by applying k-means clustering[15] in the latent space after applying SIMLR for dimension reduction (**Supplementary Note 4**). Clustering with AP demonstrates that the similarities learned by SIMLR significantly outperform Euclidean similarity and Pearson correlation. Further, k-means clustering with SIMLR consistently outperforms clustering methods recently used in single-cell studies[16] on the four data sets (**Supplementary Figs. 8** and **9**; **Supplementary Table 4**).

In addition to the four data sets with ground truth, we selected more challenging scenarios to assess SIMLR's performance. We analyzed a sparse peripheral blood mononuclear cell (PBMC) data set generated on the GemCode platform[7] with 2,700 cells and 95% zero expression counts. Using SIMLR with K-means clustering, we identified major cell types (B cells, CD4+ T cells, CD8+ T cells, natural killer cells, macrophages), including a rare megakaryocyte population of 12 cells (**Supplementary Note 5**, **Supplementary Figs. 10** and **11**, **Supplementary Table 5**). We also performed additional comparisons on a variety of simulated data sets with ground truth information to demonstrate SIMLR's superior performance under different modeling assumptions (**Supplementary Note 6** and **Supplementary Figs. 12–20**).

To demonstrate the scalability of SIMLR, we applied it to three published large-scale single-cell data sets with assigned cell types[6,7,17] (see Online Methods). We measured the consistency between our classification labels and the original cell type labels (**Supplementary Tables 6** and **7**). For the Zeisel data set[17], we adopted a two-level clustering approach, used in the original analysis, to show that SIMLR is able to dissect fine-grained heterogeneity in a hierarchical fashion. Moreover, we visualized the original classification labels from the studies to demonstrate SIMLR's effectiveness in revealing

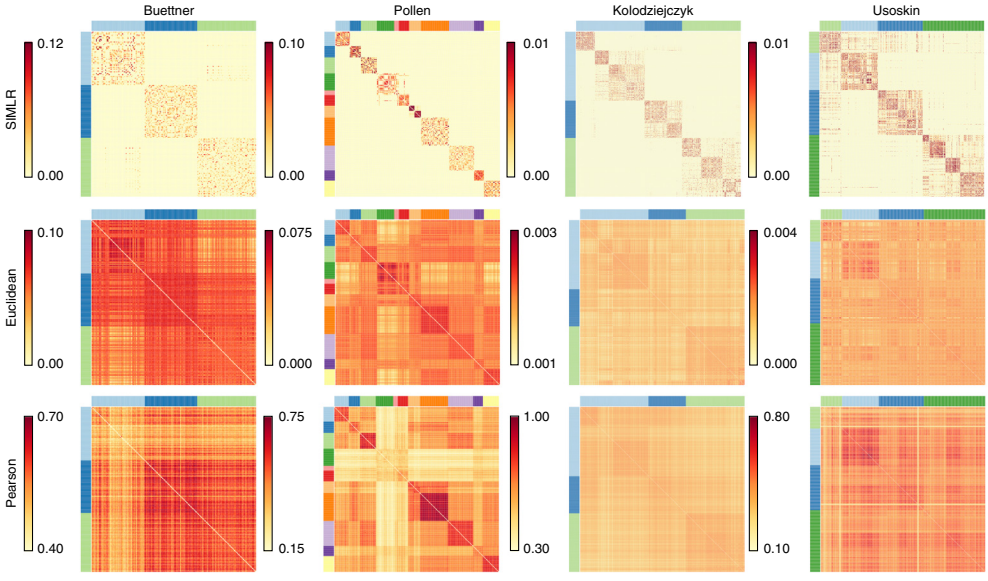

Figure 2 | Benchmark results on data sets with ground truth. Similarity matrices for similarities learned from the data by SIMLR, computed from Gaussian kernels applied to Euclidean distances and from pairwise Pearson correlations. The scales are in relative units from low to high similarity. Cells in the matrices are ordered by their true labels so that cells with the same label (indicated by the colored axes) are grouped together. SIMLR learned matrices with block structures in remarkable agreement with the true labels.

