Comparison of clustering methods for single-cell RNA sequencing data

---

Master Thesis in Biostatistics (STA495)

by

Angelo Duò
11-297-777

supervised by

Prof. Dr. Mark D. Robinson
Dr. Charlotte Soneson

Zurich, February 2018

Abstract

Single-cell RNA sequencing (scRNA-seq) is increasingly used for the characterization of cells. The characterization of the composition of cell types in a sample is one of the most common aims for scRNA-seq data. In recent years, several methods have been developed for the clustering of scRNA-seq data but so far, there has been no comparison study that has independently tested the various methods. The aim of this study is to evaluate the performance of the various clustering methods for scRNA-seq data, using publicly available datasets and simulations. Due to the high dimensionality and large technical variation of scRNA-seq data, filtering and normalisation are crucial steps in the clustering workflow. Here, we want to investigate the effect of filtering on performance using gene-wise filtered and unfiltered datasets. Another aim is to test the various methods in their default mode by using the default settings of the methods. Additionally, this study will investigate the stability and runtime. Several well-performing methods have been found such as SC3, Seurat, pcaReduce, CIDR, and SIMLR. pcaReduce suffers from instability, while CIDR is not as flexible regarding the type of expression values.

Contents

1 Introduction

Cells are one of the fundamental units of life. They show an immense complexity and diversity. Their identity and function is determined by environmental stimuli, the physical environment, the cell cycle and neighbouring cells (Wagner et al., 2016). Only recently has it been possible to investigate the full transcriptome of a single cell. Single-cell RNA sequencing (scRNA-seq) was first published by Tang et al. (2009). This method addresses new biological issues, such as the identification of rare cell populations, and allows us to measure the frequency of cell types in tissues, characterise differences in similar cell types and investigate the heterogeneity of cell states or cell lineages (Andrews and Hemberg, 2017).

A typical scRNA-seq workflow consists of the isolation of single cells, the extraction of RNA, cDNA library preparation, and the amplification and sequencing of the libraries (see Figure 1). A wide variety of scRNA-seq protocols exists, differing in throughput, full transcript or 3' sequencing, costs and automatisation. Small-scale protocols are standard PCR plate-based methods or methods in which cell isolation and library preparation are combined into one protocol. A typical small-scale method is the PCR plate-based SMART-seq2 (Picelli et al., 2013). Full transcripts are sequenced using a standard Illumina approach. Typically, hundreds of cells are processed and spike-ins from the External RNA Control Consortium (ERCC) are used for normalisation. On the other side of the spectrum are droplet-based methods such as Drop-seq and 10xChromium, which allow the processing of thousands of cells.

The differences between scRNA-seq and bulk experiments are the lower sequencing depth (100'000 - 5 million reads per cell) compared to bulk experiments, higher variability and more outliers in the scRNAseq data. scRNA-seq data suffers from technical noise, batch effects and low capture efficiency. Confounding occurs when different biological conditions are processed in different batches, making the deconvolution of technical noise and biological effect impossible. This should be avoided by an appropriate experimental design that allows for the statistical deconvolution between unwanted and wanted variation. In scRNA-seq the single experimental unit is the cell, making it not always possible to use this approach. Different cells in an experiment may need different sample processing, or their biological differences may affect the downstream analyses.

The starting amounts of the library preparation can be as low as ten picogrammes of total RNA (Picelli et al., 2013). Two main issues arising due to the low starting amount are overamplification and low capture efficiency. Low and moderate expressed genes are not captured during the reverse transcription, which leads to dropouts of genes and a zero-inflated gene expression. scRNA-seq data has a large number of zero counts, which can be split into systematic, semi-systematic and stochastic zeros (Lun et al., 2016a). Systematic zeros and semi-systematic zeros come from genes that are silent in a subpopulation or across all cells, respectively. Stochastic zeros are zero counts that have been obtained due to a low capture efficiency during the library preparation. They affect genes with a count distribution near zero and have to be dealt with in the normalisation steps.

To deal with technical noise Unique Molecular Identifiers (UMI) or spike-ins are used. UMI are short random barcodes attached to the single-stranded cDNA in the reverse transcriptase process. By counting the unique UMI reads aligned to the genome an

estimated tag count is obtained. Spike-in are added before amplification. Under the assumption that the amplification of the endogenous and exogenous RNA is similar, they can be used for library size normalisation and to remove technical noise.

