(2017), **0**, 0, *pp.* 1–31
doi:xx.xxx/sc_paper

Column subset selection for single-cell RNA-Seq clustering

SHANNON R. MCCURDY*, VASILIS NTRANOS,

LIOR PACHTER

smccurdy@berkeley.edu

Summary

The first step in the analysis of single-cell RNA sequencing (scRNA-Seq) is dimensionality reduction, which reduces noise and simplifies data visualization. However, techniques such as principal components analysis (PCA) fail to preserve non-negativity and sparsity structures present in the original matrices, and the coordinates of projected cells are not easily interpretable. Commonly used thresholding methods avoid those pitfalls, but ignore collinearity and covariance in the original matrix. We show that a deterministic column subset selection (DCSS) method possesses many of the favorable properties of PCA and common thresholding methods, while avoiding pitfalls from both. We derive new spectral bounds for DCSS. We apply DCSS to two measures of gene expression from two scRNA-Seq experiments with different clustering workflows, and compare to three thresholding methods. In each case study, the clusters based on the small subset of the complete gene expression profile selected by DCSS are similar to clusters produced from the full set. The resulting clusters are informative for cell type.

*To whom correspondence should be addressed.

Key words: Clustering; Column Subset Selection; Leverage scores; scRNA-Seq; Thresholding.

1. Introduction

Advances in RNA sequencing technology have recently made it possible to measure the genome-wide expression profile of single cells (Tang *et al.*, 2009). This promising technology is not without computational and analytical challenges, some of which include quality control, quantification, normalization, technical variability, and other confounding factors such as batch effects (Stegle, Teichmann and Marioni, 2015; Wagner, Regev and Yosef, 2016). More general challenges stem from the high dimensionality of the expression profiles: for example, selecting informative features from within the expression profiles.

One use for single-cell RNA sequencing (scRNA-Seq) data is the characterization of heterogeneity of expression within a population of cells and the discovery of new cell types through clustering of expression profiles (Zeisel *et al.*, 2015). This note explores the following question: is it possible reduce the number of features in the expression profile without a large effect on the error rate for clustering and classification? This question is inspired by the quality control and technical variability challenges of scRNA-Seq. Common techniques for quality control and technical variability reduction include simple thresholding schemes and principal components analysis (PCA). Both of these techniques reduce the number of features in the data matrix.

One commonly used technique to reduce the number of features in the data matrix involves selecting columns from the original data matrix $\mathbf{A}$, to form a column submatrix $\mathbf{C}$, by thresholding the individual columns based on a score. Frequently used scores are on measures of abundance (Lun, McCarthy and Marioni, 2016), empirical variance (Kwon, Fan and Kharchenko, 2017), abundance and empirical variance (McCarthy *et al.*), and index of dispersion (empirical variance/mean) (Satija *et al.*, 2015; Trapnell *et al.*, 2014). Read count thresholds are intended to reduce low-abundance genes (Bourgon, Gentleman and Huber, 2010) or genes with high dropout rates (Brennecke *et al.*,

2013), as these genes are not considered informative. Variance thresholding methods assume that the most variable genes are responsible for the important differences between cells (McCarthy *et al.*). Index of dispersion thresholding has a natural interpretation in terms of formal hypothesis testing, when the null model for gene abundance is the Poisson distribution (Cox and Lewis, 1966). We call these methods *simple* thresholding methods, because the score for each column i depends only on column i. Furthermore, within each column i, covariance between the rows (cells) of that column is not taken into account. By selecting columns and not linear combinations of columns from $\mathbf{A}$, the elements of $\mathbf{C}$ will maintain the properties of non-negativity, sparsity, and interpretability, an advantage over PCA, but there are no guarantees that $\mathbf{C}$ will have similar properties to the original data matrix $\mathbf{A}$.

Replacing the original data matrix of scRNA-Seq expression profiles with a rank-k PCA truncation of the profiles is another commonly used technique to reduce the number of features and the technical variability (Wagner, Regev and Yosef, 2016). To understand the PCA truncation, we must establish some matrix notation that we will use throughout this note. We orient the original data matrix $\mathbf{A}$ so that the n rows are cells and d columns are features, where $n < d$. For PCA, singular value decomposition (SVD) is performed on the column-mean centered matrix $\tilde{\mathbf{A}} = \mathbf{A} - \mathbf{1}\boldsymbol{\mu}^T$, where $\mathbf{1}$ is an $n \times 1$ column vector and $\boldsymbol{\mu} = \frac{1}{n}\mathbf{A}^T\mathbf{1}$ is a $d \times 1$ column vector of column-means. The sum of the spectrum of eigenvalues of $\tilde{\mathbf{A}}\tilde{\mathbf{A}}^T$ is proportional to the total empirical variance of $\mathbf{A}$. The rank-k PCA truncation of $\mathbf{A}$, which we call $\tilde{\mathbf{T}}$, is the rank-k SVD truncation of $\tilde{\mathbf{A}}$. SVD is reviewed in Sec. 6.1, and the formula for $\tilde{\mathbf{T}}$ is provided there. As a consequence of the SVD, the spectrum of the square of the rank-k PCA truncation $\tilde{\mathbf{T}}$ is identical to the spectrum of the square of the mean-centered data matrix $\tilde{\mathbf{A}}$ up to rank k; PCA gives a rank-k approximation to the mean-centered data $\tilde{\mathbf{A}}$ that preserves the maximum empirical variance of $\mathbf{A}$. PCA is performed to reduce technical variability under the assumption that the technical variation is primarily captured by the non-leading eigenvalues and eigenvectors of $\tilde{\mathbf{A}}\tilde{\mathbf{A}}^T$.

The drawback of replacing the original data matrix with the rank-k PCA truncation of the data that it fails to preserve non-negativity and sparsity structures present in the original data matrix, and the coordinates of projected cells are not interpretable in terms of single features.

The goal of column subset selection (CSS) is to extract from a matrix $\mathbf{A}$ a column submatrix $\mathbf{C}$ that conserves favorable properties, such as conditions on the spectrum of the column submatrix $\mathbf{C}$ (Tropp, 2009). Like the simple thresholding methods, CSS maintains the properties of non-negativity, sparsity, and interpretability, and like PCA, CSS conserves favorable matrix properties. Similar to the simple thresholding methods discussed above, each column has a score, however in CSS algorithms, the score for each column i also depends on all of the other columns. We will consider rank-k subspace leverage scores in this note. Leverage scores have been considered for regression diagnostics and outlier detection in statistics (Velleman and Welsch, 1981; Chatterjee and Hadi, 1986) and were brought to prominence more recently in the context of randomized matrix algorithms (Drineas, Mahoney and Muthukrishnan, 2006). The rank-k subspace leverage score $\tau_i(\mathbf{A}_k)$ for the i^{th} column of $\mathbf{A}$ is,

$$\tau_i(\mathbf{A}_k) = \mathbf{a}_i^T(\mathbf{A}_k\mathbf{A}_k^T)^+\mathbf{a}_i, \tag{1.1}$$

where the i^{th} column of $\mathbf{A}$ is an $(n \times 1)$-vector denoted by $\mathbf{a}_i$, $\mathbf{M}^+$ denotes Moore-Penrose pseudoinverse of $\mathbf{M}$, and $\mathbf{A}_k$ is the rank-k SVD approximation to $\mathbf{A}$, defined in Sec. 6.1. The leverage score $\tau_i(\mathbf{A}_k)$ can also be written as the solution to the following optimization problem (Cohen *et al.*, 2015),

$$\tau_i(\mathbf{A}_k) = \min_{\mathbf{A}_k\mathbf{x}=\mathbf{a}_i} ||\mathbf{x}||_2^2 \qquad \mathbf{x} \in \mathbb{R}^d, \tag{1.2}$$

where $||\mathbf{x}||_2^2$ refers to the Euclidean (L_2) norm of the vector $\mathbf{x}$. The vector $\mathbf{x}$ measures how easily the column a_i can be written as a linear combination of the columns of $\mathbf{A}_k$. Eqn. 1.2 shows that leverage scores capture the importance of each column $\mathbf{a}_i$ in the column space of $\mathbf{A}_k$ and are sensitive to collinearity between columns. We illustrate this point with a toy example in Sec. 2.1.

CSS algorithms select columns either with a random sampling procedure (such as in Drineas, Mahoney and Muthukrishnan (2006)) or a deterministic procedure. We showcase the deterministic CSS (DCSS) algorithm introduced by Papailiopoulos, Kyrillidis and Boutsidis (2014). Papailiopoulos, Kyrillidis and Boutsidis (2014) show that for datasets with power-law decay in $\tau_i(\mathbf{A}_k)$, DCSS will select a least-squares approximation for $\mathbf{A}$, $\mathbf{C}\mathbf{C}^\dagger\mathbf{A}$, requiring fewer columns with the same accuracy than random sampling methods. One of the contributions of this note is a new bound for the spectrum of the square of $\mathbf{C}$ selected by DCSS projected onto the rank-k subspace that best approximates $\mathbf{A}$ (Eqn. 2.9). This bound means that, once both $\mathbf{C}$ and $\mathbf{A}$ are projected onto the rank-k subspace that best approximates $\mathbf{A}$, $\mathbf{C}\mathbf{C}^T$ is "close" to $\mathbf{A}\mathbf{A}^T$. Another consequence is that the Frobenius norm of $\mathbf{C}$ is bounded (Eqn. 2.10). The Frobenius norm is a measure of the "size" of a matrix, so this bound provides confidence that the DCSS column matrix $\mathbf{C}$ is also similar in "size" to $\mathbf{A}$ and $\mathbf{A}_k$. In the event that DCSS is performed on a mean-centered matrix $\tilde{\mathbf{A}}$, the Frobenius norm provides a measure of empirical variance. We also show a similar bound holds for random sampling (Eqn. 2.11), and under the assumption of power-law decay, DCSS requires fewer columns for the same error than random sampling.

