Similarity Learning with Scaled Lasso for Single Cell Clustering

July 18, 2018

Introduction

One of the drawbacks of the RNA-sequencing (RNA-seq) technology is that it assumes all the cells come from a homogeneous population, and it fails to account for the different cell types. Today, the recent advances in single-cell RNA sequencing technology (scRNA-seq) allow scientists to observe cell-to-cell variation by providing the expression level measurements at the single-cell level. There have been subsequent developments of tools that can cluster the cells into subtypes by measuring cell-to-cell expression variability. However, the unique qualities of the new scRNA-seq data pose important statistical challenges that have not been fully addressed. In particular, scRNA-seq data has a high proportion of zeros. A zero entry can either mean the gene is not expressed at all or the gene is expressed at a very low level so that the sequencing tool could not detect any signal. The zeros from the second source, or the "technical zeros," can introduce bias [3]. Here, we present an unsupervised cell type learning tool for scRNA-seq data that

- detection rate
- reference panel
- hybrid approach for large scale analysis
- methodology in layman language (measure distance in different spaces)

Detection Rate Bias Correction

The zeros in RNA-seq data [3]

Low Coverage Data

Large Scale Extension

We present the results of eight "small" data sets with number of cells lower than 1050 with known true labels. [1][2][4][5][6][7][8]. The performance is evaluated by the metric of Adjusted Rand Index.

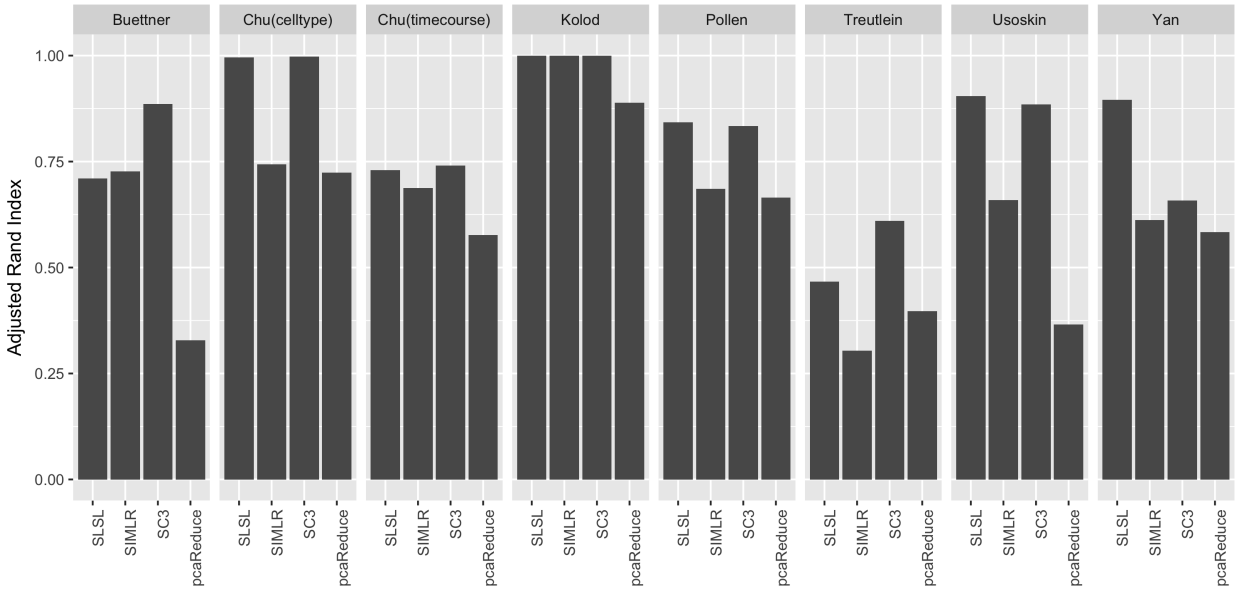


Figure 1: Comparing the Adjusted Rand Index for 8 sample data sets with benchmark methods. SLSL competes with SC3 for most data sets. The details are in Table 1.