In general, scRNA-seq experiments consist of high-dimensional data. High-dimensional data suffers from the curse of dimensionality (Wagner et al., 2016), which means the distances in high-dimensional data become unstable and subpopulations cannot be separated (Andrews and Hemberg, 2017). Additionally, computational requirements are high. Reduction of the dimension is made by two approaches. Using linear or non-linear projections of the data from the original high-dimensional to a lower-dimensional space, or by feature selection, in which uninformative genes are removed.

The characterizing of the composition of cell types in a sample is one of the most common aims for scRNA-seq data. Lately, a broad spectrum of clustering methods have been specifically developed for the clustering of single cells, but up to now no comparison study testing the methods independently is known. The aim of the study is the comparison of clustering methods for scRNA-seq data. For the evaluation, publicly available datasets are used, providing a range of the sequencing depth, dropout fractions, type of expression values, number of subpopulations and difficulty of the dataset. Whereas these datasets provide the means to test the clustering methods on realistic data it has the disadvantage of unknown ground truth. To circumvent this, simulated datasets will be considered, enabling the evaluation of the clustering results on a secured ground truth of the clusters labels. Due to the high dimensionality and large technical variation of scRNA-seq data, filtering and normalisation are crucial steps in the clustering workflow. Here we want to investigate the effect on clustering using gene-wise filtered and unfiltered datasets. Another aim is to test the methods in the default mode by using the default settings of the methods with a minimum required user effort. This is considered as important as many users will use the methods without any fine-tuning of the methods. Finally, also the run time and the stability of the methods are investigated.