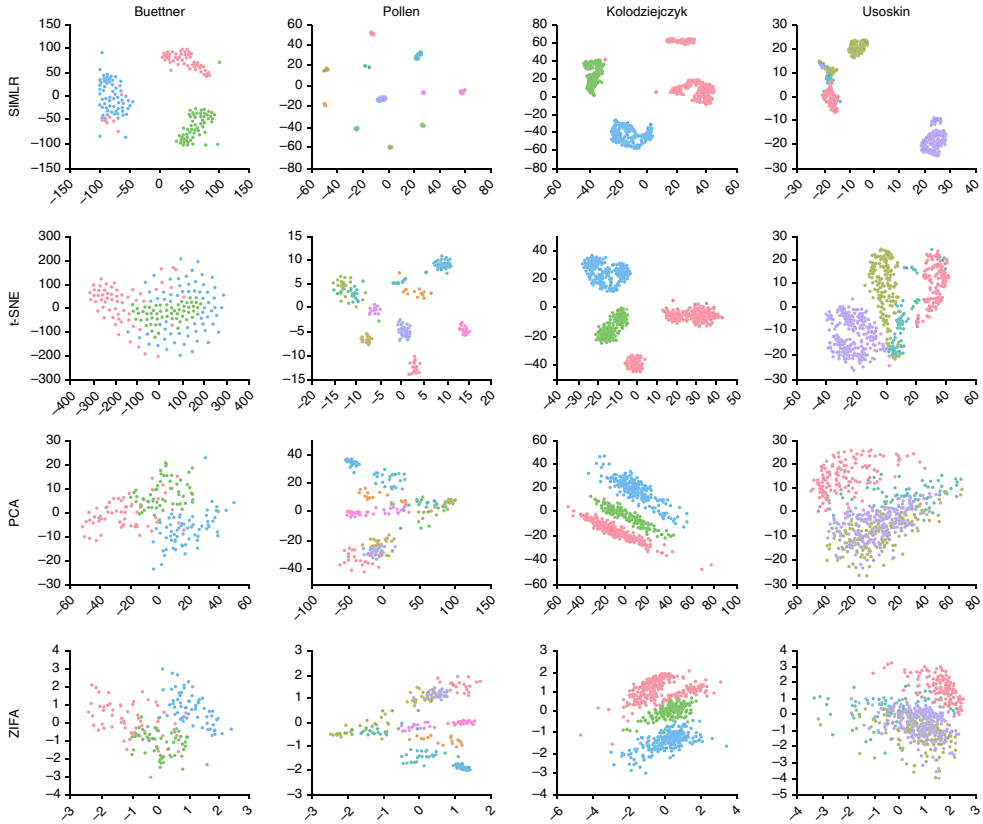

Figure 3 | Comparison of 2D visualization. The axes are in arbitrary units. Each point represents a cell and smaller distances between two cells represent greater similarity. None of the four methods used the true labels as inputs for dimension reduction, and the true label information was added in the form of distinct colors to validate the results.

meaningful structures in its 2D embedding (**Supplementary Fig. 21**). SIMLR can learn appropriate cell-to-cell distances that uncover similarity structures that would otherwise be concealed by noise or outlier effects in these large data sets.

In summary, SIMLR can determine which cells are similar in a way that generalizes across different single-cell data sets for dimension reduction, clustering and visualization applications. While SIMLR performs well on single-cell data sets that contain several clusters—a frequently used case where heterogeneity is defined by distinct cell lineages—we anticipate the multiple-kernel learning framework may still be beneficial for data that do not contain clear clusters, such as cell populations that contain cells spanning a continuum or a developmental pathway.

METHODS

Methods, including statements of data availability and any associated accession codes and references, are available in the online version of the paper.

Note: Any Supplementary Information and Source Data files are available in the online version of the paper.

ACKNOWLEDGMENTS

The authors would like to thank G.X. Zheng, J. Terry and T. Mikkelsen from 10x Genomics for providing access to the PBMC data as well as suggestions for the manuscript and the *in silico* experiments. E.P. acknowledges support from an NDSEG Fellowship and a Hertz Fellowship. J.Z. acknowledges support from a Stanford Graduate Fellowship.

AUTHOR CONTRIBUTIONS

B.W., J.Z., and S.B. conceived the study and planned experiments. B.W. designed the algorithm and implemented the software in MATLAB. D.R. and B.W. developed the software package in R. J.Z. and E.P. performed data analysis and implemented the simulation study. J.Z. and E.P. drafted the manuscript. B.W. and S.B. contributed to the manuscript. All authors read and approved the final manuscript.

COMPETING FINANCIAL INTERESTS

The authors declare competing financial interests: details are available in the online version of the paper.