	Spearman	Pearson	Euclidean	SLSL	SIMLR	SC3	pcaReduce
Kolod	1.000	1.000	1.000	1.000	1.000	1.000	0.889
Pollen	0.807	0.834	0.733	0.843	0.686	0.834	0.665
Usoskin	0.634	0.644	0.617	0.904	0.659	0.885	0.366
Buettner	0.730	0.545	0.690	0.710	0.727	0.886	0.328
Yan	0.895	0.895	0.895	0.895	0.612	0.658	0.583
Treutlein	0.442	0.415	0.441	0.467	0.304	0.610	0.397
Chu(celltype)	0.996	0.994	0.713	0.996	0.744	0.998	0.724
Chu(timecourse)	0.727	0.728	0.728	0.730	0.687	0.740	0.577

Table 1: Adjusted Rand Index result for 8 sample data sets. The first three columns use only single measurement of distance when constructing the kernel matrix. Combining all three kernel matrices for similarity learning is in the SLSL column, which we present as the final result.

1 Method

Consider $p \times n$ gene expression level matrix, where the rows represent p genes and the columns represent n cells. C is the number of true clusters. We build a kernel array $P \in \mathbb{R}^{n \times n \times L}$, which has L different measures of distance using varying kernel parameters.

1.1 Gene Filter and Detection Rate Correction

As a pre-processing step, we implemented a gene filter that reduces the data set by removing observations that don't hold much information. We found genes where more than 90% of the cells have zero counts. This is computationally fast and simple, and previous works have shown that more complex filtering process does not necessarily improve the clustering result **??**. The main goal of this procedure is to reduce the computational burden by decreasing the data size.

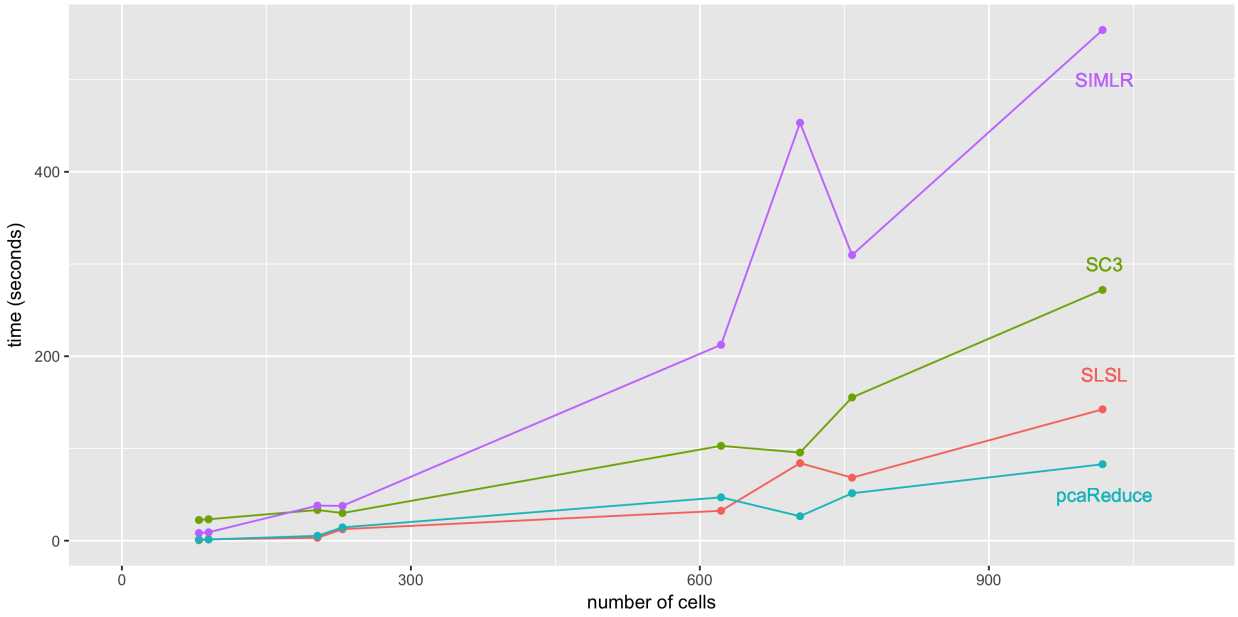


Figure 2: The time each software takes to cluster the data sets with given number of cells. It scales much better than SIMLR and SC3. The SLSL was run under the default setting with fixed number of clusters. The details are in Table 2.

	cellcount	SLSL	SC3	pcaReduce	SIMLR
Treutlein	80	0.530	22.433	1.167	8.367
Yan	90	1.529	23.235	1.310	9.150
Pollen	203	3.137	33.305	5.301	38.043
Buettner	229	12.482	29.981	14.426	37.801
Usoskin	622	32.451	102.856	47.031	212.438
Kolod	704	84.128	95.595	26.535	453.167
Chu2	758	68.363	155.362	51.452	309.754
Chu1	1,018	142.459	271.998	82.930	553.793

Table 2

Also, we

1.2 Measure Distance among the Cells

There are several ways to measure the distance between the two cells given their expression levels x_i and x_j. We explore three different ways: Euclidean distance $\|x_i - x_j\|_2$, Pearson correlation $1 - cor(x_i, x_j)$, and Spearman's rank order correlation $cov(rg_{y_i}, rg_{y_j})/(\sigma_{rg_{x_i}}\sigma_{rg_{x_j}})$. We can build three types of distance matrices D^c_{ij}, where c can be 1, 2, 3 each symbolizing Euclidean, Pearson correlation, and Spearman correlation. Then we build the similarity kernel matrix with D in the following way.