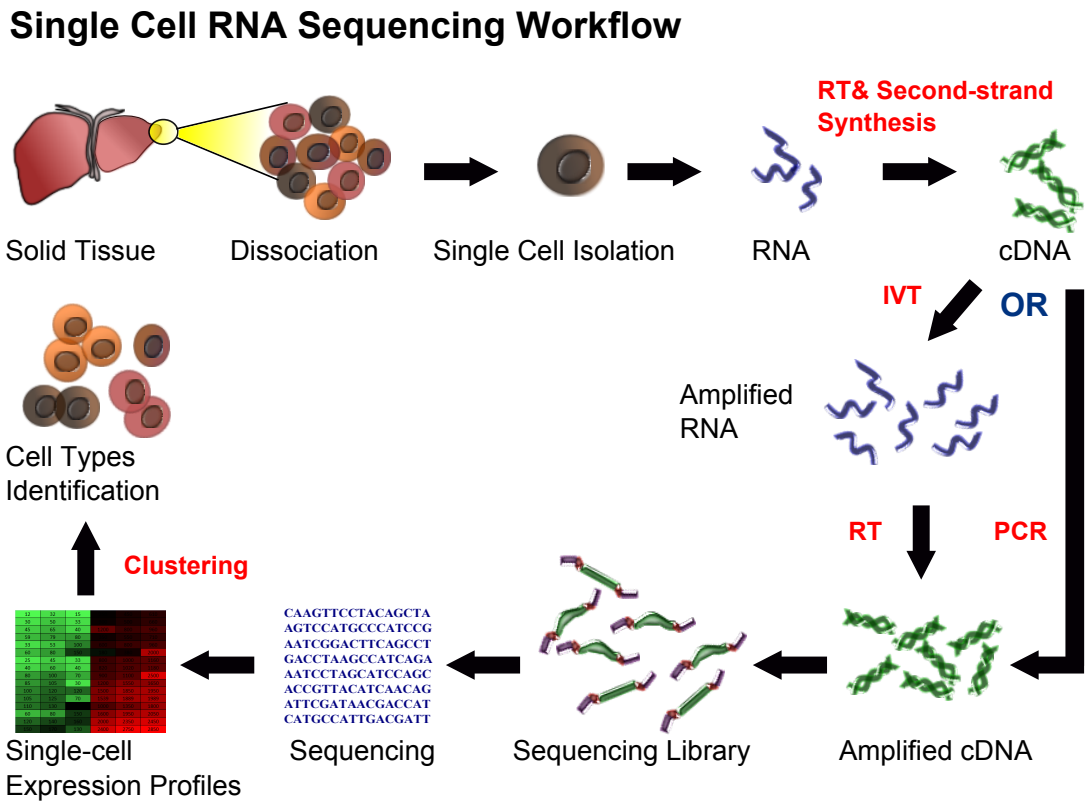


Figure 1: Single-cell RNA sequencing workflow. The workflow consists of the following steps: the isolation of the single cells and RNA, reverse transcription of the RNA to cDNA and the subsequent amplification, library preparation and sequencing. Taken from Wikipedia (`https://en.wikipedia.org/wiki/Single_cell_sequencing`).

2 Methods

2.1 Dimension reduction

All methods require a dimension reduction step before clustering. Commonly used methods are either Principal Component Analysis (PCA) (Hotelling, 1933) or t-distributed Stochastic Neighbour Embedding (tSNE, (van der Maaten and Hinton, 2008)). Given a data matrix $X(n \times p)$ with n measurements and p random variables, the aim is the reduction of dimension from p to q random variables. Then, PCA finds the linear combinations $a_1x^T, a_2x^T, \ldots, a_qx^T$ which have the successive maximum variance, subject to the constraint that its sample correlations with previous a_kx^T equal zero. Or in other words, PCA finds an orthogonal rotation of the original data in which the newly obtained first coordinates have the highest possible variance, the second coordinates the second-greatest variance etc. The linear combinations are the so-called principal components. Practically, PCA is computed by spectral decomposition of the correlation matrix R or covariance matrix S. The vectors $a_1, a_2, \ldots, a_q$ are the eigenvectors of S or R corresponding to the q largest eigenvalues (Jolliffe, 1986). Dimension reduction is achieved either by selecting only the first few PCs whose eigenvalues are above the average, or by determining the number graphically by plotting the eigenvalues (the so-called scree plot). PCA is deterministic and relatively fast but restricted to linear spaces.

In contrast, tSNE is a non-linear mapping (van der Maaten, 2013). Stochastic neighbour embedding (SNE) transforms Euclidean distances to conditional probabilities $p_{j|i}$. That is the probability that x_j is the nearest neighbour of x_i under a Gaussian centered at x_i. The low-dimensional counterpart $q_{i|j}$ is similar with a Gaussian centered at y_i and variance of $\frac{1}{\sqrt{2}}$. SNE minimises the divergence between $p_{j|i}$ and $q_{j|i}$ using the Kullback-Leibler divergence. tSNE implements a Student's t-distribution for the low dimensional space and a symmetric version of the cost function to simplify optimisation and to overcome the crowding problem. In tSNE, the cost function uses the joint probabilities p_{ij} and q_{ij} instead of conditional probabilities. To deal with large datasets, the Barnes-Hut implementation uses random walks on the nearest neighbour network with a PCA step to reduce the dimensionality of the high-dimensional data. tSNE is stochastic, depends on a perplexity parameter and distances between clusters are not preserved.

2.2 Clustering methods

Identifying unknown cell populations is one of the main uses of scRNA-seq data (Andrews and Hemberg, 2017). Ten clustering methods have been evaluated for this study. These methods and algorithms can be roughly classified into three groups: K-means, graph-clustering and hierarchical-based clustering. SC3, SIMLR, Linnorm, tSNEkmeans and RaceID use k-means in different fashions, while pcaReduce and CIDR are based on hierarchical clustering. The graph-based methods are TSCAN and Seurat. An overview of the methods is given in Table 1.

Table 1: Overview of the methods used in the study. Included in the table is a short description of the methods, the method for dimension reduction and clustering, if the method accounts for zero-inflation and if functions for normalisation and the auto-detection of the number of clusters are included.