1. Shapiro, E., Biezuner, T. & Linnarsson, S. *Nat. Rev. Genet.* **14**, 618–630 (2013).
2. Pollen, A.A. *et al. Nat. Biotechnol.* **32**, 1053–1058 (2014).
3. Usoskin, D. *et al. Nat. Neurosci.* **18**, 145–153 (2015).
4. Kolodziejczyk, A.A. *et al. Cell Stem Cell* **17**, 471–485 (2015).
5. Pierson, E. & Yau, C. *Genome Biol.* **16**, 241 (2015).
6. Macosko, E.Z. *et al. Cell* **161**, 1202–1214 (2015).
7. Zheng, G.X.Y. *et al. Nat. Commun.* **8**, 14049 (2017).
8. Bach, F.R., Lanckriet, G.R.G. & Jordan, M.I. In *Proc. 21st Int. Conf. Mach. Learn* (eds. Greiner, R. & Schuurmans, D.) 6 (ICML, 2004).
9. Gönen, M. & Alpaydin, E. *J. Mach. Learn. Res.* **12**, 2211–2268 (2011).
10. Wang, B. *et al. Nat. Methods* **11**, 333–337 (2014).
11. Buettner, F. *et al. Nat. Biotechnol.* **33**, 155–160 (2015).
12. Jolliffe, I. *Principal Component Analysis* (Wiley Online Library, 2002).
13. Van der Maaten, L. & Hinton, G. *J. Mach. Learn. Res.* **9**, 2579–2605 (2008).
14. Frey, B.J. & Dueck, D. *Science* **315**, 972–976 (2007).
15. Ding, C. & He, X. In *Proc. 21st Int. Conf. Mach. Learn* (eds. Greiner, R. & Schuurmans, D.) 225–232 (ICML, 2004).
16. Paul, F. *et al. Cell* **163**, 1663–1677 (2015).
17. Zeisel, A. *et al.* Title. *Science* **347**, 1138–1142 (2015).

ONLINE METHODS

The SIMLR framework. *Implementation and data availability.* SIMLR is freely available as both a MATLAB program and an R package (https://github.com/BatzoglouLabSU/SIMLR; https://bioconductor.org/packages/release/bioc/html/SIMLR.html). For *k*-means clustering, we used the *k*-means implementation for MATLAB and the built-in function in R. For stochastic neighbor embedding, we modified the source code of the t-SNE implementation in MATLAB and R. The processed four small-scale single-cell RNA-seq data sets are also available together with the software in both MATLAB and R formats. The three large-scale data sets are available from the corresponding author upon request. Given an $N \times M$ gene expression matrix X with N cells and M genes as an input, SIMLR solves for S, an $N \times N$ symmetric matrix that captures pairwise similarities of cells. In particular, S_{ij}, the (i,j)-th entry of S, represents the similarity between cell i and cell j. SIMLR assumes that if C separable populations exist among the N cells, then S should have an approximate block-diagonal structure with C blocks whereby cells have larger similarities to other cells within the same subpopulations. We express the general form of the distance between cell i and cell j as:

$$D\left(c_i, c_j\right) = 2 - 2\sum_l w_l K_l\left(c_i, c_j\right) \tag{1}$$

where each linear weight value w_l represents the importance of each individual kernel $K_l(\cdot,\cdot)$, which is a function of the expression of cell i and cell j (**Supplementary Note 7**).

SIMLR computes cell-to-cell similarities through the following optimization framework:

$$\begin{aligned} \underset{S,L,w}{\text{minimize}} & -\sum_{i,j,l} w_l K_l\left(c_i, c_j\right) S_{ij} + \beta \| S \|_F^2 + \\ & \gamma\, \mathbf{tr}(L^T (I_N - S) L) + \rho \sum_l w_l \log w_l \end{aligned} \tag{2}$$

$$\text{subject to } L^T L = I_C \sum_l w_l = 1,\ w_l \geq 0,\ \sum_j S_{ij} = 1, \text{ and } S_{ij} \geq 0$$

where I_N and I_C are $N \times N$ and $C \times C$ identity matrices, respectively, **tr**(.) represents the matrix trace, and β and γ are non-negative tuning parameters. $||S||_F$ denotes the Frobenius norm of S, and L denotes an auxiliary low-dimensional matrix enforcing the low rank constraint on S. The optimization problem involves solving for three variables: the similarity matrix S, the weight vector w, and an $N \times C$ rank-enforcing matrix L.