- **Build similarity matrices K_ℓ with different kernel parameters**
  Since we are using multiple kernels with different sets of parameters, use index $\ell \in \{1, 2, \cdots, L\}$ to denote that the kernel is built with parameters k_ℓ and σ_ℓ. The default initialization is $k_\ell \in \{15, 20, 30\}$ and $\sigma_\ell \in \{1, 1.2, 1.4, 1.6, 1.8, 2\}$. Now, for each ℓ, build $\mu_\ell \in \mathbb{R}^{n\times 1}$ vector that averages the distance to k nearest neighbors.

  $$\mu_{i\ell} = \frac{\sum_{j\in KNN_\ell(x_i)} \|x_i - x_j\|_2^2}{k_\ell}$$

  Now we can build the variance matrix ϵ_ℓ:

  $$\epsilon_{i,j,\ell} = \frac{(\mu_i + \mu_j)\sigma_\ell}{2}$$

  μ and ϵ account for the local structure. If two nodes are relatively isolated from other nodes, the distance between them is adjusted to be smaller. Finally, we use ϵ_ℓ to build the similarity matrix $K_\ell \in \mathbb{R}^{n\times n}$:

  $$K_{i,j,\ell} = N(D_{ij}; 0, \epsilon_{i,j,\ell})$$
  $$= \frac{1}{\sqrt{2\pi}\,\epsilon_{i,j,\ell}} exp\left(-\frac{D_{ij}}{2\epsilon^2_{i,j,\ell}}\right)$$

- **Build normalized distance matrix C**
  Build distance matrix C that again adjusts for the local structure:

  $$C_{i,j,\ell} = K_{i,i,\ell} + K_{j,j,\ell} - 2K_{i,j,\ell}$$

- **Using knn, build sparse similarity matrices P**
  For each ℓ (dropping the subscript ℓ in this section), only take the closest $k+1$ neighbors' distances and indices using C and build two matrices

  $$M, I \in \mathbb{R}^{n\times(k+1)}$$

  The i'th row of matrix M has an ordered distance for point i's $k+1$ nearest neighbors, and I has the corresponding indices of those neighbors. Then, convert this back to a similarity matrix by subtracting each column from the $(k+1)$'th column of M so that the $(k+1)$'th column becomes a vector of 0's. Removing this last column, now we have $\tilde{M} \in \mathbb{R}^{n\times k}$ that has the largest similarity in the first column and the smallest similarity in the k'th column. Then normalize $\tilde{M}$ so that each row sums to 1. The similarity measures are all 0's except these k nearest neighbors. Now create similarity matrix A by assigning these nonnegative similarity measures to its correct indices of the neighbors. Now, $A \in \mathbb{R}^{n\times n}$ is a transition probability matrix that is element-wise non-negative and has row sums of 1. Then finally create symmetric similarity matrix P from A.

  $$P = (A + A^T)/2$$

We can build $P \in \mathbb{R}^{n\times n\times L}$ for three different distance measures. We have a user input "kernel type", where they can choose from "euclidean", "pearson", "spearman", and "combined". The default, "combined" kernel concatenates the P arrays for all three distance measures to extract information from all of them.

1.3 Similarity Learning

We solve the following to learn the final similarity matrix S.

$$\min_{S,\sigma} \sum_{i=1}^{L} \frac{\|S-P_i\|_F^2}{2n^2\sigma_i} + \tau\|S\|_1 + \gamma\|S\|_F^2 + \sum_{i=1}^{L}\frac{\sigma_i}{2}, \qquad S1=1, S\geq 0$$

Above employs the scaled lasso to give different weights to each observed similarity P_i and to control the noise level simultaneously. τ imposes sparsity on the final matrix, while γ prevents the similarity matrix from becoming too close to the identity matrix, creating too many clusters. However, the empirical results show that the algorithm works best when $\gamma = 0$, so the default setting is $\gamma = 0$. The constraint of $S1 = 1$ naturally prevents the matrix from becoming too close to a zero matrix, so τ can be big. The default is $\tau = 5$. The details of algorithm are below. Construct the Lagrangian,

$$\mathcal{L} = \tau\|Q\|_1 + +\frac{1}{2}\gamma\|Q\|_F^2 + \sum_{i=1}^{L}\frac{1}{2n^2\sigma_i}\|Q-P_i\|_F^2 + \sum_{i=1}^{L}\frac{\sigma_i}{2} + \langle Y, S-Q\rangle + \frac{\mu}{2}\|S-Q\|_F^2,$$

and update each part like following.