Method	Description	dimension reduction	clustering	dropout	normalization	autodetection
tSNEkmeans	tSNE dimension reduction and kmeans clustering	tSNE	k-means	no	no	no
pcaReduce	PCA dimension reduction and k-means clustering through an iterative process. Step wise merging of cluster by joint probabilities and reducing the number of dimension by PC with lowest variance.	PCA	k-means, hierarchical clustering	no	no	no
SC3	PCA dimension reduction or Laplacian graph. k-means clustering on different dimensions. Hierarchical clustering on consensus matrix obtained by k-means	PCA	repeated k-means, hierarchical clustering on similarity matrix of k-means	no	no	yes
ZINBWaVE	zero-inflated negative binomial model	ZINB model	k-means	yes	yes	no
SIMLR	Learning of a distance metric via multiple kernel learning	tSNE	k-means	yes	no	yes
CIDR	PCA dimension reduction based on zero imputed similarities. Hierarchical clustering on a number of PC determined by variation of scree method.	PCA	hierarchical clustering	yes	no	yes
Seurat	Nearest neighbor graph based on PCA latent space	HVG and PCA	graph based	no	yes	yes
TSCAN	Ordering of cells by the use of a TSP algorithm.	PCA	graph based	no	yes	no
Linnorm	Normalisation using a subset of genes and clustering through tSNE/PCA and k-means	tSNE	k-means	yes	yes	yes
RaceID	Filtering and normalisation of cells and clustering with k-means.	-	k-means	no	yes	yes

K-means K-means clustering finds a predefined number of centers k and cell assignments, such that their within-group sum of squares is minimised (Hartigan and Wong, 1979). k cluster centers are randomly assigned. Each data point is then assigned to the nearest center using Euclidean distances. The centers are then recomputed using the average of the data points that are assigned to each of the k centers. This procedure is iterated until the algorithm converges. The initial assignment of the centers is random. Also, it's not guaranteed to find the global minimum. The drawbacks of the method are that it assumes spherical clusters and the sensitivity to scaling.

pcaReduce pcaReduce uses k-means clustering to find the number of clusters in the reduced dimension given by PCA (Žurauskienė and Yau, 2016). The main assumption is that large classes of cells are contained in low-dimension PC representation and more refined subsets of these cells types are contained in higher-dimensional PC representations. Given a gene expression matrix, the clustering algorithm starts with a k-means clustering on the PCA projections $Y_{n\times q}$ with $q+1$ clusters, where n is the number of cells and q are the number of PCs. The number of initial clusters k is typically around 30, guaranteeing that most cell types are captured. For all pairs of clusters, the joint probabilities are computed. Two clusters are merged by selecting the pair with the highest joint probability

or by sampling proportionally by the joint probabilities. The number of clusters is now decreased to $k-1$. Next, the PC with the lowest variance is deleted and a k-means clustering with $k-2$ centers is performed. This process is repeated until only one single cluster remains. When using pcaReduce q cluster partitions with $k-1$ clusters are obtained. The user can then choose the clustering with the desired number of clusters.

SC3 Implemented in the SC3 method is gene- and cell-filtering, as well as a log transformation step of the expression matrix (Kiselev et al., 2017). The filtered expression matrix is then used to compute Euclidean, Pearson and Spearman dissimilarity measures. By PCA or Laplacian graphs a lower-dimensional representation of the data is obtained. K-means clustering is then performed on the different dimensions. Next, a consensus matrix of the different clustering results is computed. The consensus matrix is a binary similarity matrix with entry one if two cells belong to the same cluster and zero otherwise. The consensus matrix is obtained by averaging the individual clusterings. The last step is a hierarchical clustering step with complete linkage. The cluster is inferred by the k level of hierarchy, where k can be supplied by the user. The method allows for the estimation of k. To reduce run time, SC3 changes the clustering method when supplied with more than 5'000 cells. Randomly selected cells are then used for the clustering approach described above. These subpopulations are then used to train a support vector machine to infer the cluster labels of the remaining cells.

SIMLR Most clustering methods rely on standard similarity metrics like Euclidean distances (Wang et al., 2017). SIMLR uses a weighted function of multiple kernels to compute a distance matrix. The assumptions is that the matrix has a block-diagonal structure, where the blocks represent the clusters c. The kernels are Gaussian kernels with a range of hyperparameters defining the variance of each kernel. The similarities are then used for data visualisation with tSNE or clustering using k-means on the latent space representations of the similarities.