In addition to the spectral bound, we present two case studies on two different scRNA-Seq experimental and analysis workflows to illustrate empirically the effect of thresholding features with DCSS compared to read count, variance, and index of dispersion on clustering and classification. To the best of our knowledge, this is the first time DCSS has been applied to scRNA-Seq data. The first case study is the genome-wide expression profiles of $3,005$ cells from the mouse cortex and hippocampus (Zeisel *et al.*, 2015) and the clustering workflow of Ntranos *et al.* (2016). The second case is the genome-wide expression profiles of $4,423$ cells from mouse bone marrow (Paul *et al.*, 2015) and the trajectory workflow of Setty *et al.* (2016). In both case studies, DCSS reduces the low abundance genes and maintains many of the most variable and over-dispersed genes. This shows that DCSS shares the best features of the simple thresholding methods and, like PCA, comes

with additional bounds on the spectrum. This supports our conclusion that DCSS can be used instead of the simple thresholding methods for quality control and to reduce technical variability, in addition to selecting informative features. In both case studies, only a small fraction of the features are necessary to obtain clusters reflecting cell types, consistent with results in (Kwon, Fan and Kharchenko, 2017). We show that the error rate between the clustering assignments computed with the complete expression profile and the reduced expression profile is small.

2. Methods

The aim of this note is to explore the effect of thresholding features, measurements of gene expression, with DCSS. We compare DCSS to simple thresholding methods and also to the complete data. These thresholding methods are the first step in the pre-processing workflow. In this section, we include the DCSS algorithm for completeness, and we describe the new bounds for DCSS.

2.1 *The DCSS algorithm (Papailiopoulos, Kyrillidis and Boutsidis, 2014)*

Algorithm 1. *The DCSS algorithm selects for the submatrix* $\mathbf{C}$ *all columns* i *with a rank-*k *subspace leverage score* $\tau_i(\mathbf{A}_k)$ *above a threshold* θ*, determined by the error tolerance* ϵ *and the rank,* k*. The algorithm is as follows.*

1. *Choose the rank,* k*, and the error tolerance,* ϵ*.*
2. *For every column* i*, calculate the rank-*k *subspace leverage scores* $\tau_i(\mathbf{A}_k)$ *(Eqn. 1.1).*
3. *Sort the columns by* $\tau_i(\mathbf{A}_k)$*, from largest to smallest. The sorted column indices are* π_i*.*
4. *Define an empty set* $\Theta = \{\}$*. Starting with the largest sorted column index* π_0*, add the corresponding column index* i *to the set* Θ*, in decreasing order, until,*

$$\sum_{i \in \Theta} \tau_i(\mathbf{A}_k) > k - \epsilon, \tag{2.3}$$

and then stop. Note that $k = \sum_{i=1}^{d} \tau_i(\mathbf{A}_k)$. *It will be useful to define* $\tilde{\epsilon} = \sum_{i \notin \Theta} \tau_i(\mathbf{A}_k)$. *Eqn. 2.3 can equivalently be written as* $\epsilon > \tilde{\epsilon}$.

5. . *If the set size* $|\Theta| < k$, *continue adding columns in decreasing order until* $|\Theta| = k$.

6. *The leverage score* $\tau_i(\mathbf{A}_k)$ *of the last column* i *included in* Θ *defines the leverage score threshold* θ.

7. *Introduce a rectangular selection matrix* $\mathbf{S}$ *of size* $d \times |\Theta|$. *If the column indexed by* (i, π_i) *is in* Θ, *then* $\mathbf{S}_{i,\pi_i} = 1$. $\mathbf{S}_{i,\pi_i} = 0$ *otherwise. The DCSS submatrix is* $\mathbf{C} = \mathbf{AS}$.

Theorem 3 of Papailiopoulos, Kyrillidis and Boutsidis (2014) states that when the rank-k subspace leverage scores exhibit a power-law decay in the sorted column index π_i,

$$\tau_{\pi_i}(\mathbf{A}_k) = \pi_i^{-a} \tau_{\pi_0}(\mathbf{A}_k) \qquad a > 1, \tag{2.4}$$

the number of sample columns selected by DCSS is,

$$|\Theta| = \max\left(\left(\tfrac{2k}{\epsilon}\right)^{\frac{1}{a}} - 1, \left(\tfrac{2k}{(a-1)\epsilon}\right)^{\frac{1}{a-1}} - 1, k \right). \tag{2.5}$$

Papailiopoulos, Kyrillidis and Boutsidis (2014) demonstrate the power-law decay behavior of many real-world datasets; we show that this behavior is a reasonable assumption for the scRNA-Seq applications in Sec. 3.

For a statistical interpretation of DCSS, consider the data $\mathbf{a}_i$, $i = 1, \ldots, d$ to be identically and independently distributed (i.i.d.) according to the degenerate multivariate distribution $\mathcal{N}(0, \mathbf{A}_k \mathbf{A}_k^T)$. See Rao (1973) pg. 527-528 for a discussion of the degenerate multivariate distribution. Then the total likelihood of the data matrix $\mathbf{A}$ is,

$$\begin{aligned} \mathcal{L}(\mathbf{A}) &= \frac{1}{(2\pi)^{\frac{1}{2}kd} \prod_{j=1}^{k} |\sigma_j|^d} \exp\left(-\tfrac{1}{2} \sum_{i=1}^{d} \mathbf{a}_i^T \left(\mathbf{A}_k \mathbf{A}_k^T\right)^+ \mathbf{a}_i \right) \\ &= \frac{1}{(2\pi)^{\frac{1}{2}kd} \prod_{i=j}^{k} |\sigma_j|^d} \exp\left(-\tfrac{1}{2} \sum_{i=1}^{d} \tau_i(\mathbf{A}_k) \right), \end{aligned} \tag{2.6}$$

where $|\sigma_j|$ are the k largest singular values of $\mathbf{A}_k$. In contrast, the total likelihood of the DCSS matrix $\mathbf{C}$ is,

$$\begin{aligned}\mathcal{L}(\mathbf{C}) &= \frac{1}{(2\pi)^{\frac{1}{2}k|\Theta|}\prod_{j=1}^{k}|\sigma_j|^{|\Theta|}}\exp\left(-\tfrac{1}{2}\sum_{i\in\Theta}\tau_i(\mathbf{A}_k)\right)\\ &= \frac{1}{(2\pi)^{\frac{1}{2}k|\Theta|}\prod_{j=1}^{k}|\sigma_j|^{|\Theta|}}\exp\left(-\tfrac{1}{2}\sum_{i\in\Theta}\tau_i(\mathbf{A}_k)-\tfrac{1}{2}\sum_{i\notin\Theta}\tau_i(\mathbf{A}_k)+\tfrac{1}{2}\tilde{\epsilon}\right)\\ &= \mathcal{L}(\mathbf{A})\exp\left(\tfrac{1}{2}\tilde{\epsilon}\right)(2\pi)^{\frac{1}{2}k(d-|\Theta|)}\prod_{j=1}^{k}|\sigma_j|^{d-|\Theta|}. \end{aligned} \tag{2.7}$$

This shows that the DCSS matrix $\mathbf{C}$ preserves the total likelihood of the data up to a factor of $\exp\left(\frac{1}{2}\tilde{\epsilon}\right) < \exp\left(\frac{1}{2}\epsilon\right)$ and a normalization constant, under the assumption that the data is i.i.d. according to $\mathcal{N}(0, \mathbf{A}_k\mathbf{A}_k^T)$. Any other selection set Θ' of the same number of columns ($|\Theta'| = |\Theta|$) will have equal or greater error ($\epsilon \leqslant \epsilon'$). This interpretation illustrates that DCSS accounts for covariance $\mathbf{A}_k\mathbf{A}_k^T$ between rows (cells). In contrast, the Poisson null model for the index of dispersion assumes independence between rows (cells) for each column (gene).

The DCSS method has two parameters, k, ϵ which jointly determine the number of columns $|\Theta|$ in the DCSS column submatrix $\mathbf{C}$. The parameter k determines the rank of interest of the SVD approximation to $\mathbf{A}$. The tuning parameter ϵ is a measure of the error tolerance in the "size" of $\mathbf{C}$ compared to $\mathbf{A}_k$. The selection of these parameters is a model selection problem, and in concert with a loss function, one could select these parameters using one's preferred model selection method (e.g. cross-validation). The aim of this note, to compare clustering performed with the complete data matrix and a column submatrix, does not have a well-defined loss function, and so we use the heuristic "elbow" method for selecting k (Jolliffe, 2002), and we choose ϵ to be 0.1 or 0.05 in our applications.

As a toy example to illustrate how DCSS differs from the simple thresholding methods, consider the following toy data matrix,

$$\mathbf{A} = \begin{pmatrix} 40 & 20 & 10 \\ 20 & 10 & 15 \end{pmatrix}. \tag{2.8}$$

If the goal is to select a column submatrix with two columns, it is easy to check that simple thresholding by mean, variance, and index of dispersion all select the first and second columns. However, the resulting column submatrix is only rank 1, because the first and second columns are linearly dependent. In contrast, DCSS with $(k = 2, \epsilon > 0.2)$ will select the first and third columns, and the resulting DCSS column submatrix will be rank 2. Unlike the first three methods, DCSS takes into account the collinearity between columns in the selection procedure. If the DCSS error tolerance for this toy example is less than 0.2, DCSS will select all three columns.

We also mention two asides: first, in applications where the number of cells is far greater than the number of gene features $(n > d)$, the method can instead be applied to $\mathbf{A}^T$ instead of $\mathbf{A}$ to filter cells instead gene features; second, the method can be modified to select columns for any rank-k subspace defined by k singular vectors of $\mathbf{A}$, and not just the leading-k subspace (e.g. drop component 1 but include component 2). This could be useful when some of the leading singular vectors are highly correlated with batch or other confounding effects.

2.2 *New bounds for DCSS*

We derive a new spectral approximation bound (Bound 2.9) for the square of the submatrix $\mathbf{C}$ selected with DCSS and projected onto the rank-k subspace that best approximates $\mathbf{A}$.