The intuition behind the first term in the formula is that the learned similarity S between two cells should be small if the distance between them is large. The second term is a regularization term that prevents the learned similarities from becoming too close to an identity matrix (**Supplementary Fig. 22**). If there are C subpopulations, the gene expressions of cells of the same subtype should have high similarity; and ideally the effective rank of S should be C. Thus, the third term along with the constraint on L enforces the low-rank structure of S; the matrix $(I_N - S)$ is essentially the graph Laplacian[18], and the trace-minimization problem enforces approximately C connected components in a similarity graph that consists of nodes representing the cells and edge weights corresponding to pairwise similarity values in S. The fourth term imposes constraints on the kernel weights to avoid selection of a single kernel; we empirically found that this regularization improves the quality of learned similarity (**Supplementary Table 8**).

Kernel construction for SIMLR. In the default implementation of SIMLR, we use Gaussian kernels with various hyperparameters. Gaussian kernels, which generate better empirical performance than do other types of kernels[9,13], take the form

$$K\left(c_i, c_j\right) = \frac{1}{\epsilon_{ij}\sqrt{2\pi}} \exp\left(-\frac{\| c_i - c_j \|_2^2}{2\epsilon_{ij}^2}\right) \tag{3}$$

where $||c_i - c_j||_2$ is the Euclidean distance between cell i and cell j. The variance, ϵ_{ij}, can be calculated with different scales:

$$\mu_i = \frac{\sum_{l \in \text{KNN}(c_i)} \| c_i - c_j \|_2}{k},\ \varepsilon_{ij} = \frac{\sigma\left(\mu_i + \mu_j\right)}{2} \tag{4}$$

where $\text{KNN}(c_i)$ represents cells that are top k neighbors of the cell i. Hence, each kernel is decided by a pair of parameters (σ, k). We set k = 10,12,14,…,30 and σ = 1.0,1.25,1.5,1.75,2, resulting in 55 different kernels in total. However, we empirically show that our method is insensitive to the number of kernels and choices of parameters (**Supplementary Fig. 23**).

Optimization algorithm. We optimize over S, L, and w. The optimization problem formulated above is nonconvex, but the objective function for each variable conditional on the other two variables being fixed is convex[19]. So we can apply an alternating convex optimization method to solve this tri-convex problem efficiently (**Supplementary Note 7**).

Initialization of S, w, *and* L. The weight of multiple kernels, w, is initialized as an uniform distribution vector; i.e.,

$$w = \left(\frac{1}{G}, \frac{1}{G}, \ldots, \frac{1}{G}\right),$$

where G is the number of kernels. The similarity matrix S is initialized as

$$S_{ij} = \sum_l w_l K_l\left(c_i, c_j\right).$$

L is initialized as the top C eigenvectors of $I_N - S$.

Step 1: fixing L *and* w *to update* S. When we minimize the objective function with respect to (w.r.t.) the similarity matrix S, we can rewrite the optimization problem as follows.

$$\begin{aligned} \underset{S}{\text{minimize}} & -\sum_{i,j}\left[\sum_l w_l K_l\left(c_i, c_j\right) + \gamma\left(LL^T\right)_{ij}\right] S_{ij} + \beta \| S \|_F^2 \\ \text{subject to} & \sum_j S_{ij} = 1 \text{ and } S_{ij} \geq 0 \text{ for all } (i, j) \end{aligned} \tag{5}$$

The first summation term in the objective as well as constraints are all linear, and the second summation in the objective is a simple quadratic form that can be solved in polynomial time[20].

Step 2: fixing S *and* w *to update* L. When we minimize the objective function w.r.t. the latent matrix L, we can rewrite the optimization problem as follows.

$$\underset{L}{\text{maximize}}\ \mathbf{tr}(L^T(S - I_N)L)$$
$$\text{subject to } L^T L = I_C \tag{6}$$

The trace of $L^T(S - I_N)L$ is maximized when L is an orthogonal basis of the eigenspace associated with the C largest eigenvalues of $(S - I_N)$[21]. Thus, L can be computed efficiently using any matrix numeric toolbox.

Step 3: fixing S *and* L *to update* w. When we minimize the objective function w.r.t. the kernel weights w, we can rewrite the optimization problem as follows.

$$\underset{w}{\text{maximimize}} \sum_l w_l \sum_{i,j} K_l\left(c_i, c_j\right) S_{ij} - \rho \sum_l w_l \log w_l$$
$$\text{subject to } \sum_l w_l = 1, w_l \geq 0 \tag{7}$$

The problem with a convex objective and linear constraints can be solved by any standard convex optimization method[20].