CIDR Clustering through Imputation and Dimensionality Reduction (CIDR) takes the high dropout rate in scRNA-seq data into account (Lin et al., 2017). The method splits the squared Euclidean distance into three terms. These consist of one term in which a given gene is non-zero for a given cell pair, a second term in which one gene is zero and a third where both are zero. The authors state that only the cases where one gene is zero have a strong influence on the distances and the subsequent dimension reduction and clustering. To reduce the dropout-induced zero inflation, the method imputes the third term by its expected value given the distribution of the dropouts. The algorithm works basically in five steps: (i) Find features that are dropout candidates. That is genes that show an expression level below a threshold T. (ii) Find the empirical dropout probability using the whole data set. (iii) Calculate the dissimilarity using Euclidean distances together with a pairwise imputation process. Features that fall below the threshold T are imputed using a weighting function. The weighting is based on the probability of being a dropout. (iv) Perform dimension reduction using PCA on the imputed distance matrix. (v) Perform hierarchical clustering using the first few PCs. The number of PCs can be determined by several methods. Here, we use an implemented variation of the scree method.

Linnorm Linnorm is a normalisation and transformation method for count data (Yip et al., 2017). The main assumption is that a homogeneously expressed gene set exists. Using this gene subset, and by ignoring zero counts, the normalisation and transformation parameters are calculated. After normalisation, the expression values should show both homoscedasticity and normality. Linnorm includes functions for subpopulation analysis by t-SNE or PCA dimension reduction and subsequent k-means or hierarchical clustering.

First, the values are scaled according to the library size. Low count genes and genes that show high technical noise are filtered out. By default, genes showing a non-zero expression in at least 75 % of the cells are retained. Note that this threshold is set to maintain at least three non-zero cells per gene, in order to calculate the skewness of the gene distributions. By gradually increasing this threshold only genes that show a negative correlation between the mean and the standard deviation (SD) is assured. A locally weighted scatter plot smoothing (LOWESS) curve is fitted on the mean versus SD relationship. The SD is scaled and outliers based on the SD are removed. Next, genes that show a high skewness are filtered out. The data is then transformed using a modified log transformation. The dimensions are reduced by PCA or tSNE and clustering using the k-means algorithm. The R packages `fpc`, `vegan`, `mclust` and `apcluster` are used to determine the number of clusters.

TSCAN TSCAN uses a pseudo-time algorithm for cell ordering (Ji and Ji, 2015). PCA dimension reduction on the preprocessed gene expression data is performed. Preprocessing is done by adding a pseudo-count and log transformation. Low expressed genes are filtered out based on the zero-proportion and their covariance. Clustering is done by model-based clustering.

RaceID Rare Cell Type IDentification (RaceID) is a method developed for the data preprocessing, detection of outlier cells, k-means clustering and the inference of rare subpopulations (Grün et al., 2015). Cells are filtered based on a minimum number of transcripts. The expression matrix is normalised by scaling by the library size. To ensure non-zero values, a pseudo-count of 0.1 is added to the expression data. Using a gene filter, genes with a high zero-proportion are filtered out as well. The implemented clustering function uses k–means and the `clusterboot` function from the R package `fpc` for clustering. In contrast to the original k-means clustering in `clusterboot`, the newly developed function can use not only euclidean distances. The number of clusters is determined by the use of the gap statistic implemented in the `clusterboot` function.

The authors state that the algorithm is based on absolute transcript counts and is not tested for datasets containing other expression values than counts. With respect datasets other than the Zheng data the results should be interpreted carefully.

Seurat Seurat uses raw counts, where filtering is both done gene- and cell-wise. A user-specified threshold for the minimum number of expressed features per cell and the minimum number of the gene-wise expression per cell has to be defined. The counts from each cell are normalised by their total counts, scaled by a factor of 10'000 and log-transformed. A set of high-variable genes (HVG) is found by calculating the average expression and dispersion for each gene. Based on the average expression the genes are divided into bins. Within each bin z-scores for the dispersion are calculated. The HVG

are then selected by their standard deviation. Dimension reduction is done by PCA. The number of PCs is determined by a permutation test or graphically by plotting the eigenvalues of the PCs. Clustering is graph-based using a KNN-graph of the Euclidean distances in the PCA space. Clustering of cells is done by the Louvain algorithm (Butler and Satija, 2017).