THEOREM 2.1 Let $\mathbf{A} \in \mathbb{R}^{n \times d}$ be a matrix of at least rank k and $\tau_i(\mathbf{A}_k)$ be defined as in Eqn. 1.1. Construct $\mathbf{C}$ following the DCSS algorithm described in Sec. 2.1. Then $\mathbf{C}$ satisfies,

$$(1 - \epsilon)\mathbf{A}_k\mathbf{A}_k^T \preceq \mathbf{U}_k\mathbf{U}_k^T\mathbf{C}\mathbf{C}^T\mathbf{U}_k\mathbf{U}_k^T \preceq \mathbf{A}_k\mathbf{A}_k^T. \tag{2.9}$$

The symbol $\preceq$ denotes the Loewner partial ordering which is reviewed in Sec 6.1. Conceptually, the Loewner ordering is the generalization of the ordering of real numbers (e.g. $1 < 1.5$) to Hermitian matrices. This bound means that after projection onto the rank-k subspace that best approximates $\mathbf{A}$, $\mathbf{C}\mathbf{C}^T$ is "close" to $\mathbf{A}\mathbf{A}^T$ on that subspace. Statements of Loewner ordering are quite powerful;

important consequences include inequalities for the eigenvalues and Euclidean distances. Some of the consequences of the Loewner ordering are reviewed in Sec 6.1. Bound 2.9 and the fact that $\mathbf{C}\mathbf{C}^T \preceq \mathbf{A}\mathbf{A}^T$ implies a bound on the Frobenius norm of $\mathbf{C}$, a measure of the "size" of a matrix,

$$(1-\epsilon)||\mathbf{A}_k||_F^2 \leqslant ||\mathbf{C}||_F^2 \leqslant ||\mathbf{A}||_F^2. \tag{2.10}$$

In the event that $\mathbf{A}$ is mean-centered, this means that the total empirical variance of $\mathbf{C}$ is bounded from below by $(1-\epsilon)$ the variance in $\mathbf{A}_k$ and bounded from above by the total variance of $\mathbf{A}$. The proof of Bound 2.9 and Bound 2.10 is included in Sec. 6.2.

One simple consequence of Bound 2.9 is the following bound,

$$(1-\epsilon)\mathbf{A}_k\mathbf{A}_k^T \preceq \mathbf{U}_k\mathbf{U}_k^T\mathbf{C}\mathbf{C}^T\mathbf{U}_k\mathbf{U}_k^T \preceq (1+\epsilon)\mathbf{A}_k\mathbf{A}_k^T. \tag{2.11}$$

Bound 2.11 also holds for $\mathbf{C}$ selected by random sampling methods with t columns (see Sec. 6.3 for the theorem and proof). Thus, DCSS selects fewer columns with the same accuracy ϵ in Bound 2.11 for power-law decay in the rank-k subspace leverage scores when,

$$|\Theta| = \max\left(\left(\tfrac{2k}{\epsilon}\right)^{\frac{1}{a}} - 1, \left(\tfrac{2k}{(a-1)\epsilon}\right)^{\frac{1}{a-1}} - 1, k\right) < \tfrac{2}{\epsilon^2}(k+m\gamma)\left(1+\tfrac{1}{3}\epsilon\right)\ln\left(\tfrac{16k}{\delta}\right) \leqslant t. \tag{2.12}$$

In this expression, m is the number of columns with zero rank-k subspace leverage score, γ is the minimum non-zero leverage score, and δ is the probability that Bound 2.11 fails to hold under random sampling.

3. Results

We present two case studies where we compare DCSS to the simple thresholding methods of variance, count, and index of dispersion. We analyze the overlap in the selected columns. We also illustrate the effect of DCSS compared to the complete data for single-cell clustering.

3.1 *Zeisel* et al. *(2015)*

As a concrete illustration of the DCSS method, we focus on the genome-wide expression profiles of 3005 cells from the mouse somatosensory cortex and hippocampal CA1 region (Zeisel *et al.*, 2015) and the clustering workflow of Ntranos *et al.* (2016). The main contribution of Ntranos *et al.* (2016) is to perform clustering directly on the partition of reads into equivalence classes (ECs) rather than on a full quantification of reads into gene expression. ECs are a partition of reads into distinct classes, such that every read in a class maps to exactly the same set of transcripts (Nicolae *et al.*, 2011). This method is computationally scalable, comparable across scRNA-Seq experiments, and can be more accurate than clustering performed on a full quantification of reads into gene expression profiles (Ntranos *et al.*, 2016).

The Ntranos *et al.* (2016) data matrix $\mathbf{A}$ is $3,005$ cells $\times$ $246,981$ EC counts. By the elbow method, we choose $k = 5$ for the DCSS workflow (Fig. 1a). We select an error tolerance of $\epsilon = 0.1$. The rank-5 subspace leverage scores and the power-law fit for the top-scored $10,000$ ECs are shown in Fig. 1b. The column submatrix $\mathbf{C}$ has only 862 ECs, or approximately 0.3% of the total ECs. These ECs contain 42.3% of the reads. These 862 ECs map to $2,748$ transcripts and to $1,642$ genes. Table 1 contains the gene ontology term enrichment analysis (The Gene Ontology Consortium, 2015) on the genes corresponding to the DCSS ($k = 5, \epsilon = 0.1$) ECs. Enrichments relevant for the brain include neuron part, neuron projection, and olfactory receptor activity. The enrichment analysis has an important caveat: because we map ECs to transcripts without positing an error model, there could be a high rate of false positives in the resulting transcripts and genes.

We compare DCSS to the three simple thresholding methods with the same number of columns in Fig. 1c and Fig. 1d. These figures show the similarities and differences in columns selected by the four thresholding methods. The simple thresholding methods have sharp boundaries in Fig. 1c, while the DCSS boundary is not linearly separable. The DCSS boundary approximately interpolates between the count and variance boundaries, and is most distinct from the index of

dispersion boundary. Fig. 1d summarizes the overlap between selected columns in a Venn diagram. These figures illustrate that the DCSS method selects columns that are highly variable, have large counts, and frequently are over-dispersed; as such, the DCSS method is prescribed for quality control and to control technical variability.

The Ntranos *et al.* (2016) workflow for the Zeisel *et al.* (2015) dataset is to perform spectral clustering on pairwise Jensen-Shannon (JS) distances derived from the partition of reads into ECs. The spectral clustering clustering algorithm used is standard; the algorithm is to perform k-means clustering on the k-dimensional SVD projection of the normalized Laplacian of the symmetric similarity matrix $\mathbf{S}$. The similarity matrix used for spectral clustering is $S(\mathbf{p}, \mathbf{q}) = 1 - D_{JS}(\mathbf{p}, \mathbf{q})$, where $D_{JS}(\mathbf{p}, \mathbf{q})$ is the JS distance between two probability mass functions $\mathbf{p}, \mathbf{q} \in \mathbb{R}^d$. JS distances are well-suited to high-dimensional data, and provide more accurate clustering than L_2 distances on scRNA-Seq data (Ntranos *et al.*, 2016). For the Zeisel *et al.* (2015) data, the probability mass function for each cell is the vector of EC counts, normalized to sum to one. For the four thresholded workflows (DCSS, count, variance, and index of dispersion), the probability mass function for each cell is the subset vector of EC counts, normalized to one.

We evaluate the average spectral clustering classification error between the complete data and thresholded workflows, regarding the complete data workflow as the ground-truth. Since spectral clustering requires a random initialization for k-means, the average is over $T = 10$ random initializations. Fig. 2 shows the average spectral clustering classification error rate for both two and nine spectral clusters for the workflow with the matrix $\mathbf{A}$ and the workflow with the column submatrix $\mathbf{C}$ for various k, ϵ. The different cells were curated into 47 subtypes by Zeisel *et al.* (2015), but we evaluate our method on courser-grained classifications because we have higher confidence in the spectral clustering ground-truth. Two spectral clusters identify neurons and non-neurons, while nine spectral clusters only loosely correspond to the nine major cell types. We also include the error for the three simple thresholding methods with the same number of columns

as the DCSS method. We find that 0.3% of the total ECs give an error rate of 1.7% compared to the complete data for two clusters for $k = 5, \epsilon = 0.1$ DCSS; only a small fraction of the gene expression profiles currently produced in scRNA-Seq experiments may be necessary to obtain the clusters reflecting cell types.

3.2 *Paul* et al. *(2015)*

As a second application of the DCSS method, we focus on the genome-wide mRNA expression profiles of $4,423$ cells from mouse bone marrow myeloid progenitors (Paul *et al.*, 2015), and the *wishbone* trajectory workflow of Setty *et al.* (2016). The contribution of Setty *et al.* (2016) to scRNA-Seq is to use diffusion maps to identify components related to the development and maturation of cells, specifically myeloid and erythroid progenitors from hematopoietic stem and progenitor cells (HSPCs).

The Setty *et al.* (2016) data matrix for the (Paul *et al.*, 2015) dataset is $\mathbf{A}$ is $4,423$ cells $\times$ $14,955$ gene unique molecular identifier (UMI) counts. The Setty *et al.* (2016) workflow is quite involved. In brief, the *wishbone* algorithm creates a nearest-neighbor Euclidean distance graph. This graph is used to estimate all of the shortest path distances between a set of randomly sampled cells and the rest of the cells, and the shortest path distances are used to make the trajectory and branch assignments. The *wishbone* algorithm acts on a set of diffusion components which are selected for immune cell differentiation through a gene-set enrichment analysis. The diffusion components are calculated from the diffusion map of the similarity matrix derived from the Gaussian kernel of the 10-nearest-neighbor Euclidean distance matrix from the 15-dimensional PCA projection of the normalized UMI gene counts (Setty *et al.*, 2016).

We choose $k = 14$ for the DCSS workflow by the elbow method (Fig. 3a). We choose $k = 14$ rather than an elbow at a smaller k because the diffusion component workflow is sensitive to more components. We select an error tolerance of $\epsilon = 0.05$. The rank-14 subspace leverage scores and

the power-law fit for the top-scored $5,000$ genes are shown in Fig. 3b. The column submatrix $\mathbf{C}$ has $4,693$ genes, or approximately 31.4% of the total genes. These genes contain 90.4% of the UMI counts.