Step 4: similarity enhancement by diffusion. We apply a diffusion-based step[22] to enhance the similarity matrix S and reduce the effects of noise and dropouts (**Supplementary Fig. 24**). Given S, we construct a transition matrix P such that

$$P_{ij} = \frac{S_{ij}\, 1_{\{j \in A_K(i)\}}}{\sum_l S_{il}\, 1_{\{l \in A_K(i)\}}} \tag{8}$$

where $\mathbf{1}_{\{.\}}$ represents the indicator function, and $A_K(i)$ represents the set of indices of cells that are the K top neighbors of cell i under the learned distance metric. Under this construction, the transition matrix is sparse, and we preserve most of the similarity structure. The diffusion-based method that we apply to enhance the similarity S has the following update scheme.

$$H_{ij}^{(t+1)} = \tau H_{ij}^{(t)} P + (1 - \tau) I_N \tag{9}$$

where we have $H_{ij}^{(0)} = S_{ij}$ as an input, and the final iteration of H_{ij} is used as the new similarity measure S_{ij}. This additional diffusion step would further alleviate the noise effects in single-cell RNA-seq data sets. However, because of its high computational complexity, we cannot apply this step when dealing with large data sets.

SIMLR iterates steps 1–4 above until convergence. We use the eigengap[10], defined as the difference between the $(C + 1)$th and Cth eigenvalues, as the convergence criterion (**Supplementary Note 7**). In our implementation, SIMLR converges within around ten iterations (**Supplementary Fig. 25**). We also included methodologies to tune the hyperparameters of the algorithm (**Supplementary Note 7** and **Supplementary Table 9**).

Dimension reduction with SIMLR. SIMLR relies on the stochastic neighbor embedding methodology[13] for dimension reduction, with an important modification: t-SNE computes the similarity of the high-dimensional data points using a Gaussian kernel as a distance measure and projects the data onto a lower dimension that preserves this similarity. Instead of using the gene expression matrix as an input to t-SNE, we use the learned cell-to-cell similarity S (**Supplementary Note 8**; **Supplementary Figs. 26** and **27**).

For visualization, we use the dimension reduction algorithm to project the data into two or three dimensions so that the hidden structures in the data can be depicted intuitively. For clustering with k-means[15], we use the same approach to reduce the dimensions to B, resulting in an $N \times B$ latent matrix Z, to which we apply k-means to assign labels to each cell. The number of reduced dimensions B is by default equal to the number of desired clusters C, where C is also the rank-constraint parameter described above. We developed two heuristics to estimate C for clustering (**Supplementary Note 9**; **Supplementary Figs. 28** and **29**).

Gene prioritization using the learned similarity. The learned cell-to-cell similarity matrix S can be leveraged to perform gene prioritization; we rank each gene by measuring how its expression values across the cells correlate with the learned cell-to-cell similarity. Given the similarity S and the expression of a gene across all cells f, we adopt the Laplacian score[23],

$$LS(f) = \frac{f' S f}{f' f} \tag{10}$$

a well-known unsupervised feature-ranking method[23] to measure the concordance between genes and similarity. The higher the Laplacian score, the more important the gene is to globally differentiate the subpopulations of cells. However, the Laplacian score is typically very sensitive to noise in the measured similarity. To overcome this issue, we propose a bootstrap approach for which we randomly subsample a proportion of cells (e.g., 80% of the total number of cells) and rank genes based on the Laplacian score on the subset of learned similarity[24].

Large-scale extension of SIMLR. We extend SIMLR to deal with data sets of tens of thousands of cells. The key idea is to use k-nearest-neighbor (KNN) similarity to approximate the full pairwise similarity. The first step is to use a state-of-the-art nearest neighbor search technique ANNOY (https://github.com/spotify/annoy). ANNOY optimizes indexing of the KNN graph by relying on the heuristic that a neighbor's neighbor is probably a neighbor too. After the construction of KNN graph, we only update similarities in these preselected top k neighbors for each cell. Since the similarity is sparse, we use Spectra (http://yixuan.cos.name/spectra/), a representative eigen-decomposition library, to solve for L (**Supplementary Note 10**). When we compute similarity enhancement, instead of calculating the closed-form solution which involves a large matrix inversion, we only apply a limited number of iterations to get an approximation of the final solution.

After we obtain the cell-to-cell similarity, we can perform both cell subpopulation identification and cell visualization. It is computationally expensive to obtain the embedding from a t-SNE-type framework. Instead, we adopted a spectral clustering algorithm which is essentially equivalent to applying k-means on our latent variable L in our SIMLR. This simple algorithm is very effective for clustering sparse similarities and scales up to tens of thousands of cells. For visualization, since we are only mapping cell-to-cell similarity into 2D or 3D space, it is still computationally tractable to apply a t-SNE-type embedding. However,

doi:10.1038/nmeth.4207

we modified the Barnes–Hut algorithm in t-SNE[25] by utilizing the sparse properties of our similarity (**Supplementary Note 10**).