ZINBWaVE Zero-inflated Negative Binomial-based Wanted Variation Extraction (ZINB-WaVE) uses zero-inflated negative binomial distribution to model the count data (Risso et al., 2017). By the use of a ZINB distribution, dropouts and over-dispersion are taken into account. The model allows for the inclusion of cell-level and gene-level covariates, such as batch effects or quality control measures. Included are also unobserved sample-level covariates, these are inferred from the data and can represent the unwanted variation or the biological effects of interest. The parameter estimation is done through a penalised likelihood approach. ZINBWaVE can be used for a low-dimensional representation of the data and in this study, clustering is done by the use of k-means.

2.3 Datasets

The datasets Kumar, Trapnell and Koh were downloaded from the *conquer* repository `http://imlspenticton.uzh.ch:3838/conquer`. The Zheng data was downloaded from the *10xGenomics* repository:
`https://support.10xgenomics.com/single-cell-gene-expression/datasets`. For the Kumar, Trapnell and Koh data Transcripts Per Million (TPM) on the count-scale which were independent of differences in the transcript usage between samples are used (Soneson et al., 2015). The Zheng data consists of UMI counts. Figure 2 shows the datasets in the tSNE space. Table 2 shows several summary statistics of the datasets. To have a measure of the compactness and difficulty of the datasets the silhouette widths were computed. Reported are the average silhouette widths for each dataset.

Kumar et al. (2014) The Kumar dataset consists of *Dgcr*8-knockout and V6.5 variotypes from mouse embryonic stem cells (mESCs). Cells were cultured on a serum with a Leukaemia Inhibitory Factor (LIF) or under Erk and GSK3 signalling inhibition (2Li). The authors investigated the expression of pluripotency factors and their involvement in the heterogeneity of pluripotent stem cells. Sample preparation and whole transcriptome amplification was done using a Fluidigm C1 system and following a SMARTer protocol. Sequencing was done using the Illumina system with paired-end reads. Sequencing depth was 1 million reads per cell.

Trapnell et al. (2014) Trapnell et al. (2014) used human skeletal muscle myoblast cells to investigate temporal differentiation. Cells were expanded under high–mitogen conditions. Differentiation was induced by switching to low-serum medium. Cells were captured before switching to low-serum medium (T0), after 24 h (T24) and 48h (T48). Between 49 and 77 cells were isolated at each time point and used for single mRNA-Seq library preparation by the use of a Fluidigm C1 system and following a SMARTer protocol. Libraries were sequenced with paired-end sequencing on a HiSeq 2500 (Illumina)

platform. Sequencing depth was 4 million reads per library. The authors excluded libraries that contained fewer than 1 million reads.

Koh et al. (2016) H7 human embryonic stem cells (hESCs) were used to study human mesoderm development. Starting from undifferentiated stem cells, several differentiation stages, sorted by time point and further refined by fluorescence-activated cell sorting (FACS) were isolated. Finally, ten different cell lines were obtained: undifferentiated H7 hESCs (H7hESC), anterior primitive streak populations (APS), mid primitive streak populations (MPS), lateral mesoderm (D2LtM), FACS-purified DLL1$^+$ paraxial mesoderm populations (DLL1pPXM), early somite progenitor populations (ESMT), PDGFR$^+$ sclerotome populations (Sclrtm) and two different dermomyotome populations (D5CntrlDRmmtm). In total, ten different cell types were then sequenced on a Fluidigm C1 system and following a SMARTer protocol. Libraries were sequenced through paired-end sequencing on a HiSeq 2500 (Illumina) platform. Sequencing depth was 1 - 2 million reads per cell.

Zheng et al. (2017) FACS-purified fresh peripheral blood mononuclear cells (PBMCs) were used to assess the performance of the 10xGenomics Chromium system. Sample preparation and library construction were done with a 10xGenomics Chromium system. The libraries were sequenced using an Illumina system. For this study, the datasets for CD19$^+$B, CD8$^+$CD45RA$^+$ naive cytotoxic, CD14$^+$ monocytes and CD4$^+$/CD25$^+$ regulatory T cells were used to construct an artificial population. From each library, 500 cells were sampled before being merged to obtain a single expression matrix.