We compare DCSS thresholding with $k = 14, \epsilon = 0.05$ to the three simple thresholding methods with the same number of columns in Fig. 3c and Fig. 3d. The distribution of columns on the count-variance plots are qualitatively different between the Paul *et al.* (2015) data (Fig. 3c) and the (Zeisel *et al.*, 2015) data (Fig. 1c). This difference is expected due to the differences between ECs and gene UMI counts. Although the index of dispersion method is more differentiated from the other methods on the Paul *et al.* (2015) dataset, the behavior of the DCSS method in relation to the simple thresholding methods is similar between the datasets.

We calculate the average *wishbone* classification error between the two workflows, again regarding the complete data workflow as the ground-truth. Since the *wishbone* algorithm utilizes random sampling, the average is over $T = 10$ *wishbone* branch assignments. The original *wishbone* analysis included only diffusion components 1 and 2. We additionally include diffusion component 4, since it is also enriched for immune cell differentiation according to the GSEA. For the Paul *et al.* (2015) dataset, *wishbone* assigns cells to three branches. Setty *et al.* (2016) used the behavior of four markers (CD34, Gata1, Gata2, and Mpo) to verify that the three branches correspond to HSPCs, myeloid progenitors, and erythroid progenitors, and the behavior does not change with the inclusion of component 4. Fig. 3 shows the average branch assignment classification error rate for the workflow with the matrix $\mathbf{A}$ and the workflow with the column submatrix $\mathbf{C}$ for various k, ϵ, and also the three simple thresholding methods with the same number of columns as the DCSS method for each k, ϵ point. Not all the thresholding methods successfully complete the *wishbone* workflow at large ϵ, due to the sensitivity of the diffusion component GSEA enrichment analysis, which we perform with keyword string matching. We find that for the $k = 14, \epsilon = 0.05$ DCSS, 31.4% of the total genes give an error rate of 3.3% for three branch assignments compared to the

complete data; this supports our conclusion that only a small fraction of the gene expression profile from scRNA-Seq experiments may be necessary to obtain meaningful cell-type classifications.

4. Discussion

scRNA-Seq experiments allow researchers to probe the cell-specific heterogeneity in gene expression. Quality control and technical variability are significant challenges for scRNA-Seq experiments, and additionally the whole-genome expression profile is high-dimensional. In this note, we explore three existing simple thresholding schemes– count, variance, and index of dispersion– and propose a novel application of a thresholding scheme – DCSS– to select informative features and control quality and technical variability. We prove a bound on the "closeness" of the DCSS data submatrix to the complete data matrix (Eqn. 2.9), enlarging upon the existing set of error guarantees for DCSS (Papailiopoulos, Kyrillidis and Boutsidis, 2014), and illustrating the advantage of DCSS over the three simple thresholding schemes. Other advantages of DCSS include sensitivity to collinearity of features and covariance of cells. Since scRNA-Seq experiments are frequently used to cluster and classify cells, we choose the error rate for clustering and classification compared to the complete data as the evaluation metric for these thresholding schemes.

We present two case studies, the first on mouse cortex and hippocampus scRNA-Seq (Zeisel *et al.*, 2015; Ntranos *et al.*, 2016), and the second on mouse bone marrow scRNA-Seq (Paul *et al.*, 2015; Setty *et al.*, 2016). For the mouse cortex, the data matrix is cells $\times$ ECs, and only an incredibly small fraction of the ECs are necessary to obtain neuron and non-neuron cell clusters. For the mouse bone marrow, the data matrix is cells $\times$ genes, and only a small fraction of the genes are necessary to obtain HSPC, myeloid progenitor, and erythroid progenitor branch assignments. For both case studies, DCSS performs similarly to the simple thresholding schemes, in that it reduces the low abundance genes, maintains the most variable and over-dispersed genes. This supports our recommendation to use DCSS to control quality and technical variability. In both

case studies, the error rate between the clustering computed with the complete expression profile and the reduced expression profile is small, suggesting that the clustering algorithms rely on a small subset of informative features.

5. Software

The Python-package containing code to perform the methods described in the article can be found at `https://github.com/srmcc/dcss_single_cell.git`. The package also contains code to download the datasets used as examples in the article.

Acknowledgments

Research reported in this publication was supported by the National Human Genome Research Institute of the National Institutes of Health under Award Number [F32HG008713]. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health. SRM would like to acknowledge Ilan Shomorony and Robert Tunney for useful comments.

Conflict of Interest: None declared.

References

Bourgon, Richard, Gentleman, Robert and Huber, Wolfgang. (2010, May). Independent filtering increases detection power for high-throughput experiments. *Proceedings of the National Academy of Sciences of the United States of America* **107**(21), 9546–9551.

Brennecke, Philip, Anders, Simon, Kim, Jong Kyoung, Kołodziejczyk, Aleksandra A., Zhang, Xiuwei, Proserpio, Valentina, Baying, Bianka, Benes, Vladimir, Teichmann, Sarah A., Marioni, John C. *et al.* (2013, November). Accounting for technical noise in

single-cell RNA-seq experiments. *Nature Methods* **10**(11), 1093–1095.

Chatterjee, Samprit and Hadi, Ali S. (1986, August). Influential Observations, High Leverage Points, and Outliers in Linear Regression. *Statistical Science* **1**(3), 379–393.

Cohen, Michael B., Lee, Yin Tat, Musco, Cameron, Musco, Christopher, Peng, Richard and Sidford, Aaron. (2015). Uniform Sampling for Matrix Approximation. In: *Proceedings of the 2015 Conference on Innovations in Theoretical Computer Science*, ITCS '15. New York, NY, USA: ACM. pp. 181–190.

Cohen, Michael B., Musco, Cameron and Musco, Christopher. (2017). Input Sparsity Time Low-rank Approximation via Ridge Leverage Score Sampling. In: *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '17. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics. pp. 1758–1777.

Cox, D. R. and Lewis, P. A. W. (1966). *The statistical analysis of series of events*, Methuen's monographs on applied probability and statistics. London: Methuen.

Drineas, Petros, Mahoney, Michael W. and Muthukrishnan, S. (2006). Subspace sampling and relative-error matrix approximation: Column-based methods. In: *In Proc. of the 10th RANDOM*. pp. 316–326.

Eckart, C. and Young, G. (1936). The approximation of one matrix by another of lower rank. *Psychometrika* **1**, 211–218.

Horn, Roger A. and Johnson, Charles R. (2013). *Matrix analysis*, 2nd ed edition. New York: Cambridge University Press.

Jolliffe, I. T. (2002). *Principal component analysis*, 2nd ed edition., Springer series in statistics. New York: Springer.

Kwon, Haejoon, Fan, Jean and Kharchenko, Peter. (2017, January). Comparison of Principal Component Analysis and t-Stochastic Neighbor Embedding with Distance Metric Modifications for Single-cell RNA-sequencing Data Analysis. *bioRxiv*, 102780.

Lun, Aaron T. L., McCarthy, Davis J. and Marioni, John C. (2016). A step-by-step workflow for low-level analysis of single-cell RNA-seq data with Bioconductor. *F1000Research* **5**, 2122.

McCarthy, Davis J., Campbell, Kieran R., Lun, Aaron T. L. and Wills, Quin F. Scater: pre-processing, quality control, normalization and visualization of single-cell RNA-seq data in R. *Bioinformatics*.

Nicolae, Marius, Mangul, Serghei, Măndoiu, Ion I. and Zelikovsky, Alex. (2011). Estimation of alternative splicing isoform frequencies from RNA-Seq data. *Algorithms for molecular biology: AMB* **6**(1), 9.

Ntranos, Vasilis, Kamath, Govinda M., Zhang, Jesse M., Pachter, Lior and Tse, David N. (2016). Fast and accurate single-cell RNA-seq analysis by clustering of transcript-compatibility counts. *Genome Biology* **17**(1), 1–14.

Papailiopoulos, Dimitris, Kyrillidis, Anastasios and Boutsidis, Christos. (2014). Provable Deterministic Leverage Score Sampling. In: *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '14. New York, NY, USA: ACM. pp. 997–1006.

Paul, Franziska, Arkin, Ya'ara, Giladi, Amir, Jaitin, Diego Adhemar, Kenigsberg, Ephraim, Keren-Shaul, Hadas, Winter, Deborah, Lara-Astiaso, David, Gury, Meital, Weiner, Assaf, David, Eyal, Cohen, Nadav, Lauridsen, Felicia Kathrine Bratt, Haas, Simon, Schlitzer, Andreas, Mildner, Alexander, Ginhoux,

Florent, Jung, Steffen, Trumpp, Andreas, Porse, Bo Torben, Tanay, Amos *et al.* (2015, December). Transcriptional Heterogeneity and Lineage Commitment in Myeloid Progenitors. *Cell* **163**(7), 1663–1677.

Rao, C. Radhakrishna. (1973). *Linear statistical inference and its applications*, 2d ed edition., Wiley series in probability and mathematical statistics. New York: Wiley.

Satija, Rahul, Farrell, Jeffrey A., Gennert, David, Schier, Alexander F. and Regev, Aviv. (2015, May). Spatial reconstruction of single-cell gene expression data. *Nature Biotechnology* **33**(5), 495–502.

Setty, Manu, Tadmor, Michelle D., Reich-Zeliger, Shlomit, Angel, Omer, Salame, Tomer Meir, Kathail, Pooja, Choi, Kristy, Bendall, Sean, Friedman, Nir and Pe'er, Dana. (2016, June). Wishbone identifies bifurcating developmental trajectories from single-cell data. *Nature Biotechnology* **34**(6), 637–645.

Stegle, Oliver, Teichmann, Sarah A. and Marioni, John C. (2015, March). Computational and analytical challenges in single-cell transcriptomics. *Nature Reviews. Genetics* **16**(3), 133–145.

Tang, Fuchou, Barbacioru, Catalin, Wang, Yangzhou, Nordman, Ellen, Lee, Clarence, Xu, Nanlan, Wang, Xiaohui, Bodeau, John, Tuch, Brian B., Siddiqui, Asim, Lao, Kaiqin *et al.* (2009, May). mRNA-Seq whole-transcriptome analysis of a single cell. *Nature Methods* **6**(5), 377–382.