- **update** Q

$$\arg\min_Q \tau\|Q\|_1 + \frac{\gamma}{2}tr(Q^TQ) + \sum_{i=1}^{L}\frac{1}{2n^2\sigma_i}tr(Q^TQ - 2P_i^TQ + P_i^TP_i) + Y^TQ + \frac{\mu}{2}tr(Q^TQ - 2S^TQ + S^TS)$$

$$= \arg\min_Q \tau\|Q\|_1 + tr\left(\left(\frac{\mu}{2}+\frac{\gamma}{2}+\frac{1}{2}\sum_{i=1}^{L}\frac{1}{n^2\sigma_i}\right)Q^TQ - \left(\sum_{i=1}^{L}\frac{P_i}{n^2\sigma_i} + Y + \mu S\right)^T Q\right)$$

Denoting $\Phi = \sum_{i=1}^{L}\frac{1}{n^2\sigma_i}$,

$$\arg\min_Q \tau\|Q\|_1 + \left(\frac{\mu+\gamma+\Phi}{2}\right)\left\|Q - \frac{1}{\mu+\gamma+\Phi}\left(\sum_{i=1}^{L}\frac{P_i}{n^2\sigma_i} + Y + \mu S\right)Q\right\|_F^2$$

Take SVD of $\frac{1}{\mu+\gamma+\Phi}\left(\sum_{i=1}^{L}\frac{P_i}{n^2\sigma_i} + Y + \mu S\right)$ and take the soft threshold on the singular values using $\frac{\tau}{\mu+\gamma+\Phi}$.

- **update** S

$$\arg\min_S \left\|S - \left(Q - \frac{Y}{\mu}\right)\right\|_F^2, \qquad S\geq 0, S1 = 1,$$

Project $Q - \frac{Y}{\mu}$ onto the domain $S \geq 0$, $S1 = 1$.

- **update** σ

$$\arg\min_{\sigma_i} \sum_{i=1}^{L}\frac{1}{2n^2\sigma_i}\|Q-P_i\|_F^2 + \sum_{i=1}^{L}\frac{\sigma_i}{2}$$

Taking the first order condition leads to

$$\sigma_i^2 = \frac{n^2}{\|Q-P_i\|_F^2}$$

1.4 Network Diffusion

From the adjacency matrix for the K nearest neighbors A, We construct a transition matrix T such that

$$T_{ij} = \frac{S_{ij} 1_{j \in A_K(i)}}{\sum_l S_{il} 1_{l \in A_K(i)}}$$

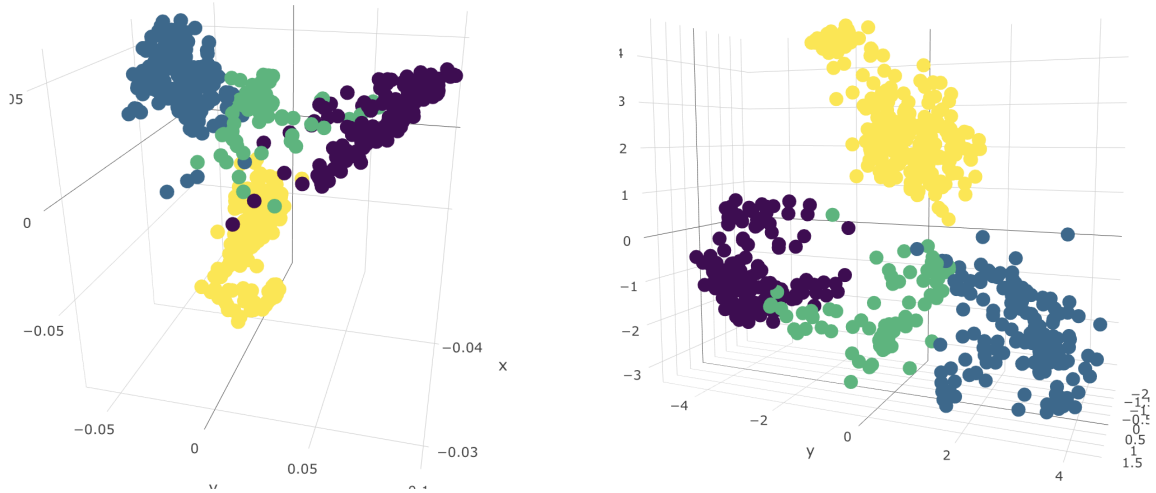
Using this, we update update the learned similarity matrix

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

where τ is empirically specified as 0.7.