Data sources of single-cell RNA-seq. We used seven single-cell RNA-seq data sets in our paper. The first four data sets contain less than 1,000 cells each, while the last three data sets contain thousands to tens of thousands of cells (**Supplementary Table 10**). Below are detailed descriptions of all the single-cell RNA-seq data sets.

(1) 11 cell populations, including neural cells and blood cells (Pollen data set)[2]. This data set was designed to test the utility of low-coverage single-cell RNA-seq in identifying distinct cell populations, and thus contains a mixture of diverse cell types: skin cells, pluripotent stem cells, blood cells, and neural cells. This data set includes samples sequenced at both high and low depth; we analyzed the high-depth samples, which were sequenced to an average of 8.9 million reads per cell.
(2) Neuronal cells with sensory subtypes (Usoskin data set)[3]. This data set contains 622 cells from the mouse dorsal root ganglion, with an average of 1.14 million reads per cell. The authors divided the cells into four neuronal types: peptidergic nociceptors, nonpeptidergic nociceptors, neurofilament containing, and tyrosine hydroxylase containing.
(3) Embryonic stem cells under different cell cycle stages (Buettner data set)[11]. This data set was obtained from a controlled study that quantified the effect of the cell cycle on gene expression level in individual mouse embryonic stem cells (mESCs). An average of 0.5 million reads were obtained for each of the 182 cells, and at least 20% of the reads were mapped to known exons on the mm9 mouse genome. The cells were sorted for three stages of the cell cycle using fluorescence-activated cell sorting, and they were validated using gold-standard Hoechst staining.
(4) Pluripotent cells under different environment conditions (Kolodziejczyk data set)[4]. This data set was obtained from a stem cell study on how different culture conditions influence pluripotent states of mESCs. This study quantified the expression levels of about 10,000 genes across 704 mESCs from nine different experiments involving three different culture conditions. An average of 9 million reads were obtained for each cell, and over 60% of the reads mapped to exons on the *Mus musculus* genome.
(5) Mouse retina cells with 39 subtypes (Macoskco data set)[6]. Obtained by Drop-seq, a droplet-based high-throughput technique[7], this data set includes UMI (3-end) counts for 44,808 cells (identified by their customized computational pipeline). The cell types were classified via PCA and density-based clustering, and they were validated by differential gene expression. In line with the original processing procedure[6], we filtered out cells with less than 900 genes (resulting in 11,040 cells) for unsupervised analysis.
(6) PBMCs from a healthy human (PBMC68k data set)[7]. scRNA-seq libraries were generated by the 10× Genomics GemCode platform, a droplet-based high-throughput technique[7], and 68,560 cells with UMI (3′-end) counts were identified by their customized computational pipeline. This cell population includes major immune cell types in a healthy human.
(7) Cells from the mouse cortex and hippocampus (Zeisel data set)[17], collected using unique molecule identifier (UMI) assays and 3′-end counting. 3,005 cells from the mouse brain were collected, and 47 subtypes were identified by hierarchical biclustering and validated by gene markers.

For all the data sets above, we applied a logarithmic transformation

$$f(X) = \log_{10}(X+1)$$

to the single-cell raw expression data prior to analysis.

18. von Luxburg, U. *Stat. Comput.* **17**, 395–416 (2007).
19. Wang, B. *et al. Adv. Neural Inf. Process. Syst.* 3297–3305 (2016).
20. Nesterov, Y., Nemirovskii, A. & Ye, Y. *Interior-Point Polynomial Algorithms in Convex Programming* (SIAM, 1994).
21. Parlett, B.N. *The Symmetric Eigenvalue Problem* (SIAM, 1980).
22. Yang, J. & Leskovec, J. In *Proc. 10th IEEE Conf. Data Min.* (eds. Webb, G.I. *et al.*) 599–608 (IEEE, 2010).
23. He, X., Cai, D. & Niyogi, P. *Adv. Neural Inf. Process. Syst.* **18**, 507–514 (2005).
24. Kolde, R., Laur, S., Adler, P. & Vilo, J. *Bioinformatics* **28**, 573–580 (2012).
25. Van Der Maaten, L. *J. Mach. Learn. Res.* **15**, 3221–3245 (2014).