Simulated datasets Using the Splatter package, expression data were simulated (Zappia et al., 2017). Parameters for the simulation were estimated from a subpopulation of the Kumar dataset. Embryonic stem cell variotypes V6.5 with signalling inhibition and LIF were used for the estimation. SimDataKumar consists of 500 cells with four subpopulations. The fractions per subpopulations were 0.1, 0.15, 0.5 and 0.25 of the total cell population. The probability that a gene is differentially expressed is 0.05, 0.1, 0.2 and 0.4 in the four different groups. Similarly, SimDataKumar2 consists of four subgroups with fractions of 0.2, 0.15, 0.4 and 0.25 of a total of 500 cells. The fractions of differentially-expressed genes were lower, with a probability of 0.01, 0.05, 0.05 and 0.08. The spike-in RNA is excluded before the parameter estimation.

Table 2: Summary statistics of the datasets. Shown are the number of cells, the number of subpopulations, the median number of genes, the median total read counts in millions, the average silhouette widths and the scRNA-seq protocols.

Dataset	Sequencing Method	No of cells	Median no. of genes	Median total counts (mio)	Avg. silhouette	No of subpopulations	Ref.
Kumar	SMARTer	268	25,894	1.67	0.53	3	*Kumar et. al (2014)*
Trapnell	SMARTer	288	13,616	1.86	0.04	3	*Trapnell et. al (2014)*
Zheng	10xChromium	2000	489	< 0.01	0.1	4	*Zheng et al. (2017)*
Koh	SMARTer	651	13,765	1.3	-0.04	10	*Koh et al. (2016)*
simDataKumar	-	500	29,861	1.78	0.15	4	*Simulation*
simDataKumar 2	-	500	29,974	1.78	0.03	4	*Simulation*