The Gene Ontology Consortium. (2015, January). Gene Ontology Consortium: going forward. *Nucleic Acids Research* **43**(D1), D1049–D1056.

Trapnell, Cole, Cacchiarelli, Davide, Grimsby, Jonna, Pokharel, Prapti, Li, Shuqiang, Morse, Michael, Lennon, Niall J., Livak, Kenneth J., Mikkelsen, Tar-

JEI S. AND RINN, JOHN L. (2014, April). The dynamics and regulators of cell fate decisions are revealed by pseudotemporal ordering of single cells. *Nature Biotechnology* **32**(4), 381–386.

TROPP, JOEL A. (2009). Column Subset Selection, Matrix Factorization, and Eigenvalue Optimization. In: *Proceedings of the Twentieth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '09. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics. pp. 978–986.

TROPP, JOEL A. (2015, May). An Introduction to Matrix Concentration Inequalities. *Found. Trends Mach. Learn.* **8**(1-2), 1–230.

VELLEMAN, PAUL F. AND WELSCH, ROY E. (1981). Efficient Computing of Regression Diagnostics. *The American Statistician* **35**(4), 234–242.

VIB / UGENT BIOINFORMATICS AND EVOLUTIONARY GENOMICS. Calculate and draw custom Venn diagrams: http://bioinformatics.psb.ugent.be/webtools/Venn/.

WAGNER, ALLON, REGEV, AVIV AND YOSEF, NIR. (2016, November). Revealing the vectors of cellular identity with single-cell genomics. *Nature Biotechnology* **34**(11), 1145–1160.

ZEISEL, AMIT, MUÑOZ-MANCHADO, ANA B., CODELUPPI, SIMONE, LÖNNERBERG, PETER, MANNO, GIOELE LA, JURÉUS, ANNA, MARQUES, SUELI, MUNGUBA, HERMANY, HE, LIQUN, BETSHOLTZ, CHRISTER, ROLNY, CHARLOTTE, CASTELO-BRANCO, GONÇALO, HJERLING-LEFFLER, JENS *et al.* (2015, March). Cell types in the mouse cortex and hippocampus revealed by single-cell RNA-seq. *Science* **347**(6226), 1138–1142.

6. Appendix

6.1 *Brief linear algebra review (Horn and Johnson, 2013)*

The *singular value decomposition* (SVD) of any complex matrix $\mathbf{A}$ is $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\dagger$, where $\mathbf{U}$ and $\mathbf{V}$ are square unitary matrices ($\mathbf{U}^\dagger\mathbf{U} = \mathbf{U}\mathbf{U}^\dagger = \mathbf{I}, \mathbf{V}^\dagger\mathbf{V} = \mathbf{V}\mathbf{V}^\dagger = \mathbf{I}$), $\mathbf{\Sigma}$ is a rectangular diagonal matrix with real non-negative non-increasingly ordered entries. $\mathbf{U}^\dagger$ is the complex conjugate and transpose of $\mathbf{U}$, and $\mathbf{I}$ is the identity matrix. The diagonal elements of $\mathbf{\Sigma}$ are called the *singular values*, and they are the positive square roots of the eigenvalues of both $\mathbf{A}\mathbf{A}^\dagger$ and $\mathbf{A}^\dagger\mathbf{A}$, which have eigenvectors $\mathbf{U}$ and $\mathbf{V}$, respectively. $\mathbf{U}$ and $\mathbf{V}$ are the *left* and *right singular vectors* of $\mathbf{A}$.

Defining $\mathbf{U}_k$ as the first k columns of $\mathbf{U}$ and analogously for $\mathbf{V}$, and $\mathbf{\Sigma}_k$ the square diagonal matrix with the first k entries of $\mathbf{\Sigma}$, then $\mathbf{A}_k = \mathbf{U}_k\mathbf{\Sigma}_k\mathbf{V}_k^\dagger$ is the rank-k SVD approximation to $\mathbf{A}$, and $\mathbf{T} = \mathbf{A}\mathbf{V}_k = \mathbf{U}_k\mathbf{\Sigma}_k$ is a rank-k SVD truncation of $\mathbf{A}$. Furthermore, we refer to matrix with only the last $n-k$ columns of $\mathbf{U}, \mathbf{V}$ and last $n-k$ entries in $\mathbf{\Sigma}$ as $\mathbf{U}_{\setminus k}, \mathbf{V}_{\setminus k}$, and $\mathbf{\Sigma}_{\setminus k}$.

The Moore-Penrose pseudo inverse of a rank r matrix $\mathbf{A}$ is given by $\mathbf{A}^+ = \mathbf{U}_r\mathbf{\Sigma}_r^{-1}\mathbf{V}_r^\dagger$.

The Frobenius norm $||\mathbf{A}||_F$ of a matrix $\mathbf{A}$ is given by $||\mathbf{A}||_F = \sqrt{\mathrm{tr}\,(\mathbf{A}\mathbf{A}^\dagger)}$. The spectral norm $||\mathbf{A}||_2$ of a matrix $\mathbf{A}$ is given by the largest singular value of $\mathbf{A}$.

The Eckart-Young-Mirsky theorem (Eckart and Young, 1936) states that, for $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\dagger$ the SVD of $\mathbf{A}$, and $\mathbf{B}$ any complex matrix with compatible dimension to $\mathbf{A}$ and rank $\leqslant k$,

$$\begin{aligned} \mathbf{A}_k &= \mathrm{argmin}_{\mathrm{rank}(\mathbf{B})\leqslant k}||\mathbf{A}-\mathbf{B}||_F \\ \min_{\mathrm{rank}(\mathbf{B})\leqslant k} ||\mathbf{A}-\mathbf{B}||_F &= \sqrt{\mathrm{tr}\,(\mathbf{\Sigma}_{\setminus k}\mathbf{\Sigma}_{\setminus k}^T)}. \end{aligned} \tag{6.13}$$

The minimizer $\mathbf{A}_k$ is unique if and only if $\sigma_{k+1} \neq \sigma_k$, where σ_i are the respective non-increasingly ordered singular values in $\mathbf{\Sigma}$.

A square complex matrix $\mathbf{F}$ is *Hermitian* if $\mathbf{F} = \mathbf{F}^\dagger$. Symmetric positive semi-definite (S.P.S.D) matrices are Hermitian matrices. The set of $n \times n$ Hermitian matrices is a real linear space. As such, it is possible to define a *partial ordering* (also called a Loewner partial ordering, denoted

by $\preceq$) on the real linear space. One matrix is "greater" than another if their difference lies in the closed convex cone of S.P.S.D. matrices. Let $\mathbf{F}, \mathbf{G}$ be Hermitian and the same size, and $\mathbf{x}$ a complex vector of compatible dimension. Then,

$$\mathbf{F} \preceq \mathbf{G} \iff \mathbf{x}^\dagger \mathbf{F} \mathbf{x} \leqslant \mathbf{x}^\dagger \mathbf{G} \mathbf{x} \quad \forall \mathbf{x} \neq \mathbf{0}. \tag{6.14}$$

A few simple consequences of the Loewner partial ordering are as follows. If $\mathbf{F}$ is Hermitian and S.P.S.D., then $\mathbb{0} \preceq \mathbf{F}$, where $\mathbb{0}$ is the zero matrix.

If $\mathbf{F}$ is Hermitian with smallest and largest eigenvalues $\lambda_{\min}(\mathbf{F}), \lambda_{\max}(\mathbf{F})$, respectively, then,

$$\lambda_{\min}(\mathbf{F})\mathbf{I} \preceq \mathbf{F} \preceq \lambda_{\max}(\mathbf{F})\mathbf{I}. \tag{6.15}$$

Let $\mathbf{F}, \mathbf{G}$ be Hermitian and the same size, and let $\mathbf{H}$ be any complex rectangular matrix of compatible dimension. The *conjugation rule* is,

$$\text{If } \mathbf{F} \preceq \mathbf{G}, \text{ then } \mathbf{H}\mathbf{F}\mathbf{H}^\dagger \preceq \mathbf{H}\mathbf{G}\mathbf{H}^\dagger. \tag{6.16}$$

In addition, let $\lambda_i(\mathbf{F})$ and $\lambda_i(\mathbf{G})$ be the non-decreasingly ordered eigenvalues of $\mathbf{F}, \mathbf{G}$. Then,

$$\text{If } \mathbf{F} \preceq \mathbf{G}, \text{ then } \forall i,\ \lambda_i(\mathbf{F}) \leqslant \lambda_i(\mathbf{F}). \tag{6.17}$$

Since the trace of a matrix $\mathbf{F}$ is the sum of its eigenvalues, $\operatorname{tr}\mathbf{F} = \sum_i \lambda_i(\mathbf{F})$, and the Loewner ordering implies the ordering of eigenvalues (Eqn. 6.17), the Loewner ordering also implies the ordering of their sum,

$$\text{If } \mathbf{F} \preceq \mathbf{G}, \text{ then } \operatorname{tr}\mathbf{F} \leqslant \operatorname{tr}\mathbf{G}. \tag{6.18}$$

Let $\mathbf{F}_1, \mathbf{G}_1, \mathbf{F}_2, \mathbf{G}_2$ be Hermitian and the same size. Then if $\mathbf{F}_1 \preceq \mathbf{G}_1$ and $\mathbf{F}_2 \preceq \mathbf{G}_2$, then

$$\mathbf{F}_1 + \mathbf{F}_2 \preceq \mathbf{G}_1 + \mathbf{G}_2. \tag{6.19}$$

As a simple consequence of Eqn. 6.14, consider the real matrices $\mathbf{F}\mathbf{F}^T$ and $\mathbf{G}\mathbf{G}^T$, and the vector $\mathbf{x}$ which has a one in row i and a minus one in row j, and zeros elsewhere. The Euclidean

distance between rows i, j with respect to $\mathbf{G}$ is $d_{i,j}(\mathbf{G})$:

$$d_{i,j}(\mathbf{G}) = \mathbf{x}^T \mathbf{G}\mathbf{G}^T \mathbf{x}. \tag{6.20}$$

Thus, if $\mathbf{F}\mathbf{F}^T \preceq \mathbf{G}\mathbf{G}^T$, by Eqn. 6.14 with appropriate vectors, $d_{i,j}(\mathbf{F}) \leqslant d_{i,j}(\mathbf{G}) \forall i, j$.