1.5 Dimension Reduction and Kmeans

Two options : PCA and tSNE. PCA is faster. Checking performance now.. tSNE almost always performs better. It's around 30 times slower. The visualization is shown below. PCA is on the left and t-SNE is on the right.



1.6 Large Scale Extension

Building multiple kernels requires $n \times n \times L$ array and it quickly becomes burdensome when n is large. We make use of consensus clustering technique to take the cells by batch and combine the results. First, divide n cells into m groups where each group has approximately equal to or less than 500 cells. Then, we take all possible $\binom{m}{2}$ pairs to run the algorithm separately. We assign flexible number of clusters for each group, letting the algorithm decide C by eigengap criterion. This does not force each group to have a fixed number of clusters, and this helps in cases where there are rare cell types that only exist in few groups. Then we use *diceR* package and its CSPA function for consensus clustering.

1.7 Drop-seq and 10X Data

When the coverage is low,

References

[1] Florian Buettner, Kedar N Natarajan, F Paolo Casale, Valentina Proserpio, Antonio Scialdone, Fabian J Theis, Sarah A Teichmann, John C Marioni, and Oliver Stegle. Computational analysis of cell-to-cell heterogeneity in single-cell rna-sequencing data reveals hidden subpopulations of cells. *Nature biotechnology*, 33(2):155, 2015.

[2] Li-Fang Chu, Ning Leng, Jue Zhang, Zhonggang Hou, Daniel Mamott, David T Vereide, Jeea Choi, Christina Kendziorski, Ron Stewart, and James A Thomson. Single-cell rna-seq reveals novel regulators of human embryonic stem cell differentiation to definitive endoderm. *Genome biology*, 17(1):173, 2016.

[3] Stephanie C Hicks, F William Townes, Mingxiang Teng, and Rafael A Irizarry. Missing data and technical variability in single-cell rna-sequencing experiments. *Biostatistics*, 2017.

[4] Aleksandra A Kolodziejczyk, Jong Kyoung Kim, Jason CH Tsang, Tomislav Ilicic, Johan Henriksson, Kedar N Natarajan, Alex C Tuck, Xuefei Gao, Marc Bühler, Pentao Liu, et al. Single cell rna-sequencing of pluripotent states unlocks modular transcriptional variation. *Cell stem cell*, 17(4):471–485, 2015.

[5] Alex A Pollen, Tomasz J Nowakowski, Joe Shuga, Xiaohui Wang, Anne A Leyrat, Jan H Lui, Nianzhen Li, Lukasz Szpankowski, Brian Fowler, Peilin Chen, et al. Low-coverage single-cell mrna sequencing reveals cellular heterogeneity and activated signaling pathways in developing cerebral cortex. *Nature biotechnology*, 32(10):1053, 2014.

[6] Barbara Treutlein, Doug G Brownfield, Angela R Wu, Norma F Neff, Gary L Mantalas, F Hernan Espinoza, Tushar J Desai, Mark A Krasnow, and Stephen R Quake. Reconstructing lineage hierarchies of the distal lung epithelium using single-cell rna-seq. *Nature*, 509(7500):371, 2014.

[7] Dmitry Usoskin, Alessandro Furlan, Saiful Islam, Hind Abdo, Peter Lönnerberg, Daohua Lou, Jens Hjerling-Leffler, Jesper Haeggström, Olga Kharchenko, Peter V Kharchenko, et al. Unbiased classification of sensory neuron types by large-scale single-cell rna sequencing. *Nature neuroscience*, 18(1):145, 2015.

[8] Liying Yan, Mingyu Yang, Hongshan Guo, Lu Yang, Jun Wu, Rong Li, Ping Liu, Ying Lian, Xiaoying Zheng, Jie Yan, et al. Single-cell rna-seq profiling of human preimplantation embryos and embryonic stem cells. *Nature Structural and Molecular Biology*, 20(9):1131, 2013.