Furthermore, let $\mathbf{F}$ be Hermitian and dimension n, $\mathbf{U}_k$ be a semi-orthogonal rectangular matrix ($\mathbf{U}_k^\dagger \mathbf{U}_k = \mathbf{I}$) of compatible dimension $n \times k$, $1 \leqslant k \leqslant n$, and $\lambda_i(\mathbf{M})$ refer to the non-decreasingly ordered eigenvalues of a matrix $\mathbf{M}$. Then the upper bound of the *Poincaré separation theorem* states,

$$\lambda_i(\mathbf{U}_k^\dagger \mathbf{F} \mathbf{U}_k) \preceq \lambda_{n-k+i}(\mathbf{F}) \quad i = 1, \ldots, k. \tag{6.21}$$

We will also use the von Neumann trace inequality. Let $\mathbf{F}, \mathbf{G}$ be complex matrices of compatible dimension and minimum dimension n. Let $\sigma_i(\mathbf{F}), \sigma_i(\mathbf{G})$ be the respective non-increasingly ordered singular values. Then

$$Re(\operatorname{tr} \mathbf{F}\mathbf{G}^\dagger) \leqslant \sum_{i=1}^{n} \sigma_i(\mathbf{F}) \sigma_i(\mathbf{G}). \tag{6.22}$$

6.2 *Proof of Bound 2.9*

The upper bound (Bound 2.9) in Theorem 2.1 follows from the fact that $\mathbb{0} \preceq \mathbf{I} - \mathbf{S}\mathbf{S}^T$ and the conjugation rule (Eqn. 6.16),

$$\mathbb{0} \preceq \mathbf{A}(\mathbf{I} - \mathbf{S}\mathbf{S}^T)\mathbf{A}^T = \mathbf{A}\mathbf{A}^T - \mathbf{C}\mathbf{C}^T. \tag{6.23}$$

This upper bound is true for any column selection of $\mathbf{A}$. A second application of the conjugation rule gives the upper bound in Bound 2.9.

For the lower bound (Bound 2.9), consider the quantity $\mathbf{Y} = \mathbf{\Sigma}_k^{-1} \mathbf{U}_k^T \mathbf{A}(\mathbf{I} - \mathbf{S}\mathbf{S}^T)\mathbf{A}^T \mathbf{U}_k \mathbf{\Sigma}_k^{-1} = \mathbf{V_k}^T (\mathbf{I} - \mathbf{S}\mathbf{S}^T)\mathbf{V_k}$. By the conjugation rule (Eqn. 6.16) on Eqn. 6.23, $\mathbb{0} \preceq \mathbf{Y}$, so $\mathbf{Y}$ is S.P.S.D. By the construction of DCSS (Eqn. 2.3) $\operatorname{tr} \mathbf{Y} = \sum_{i \notin \Theta} \sum_{l=1}^{k} V_{il}^2 = \tilde{\epsilon} < \epsilon$, and because $\mathbf{Y}$ is S.P.S.D.,

$\lambda_{\max}(\mathbf{Y}) \leqslant \operatorname{tr}\mathbf{Y}$. By Eqn. 6.15 and the previous facts, $\mathbf{Y} \preceq \lambda_{\max}(\mathbf{Y})\mathbf{I} \preceq \epsilon\mathbf{I}$. As a result of the conjugation rule applied to this upper bound,

$$\mathbf{U}_k\mathbf{\Sigma}_k\mathbf{Y}\mathbf{\Sigma}_k\mathbf{U}_k^T = \mathbf{A}_k\mathbf{A}_k^T - \mathbf{U}_k\mathbf{U}_k^T\mathbf{C}\mathbf{C}^T\mathbf{U}_k\mathbf{U}_k^T \preceq \epsilon\mathbf{A}_k\mathbf{A}_k^T$$
$$(1-\epsilon)\mathbf{A}_k\mathbf{A}_k^T \preceq \mathbf{U}_k\mathbf{U}_k^T\mathbf{C}\mathbf{C}^T\mathbf{U}_k\mathbf{U}_k^T, \tag{6.24}$$

providing the lower bound of Bound 2.9.

For Bound 2.10, the lower bound of Bound 2.9 implies,

$$(1-\epsilon)\operatorname{tr}\mathbf{A}_k\mathbf{A}_k^T \leqslant \operatorname{tr}\mathbf{U}_k^T\mathbf{C}\mathbf{C}^T\mathbf{U}_k, \tag{6.25}$$

by Eqn. 6.18 and the cyclic property of the trace. Similarly, Eqn. 6.23 implies $\operatorname{tr}\mathbf{C}\mathbf{C}^T \leqslant \operatorname{tr}\mathbf{A}\mathbf{A}^T$. Since $\mathbf{U}_k$ is semi-orthogonal ($\mathbf{U}_k^T\mathbf{U}_k = \mathbf{I}$), by Eqn. 6.21, every ordered eigenvalue of $\mathbf{U}_k^T\mathbf{C}\mathbf{C}^T\mathbf{U}_k$ is smaller than its counterpart ordered eigenvalue of $\mathbf{C}\mathbf{C}^T$. Since the trace is the sum of eigenvalues, this implies Bound 2.10,

$$(1-\epsilon)\operatorname{tr}\mathbf{A}_k\mathbf{A}_k^T \leqslant \operatorname{tr}\mathbf{U}_k^T\mathbf{C}\mathbf{C}^T\mathbf{U}_k \leqslant \operatorname{tr}\mathbf{C}\mathbf{C}^T \leqslant \operatorname{tr}\mathbf{A}\mathbf{A}^T. \tag{6.26}$$

Note that if $\mathbf{A}$ is full rank and $k = rank(\mathbf{A}) = n$, then Bound 2.9 becomes,

$$(1-\epsilon)\mathbf{A}\mathbf{A}^T \preceq \mathbf{C}\mathbf{C}^T \preceq \mathbf{A}\mathbf{A}^T. \tag{6.27}$$

6.3 *Proof of Bound 2.11 for random sampling.*

The following theorem pertains to a new spectral bound for the square $\mathbf{C}$ selected by rank-k subspace leverage scores and the random sampling procedure from Drineas, Mahoney and Muthukrishnan (2006).

THEOREM 6.1 Let $\mathbf{A} \in \mathbb{R}^{n\times d}$ be a matrix of at least rank k and $\tau_i(\mathbf{A}_k)$ be defined as in Eqn. 1.1. Construct $\mathbf{C}$ by sampling t columns of $\mathbf{A}$, reweighted to $\frac{1}{\sqrt{tp_i}}\mathbf{a}_i$, with probability $p_i = (\tau_i(\mathbf{A}_k)+\gamma\mathbb{1}(\tau_i(\mathbf{A}_k)=0))/(\sum_{i=1}^d p_i)$, where $\mathbb{1}()$ is the indicator function and γ is a small, positive,

non-zero number $\gamma = \min_{\tau_i(\mathbf{A}_k)>0}(\tau_i(\mathbf{A}_k))$. Let $m = \sum_{i=1}^{d} \mathbb{1}(\tau_i(\mathbf{A}_k) = 0)$, $\sum_{i=1}^{d} p_i = k + m\gamma$. If the number of selected columns $t \geqslant \frac{2}{\epsilon^2}(k + m\gamma)\left(1 + \frac{1}{3}\epsilon\right)\ln\left(\frac{16k}{\delta}\right)$, then with probability $1 - \delta$, the matrix $\mathbf{C}$ satisfies:

$$(1-\epsilon)\mathbf{A}_k\mathbf{A}_k^T \preceq \mathbf{U}_k\mathbf{U}_k^T\mathbf{C}\mathbf{C}^T\mathbf{U}_k\mathbf{U}_k^T \preceq (1+\epsilon)\mathbf{A}_k\mathbf{A}_k^T. \tag{6.28}$$

If $\mathbf{A}$ is full rank and $k = rank(\mathbf{A}) = n$, then Bound 6.28 becomes,

$$(1-\epsilon)\mathbf{A}\mathbf{A}^T \preceq \mathbf{C}\mathbf{C}^T \preceq (1+\epsilon)\mathbf{A}\mathbf{A}^T. \tag{6.29}$$

The proof of Theorem 6.1 is similar in structure to Theorem 3 in Cohen, Musco and Musco (2017). Theorem 3 in Cohen, Musco and Musco (2017) pertains to a different type of leverage score.

Consider the quantity $\mathbf{Y} = \mathbf{\Sigma}_k^{-1}\mathbf{U}_k^T(\mathbf{C}\mathbf{C}^T - \mathbf{A}\mathbf{A}^T)\mathbf{U}_k\mathbf{\Sigma}_k^{-1}$. Note the sign change from Sec. 6.2. This can be rewritten as,

$$\begin{aligned}
\mathbf{Y} &= \sum_{j=1}^{t} \mathbf{\Sigma}_k^{-1}\mathbf{U}_k^T(\mathbf{c}_j\mathbf{c}_j^T - \tfrac{1}{t}\mathbf{A}\mathbf{A}^T)\mathbf{U}_k\mathbf{\Sigma}_k^{-1} \\
\mathbf{Y} &= \sum_{j=1}^{t} \mathbf{X}_j, \\
\forall j, (\mathbf{X}_j)_i &= \tfrac{1}{t}\mathbf{\Sigma}_k^{-1}\mathbf{U}_k^T(\tfrac{1}{p_i}\mathbf{a}_i\mathbf{a}_i^T - \mathbf{A}\mathbf{A}^T)\mathbf{U}_k\mathbf{\Sigma}_k^{-1} \quad \text{with categorical probability } p_i.
\end{aligned} \tag{6.30}$$

If $||\mathbf{Y}||_2 \leqslant \epsilon$, then $-\epsilon\mathbf{I} \preceq \mathbf{Y} \preceq \epsilon\mathbf{I}$, and Bound 6.28 follows from this and the definition of $\mathbf{Y}$. Thus, the proof of Bound 6.28 relies on showing that $||\mathbf{Y}||_2 \leqslant \epsilon$. We use an intrinsic dimension matrix Bernstein inequality ((Tropp, 2015) , Theorem 7.3.1), specialized to Hermitian matrices, to show that $||\mathbf{Y}||_2$ is small with high probability. The Bernstein inequality requires that, for a finite sequence $\mathbf{Y} = \sum_{j=1}^{t} \mathbf{X}_j$ of random Hermitian matrices $\mathbf{X}_j$ of the same size,

1. $\forall j, \mathbb{E}(\mathbf{X}_j) = 0$,
2. $\forall j, ||\mathbf{X}_j||_2 \leqslant L$,
3. and that $\sum_j \mathbb{E}(\mathbf{X}_j\mathbf{X}_j^T) \preceq \mathbf{V}$.

Then, for $\epsilon \geqslant \sqrt{||\mathbf{V}||_2} + L/3$,

$$\mathbf{P}(||\mathbf{Y}||_2 \geqslant \epsilon) \leqslant 8\frac{\mathrm{tr}\,\mathbf{V}}{||\mathbf{V}||_2}\exp\left(-\frac{\epsilon^2/2}{\epsilon L/3+||\mathbf{V}||_2}\right). \tag{6.31}$$

Requirement 1 is satisfied because,

$$\mathbb{E}(\mathbf{X}_j) = \sum_{i=1}^{d} p_i(\mathbf{X}_j)_i = \tfrac{1}{t}\mathbf{\Sigma}_k^{-1}\mathbf{U}_k^T(\sum_{j=1}^{d}\mathbf{a}_i\mathbf{a}_i^T - \mathbf{A}\mathbf{A}^T)\mathbf{U}_k\mathbf{\Sigma}_k^{-1} = 0. \tag{6.32}$$

To show that requirement 2 is satisfied, we need the following fact:

$$\mathbf{\Sigma}_k^{-1}\mathbf{U}_k^T\mathbf{a}_i\mathbf{a}_i^T\mathbf{U}_k\mathbf{\Sigma}_k^{-1} \preceq \tau_i(\mathbf{A}_k)\mathbf{I}. \tag{6.33}$$

Eqn. 6.33 follows from the fact that for all $\mathbf{y} \in \mathbb{R}^k$,

$$\mathbf{y}^T\mathbf{U}_k\mathbf{\Sigma}_k^{-1}\mathbf{U}_k^T\mathbf{a}_i\mathbf{a}_i^T\mathbf{U}_k\mathbf{\Sigma}_k^{-1}\mathbf{U}_k^T\mathbf{y} = \mathrm{tr}\left(\left(\mathbf{y}\mathbf{y}^T\right)\left(\mathbf{U}_k\mathbf{\Sigma}_k^{-1}\mathbf{U}_k^T\mathbf{a}_i\mathbf{a}_i^T\mathbf{U}_k\mathbf{\Sigma}_k^{-1}\mathbf{U}_k^T\right)\right) \leqslant \tau_i(\mathbf{A}_k)\mathbf{y}^T\mathbf{y}.$$

where the inequality comes from the Von Neumann trace inequality (Eqn. 6.22) applied to the product of two rank 1 matrices. Using eqn. 6.33 in the definition of $\mathbf{X}_i$ gives,

$$\begin{aligned}\mathbf{X}_j = \tfrac{1}{tp_i}\mathbf{\Sigma}_k^{-1}\mathbf{U}_k^T\mathbf{a}_i\mathbf{a}_i^T\mathbf{U}_k\mathbf{\Sigma}_k^{-1} - \tfrac{1}{t}\mathbf{I} &\preceq \tfrac{1}{tp_i}\tau_i(\mathbf{A}_k)\mathbf{I} - \tfrac{1}{t}\mathbf{I} \\ &= \tfrac{(k+m\gamma)\tau_i(\mathbf{A}_k)}{t(\tau_i(\mathbf{A}_k)+\gamma\mathbb{1}(\tau_i(\mathbf{A}_k)=0))}\mathbf{I} - \tfrac{1}{t}\mathbf{I} \\ &\preceq \tfrac{k+m\gamma}{t}\mathbf{I},\end{aligned} \tag{6.34}$$

and $||\mathbf{X}_j||_2 \leqslant L = \frac{k+m\gamma}{t}$ follows immediately.

To show that requirement 3 is satisfied, we compute directly,

$$\begin{aligned}\mathbb{E}(\mathbf{Y}^2) &= t\mathbb{E}(\mathbf{X}_j\mathbf{X}_j^T) \\ &= t\sum_{i=1}^{d} p_i\left(\left(\tfrac{1}{t}\mathbf{\Sigma}_k^{-1}\mathbf{U}_k^T(\tfrac{1}{p_i}\mathbf{a}_i\mathbf{a}_i^T - \mathbf{A}\mathbf{A}^T)\mathbf{U}_k\mathbf{\Sigma}_k^{-1}\right)\left(\tfrac{1}{t}\mathbf{\Sigma}_k^{-1}\mathbf{U}_k^T(\tfrac{1}{p_i}\mathbf{a}_i\mathbf{a}_i^T - \mathbf{A}\mathbf{A}^T)\mathbf{U}_k\mathbf{\Sigma}_k^{-1}\right)\right) \\ &= t\sum_{i=1}^{d} p_i\left(\left(\tfrac{1}{t}\mathbf{\Sigma}_k^{-1}\mathbf{U}_k^T(\tfrac{1}{p_i}\mathbf{a}_i\mathbf{a}_i^T - \mathbf{A}\mathbf{A}^T)\mathbf{U}_k\mathbf{\Sigma}_k^{-1}\right)\left(\tfrac{1}{tp_i}\mathbf{\Sigma}_k^{-1}\mathbf{U}_k^T\mathbf{a}_i\mathbf{a}_i^T\mathbf{U}_k\mathbf{\Sigma}_k^{-1}\right)\right) \\ &= t\sum_{i=1}^{d} p_i\left(\tfrac{1}{t^2p_i^2}\mathbf{\Sigma}_k^{-1}\mathbf{U}_k^T\mathbf{a}_i\mathbf{a}_i^T\mathbf{U}_k\mathbf{\Sigma}_k^{-2}\mathbf{U}_k^T\mathbf{a}_i\mathbf{a}_i^T\mathbf{U}_k\mathbf{\Sigma}_k^{-1}\right) - \tfrac{1}{t}\mathbf{I} \\ &\preceq \sum_{i=1}^{d}\left(\tfrac{1}{tp_i}\mathbf{\Sigma}_k^{-1}\mathbf{U}_k^T\mathbf{a}_i\mathbf{a}_i^T\mathbf{U}_k\mathbf{\Sigma}_k^{-1}\tau_i(\mathbf{A}_k)\mathbf{I}\right) - \tfrac{1}{t}\mathbf{I}\end{aligned}$$

$$\preceq \tfrac{k+m\gamma}{t} \sum_{i=1}^{d} \left(\mathbf{\Sigma}_k^{-1} \mathbf{U}_k^T \mathbf{a}_i \mathbf{a}_i^T \mathbf{U}_k \mathbf{\Sigma}_k^{-1} \right) = \tfrac{k+m\gamma}{t} \mathbf{I} = \mathbf{V}. \tag{6.35}$$

It follows immediately that $||\mathbf{V}||_2 = \frac{k+m\gamma}{t}$ and $\operatorname{tr} \mathbf{V} = \frac{k(k+m\gamma)}{t}$.

Then, for $\epsilon \geqslant \sqrt{\frac{k+m\gamma}{t}} + \frac{k+m\gamma}{3t}$,

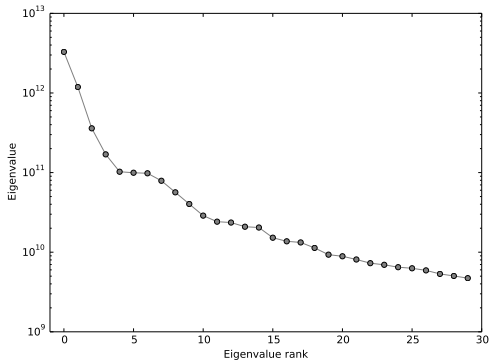
$$\mathbf{P}(||\mathbf{Y}||_2 \geqslant \epsilon) \leqslant 8k \exp\left(-\tfrac{t\epsilon^2/2}{(k+m\gamma)(\epsilon/3+1)} \right) \leqslant \tfrac{1}{2}\delta. \tag{6.36}$$

Solving for t as a function of ϵ, δ, and γ gives,

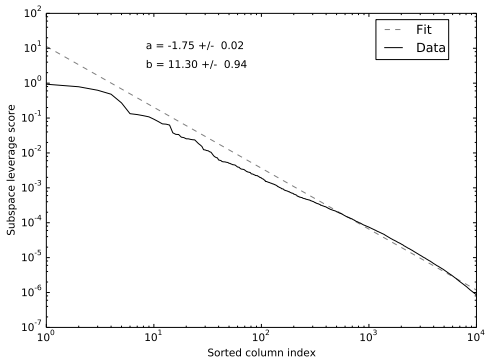
$$t \geqslant \tfrac{2}{\epsilon^2}(k + m\gamma)\left(1 + \tfrac{1}{3}\epsilon\right) \ln\left(\tfrac{16k}{\delta}\right). \tag{6.37}$$

Bound 6.28 also holds for $\mathbf{C}$ selected by the DCSS algorithm, as a consequence of Bound 2.9. Thus DCSS selects fewer columns with the same accuracy for power-law decay for Bound 6.28 when $|\Theta| < t$.

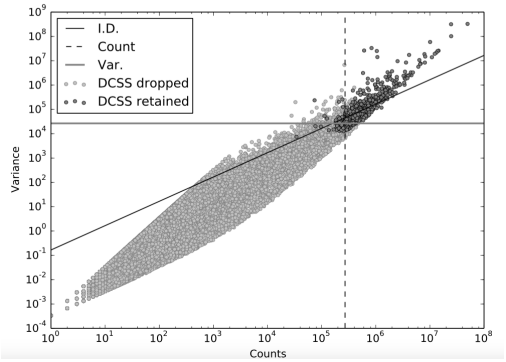
7. Figures and Tables



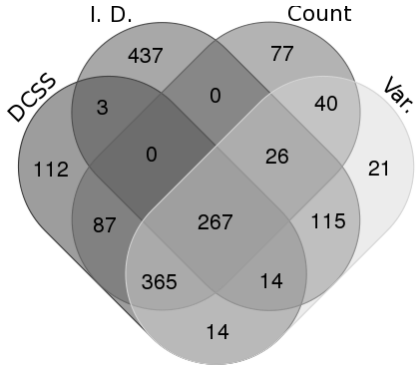
(a) Eigenvalues for $\mathbf{A}\mathbf{A}^T$. The first "elbow" occurs at the fifth largest eigenvalue.



(b) Power-law decay of $k = 5$ subspace leverage scores with index. The fit is to Score = b $\times$ (Index) ^{a}.

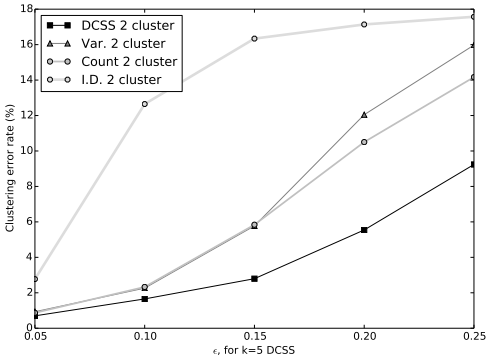


(c) Count-Variance plot for each column of $\mathbf{A}$. The color for each column represents whether the column is selected or not by $k = 5, \epsilon = 0.1$ DCSS. The plot also shows the thresholds for count, variance, and index of dispersion with same number of selected columns as DCSS.

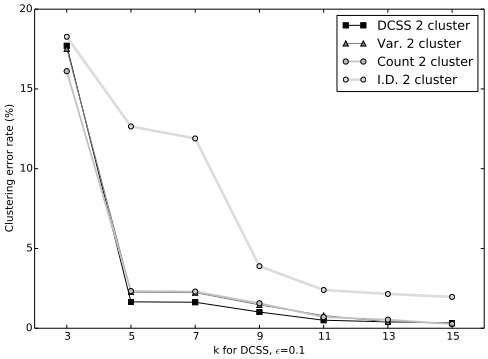


(d) Venn diagram comparing the overlap between selected columns between $k = 5, \epsilon = 0.1$ DCSS, count, variance, and index of dispersion thresholding. Figure tool credit: VIB / UGent Bioinformatics and Evolutionary Genomics.

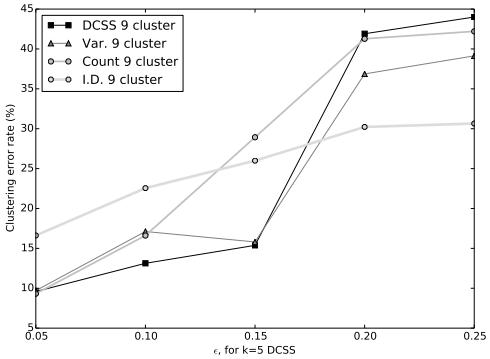
Fig. 1: Figures for the Zeisel *et al.* (2015) and Ntranos *et al.* (2016) dataset.



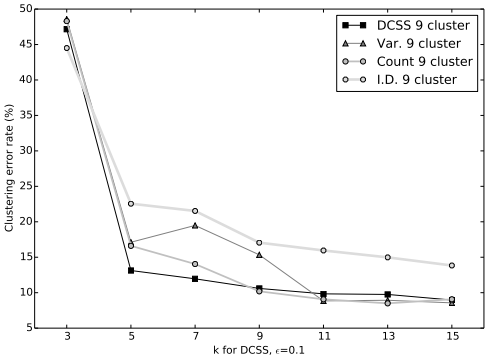
(a) Clustering error rate for two clusters, varying ϵ with $k = 5$ for DCSS.



(b) Clustering error rate for two clusters, varying k with $\epsilon = 0.1$ for DCSS.

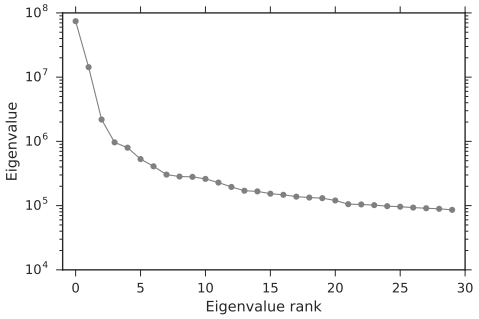


(c) Clustering error rate for nine clusters, varying ϵ with $k = 5$ for DCSS.

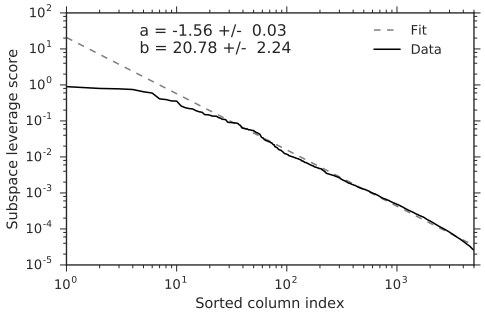


(d) Clustering error rate for nine clusters, varying k with $\epsilon = 0.1$ for DCSS.

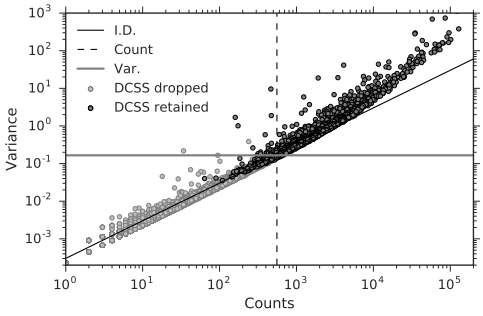
Fig. 2: Average spectral clustering error for two and nine clusters for DCSS, count, variance, and index of dispersion thresholding for the Zeisel *et al.* (2015) and Ntranos *et al.* (2016) dataset.



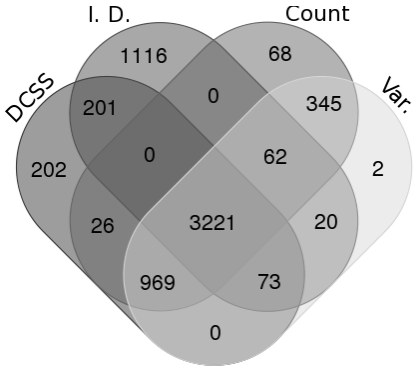
(a) Eigenvalues for $\mathbf{A}\mathbf{A}^T$. "Elbows" are not as apparent as in Fig. 1a. We choose the elbow at the fourteenth eigenvalue due to the sensitivity of the diffusion component GSEA enrichment analysis.



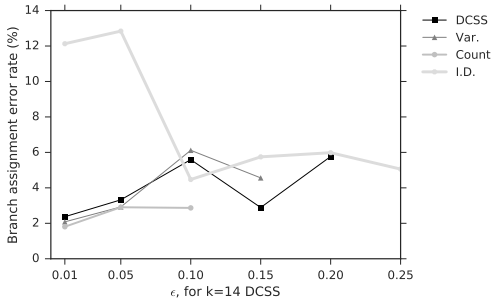
(b) Power-law decay of $k = 14$ subspace leverage scores with index. The fit is to Score = b × (Index) ^{a}.



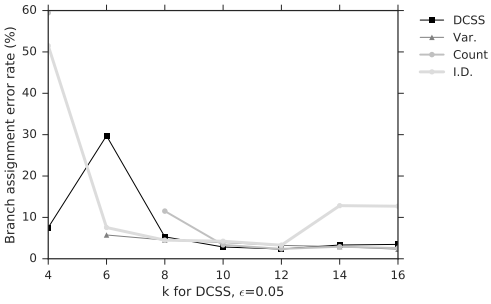
(c) Count-Variance plot for each column of $\mathbf{A}$. The color for each column represents whether the column is selected or not by $k = 14, \epsilon = 0.05$ DCSS. The plot also shows the thresholds for count, variance, and index of dispersion with same number of selected columns as DCSS.



(d) Venn diagram comparing the overlap between selected columns between $k = 14, \epsilon = 0.05$ DCSS, count, variance, and index of dispersion thresholding. Figure tool credit: VIB / UGent Bioinformatics and Evolutionary Genomics.



(e) Branch assignment error rate, varying ϵ with $k = 14$ for DCSS.



(f) Branch assignment error rate, varying k with $\epsilon = 0.05$ for DCSS

Fig. 3: Figures for the Paul *et al.* (2015) and Setty *et al.* (2016) dataset.

Table 1: PANTHER overrepresentation test (release 20160715) with the GO Ontology database (release 2016-08-22) for the $k = 5, \epsilon = 0.1$ DCSS 862 ECs mapped to $1,642$ genes.

Type	Gene ontology (GO) term	Bonferroni p-value
Biological process	cellular component organization (GO:0016043)	1.12E-02
Biological process	cellular component organization or biogenesis (GO:0071840)	8.01E-03
Biological process	localization (GO:0051179)	4.37E-02
Cellular component	neuron projection (GO:0043005)	4.52E-04
Cellular component	neuron part (GO:0097458)	8.24E-05
Cellular component	cell projection (GO:0042995)	8.36E-03
Cellular component	cytoplasm (GO:0005737)	1.59E-02
Cellular component	intracellular part (GO:0044424)	4.89E-02
Molecular function	enzyme binding (GO:0019899)	3.35E-02
Molecular function	olfactory receptor activity (GO:0004984)	1.30E-02