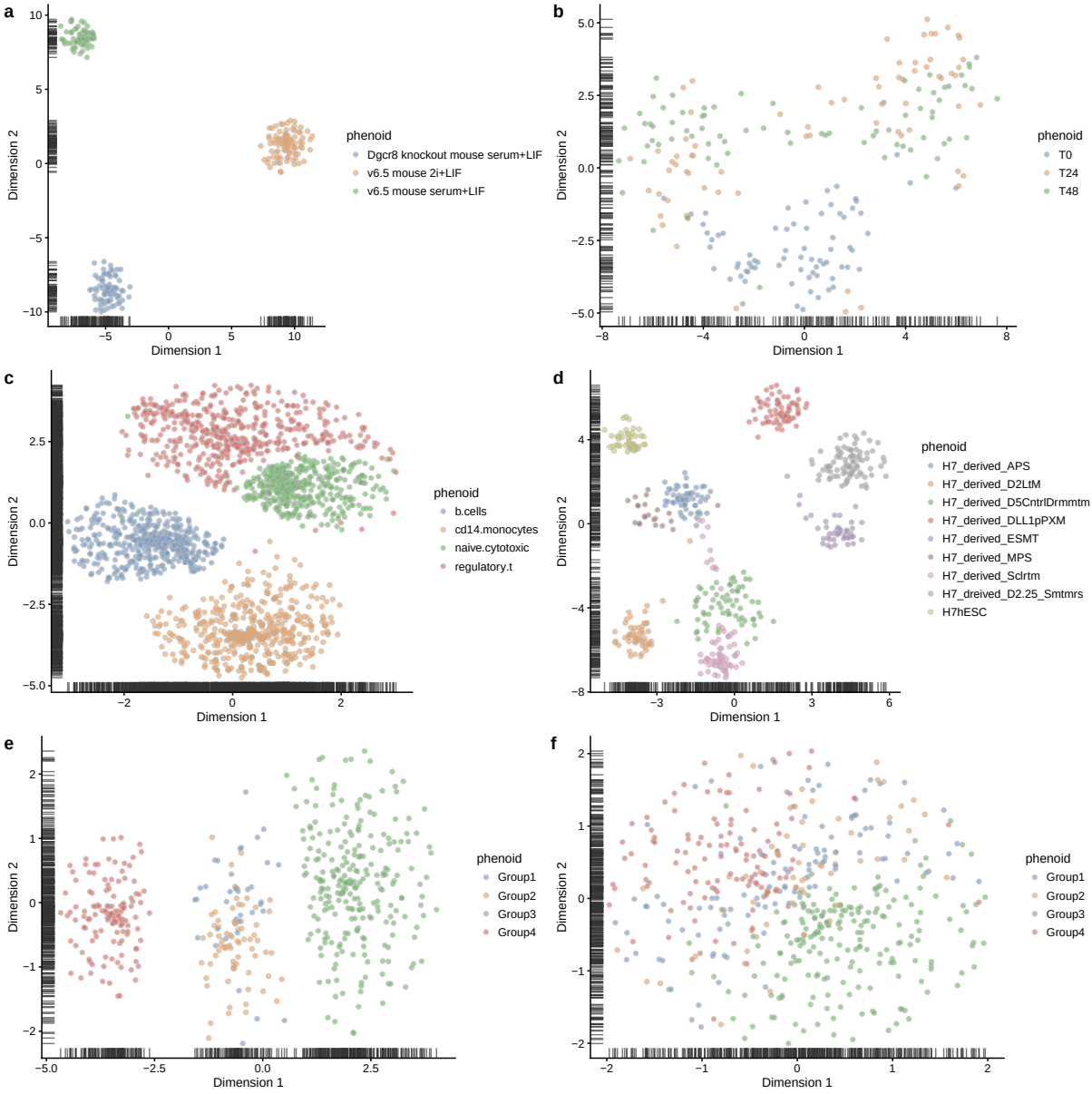


Figure 2: tSNE representation of the datasets (a) Kumar , (b) Trapnell, (c) Zheng, (d) Koh, (e) simDataKumar and (f) simDataKumar2 . The subpopulations in the datasets are color coded.

2.4 Data transformation and normalisation

Data transformation RNA-seq read counts may suffer from heteroscedasticity and skewness (Zwiener et al., 2014). Genes with higher mean have on average a higher variance across cells leading to unequal variances between different genes. To handle this property different transformation were considered. Namely, a binary logarithmic transformation with a pseudo-count of one, arcus sinus transformations and a variance-stabilising transformation (VST) from the DESeq package (Anders and Huber, 2010).

Log transformations are often used when dealing with skewed data and are a standard approach for normalising RNA-seq data. A pseudo-count of one is added to avoid taking the log of zero, and the transformation is defined as:

$$x_{ij}^{log} = log_2(x_{ij} + 1) \tag{1}$$

where x_{ij} are the counts of gene i and cell j. Log transformations will have an impact on extreme values. However, they will not address the problem of heteroscedasticity. Additionally, arcus sinus transformation were considered

$$x_{ij}^{arcsin} = \arcsin \frac{\sqrt{x_{ij}}}{c} \tag{2}$$

After transformation, the mean and the variances should be independent. VST addresses the problem of extreme values and unequal variances across genes. The counts are assumed to follow a Negative Binomial Distribution. The gene wise variances are then given by the relationship $\nu(\mu_j) := \sigma_j = \mu_j + \phi\mu_j^2$. The dispersion parameter ϕ is is defined as $\phi = a_0 + \frac{a_1}{\mu_j}$ with two constants a_0 and a_1 estimated using a generalized linear model. The transformed expressions are then derived by the variance-mean relation by

$$x_{ij}^{vst} = \int_0^{x_{ij}} \frac{1}{\nu(\mu_j)} d\mu_j \tag{3}$$

After such transformation, the mean and the variances of the genes should be independent, especially the high variances for low mean counts. For the study, a binary logarithmic transformation plus a pseudo-count of one is used. The mean-SD dependence for different transformations is shown in Figure 3.

Filtering and normalisation The quality control of the data sets follows Lun et al. (2016b). In the first step, genes that are not expressed in any cell (systematic–zeros) are removed in order to reduce the size of the expression matrix. Cells were filtered based on the library size and the total number of genes. Cells with $\log_{10}$ library sizes that are more than three median absolute deviations (MADs) below the median log-library size were filtered out (see Appendix A.2, Figures 16, 17, 18, 19, 20 and 21; a and b). The same filter was used with respect to the total number of genes per cell. For the Kumar and the Zheng dataset, ERCCs and mitochondrial counts were available. Cells with large proportions of ERCC or mitochondrial RNA are seen as low-quality cells. In the Kumar dataset, cells with a median ERCC proportion above three MADs are as well removed. The same filter was used for mitochondrial RNA in the Zheng data.

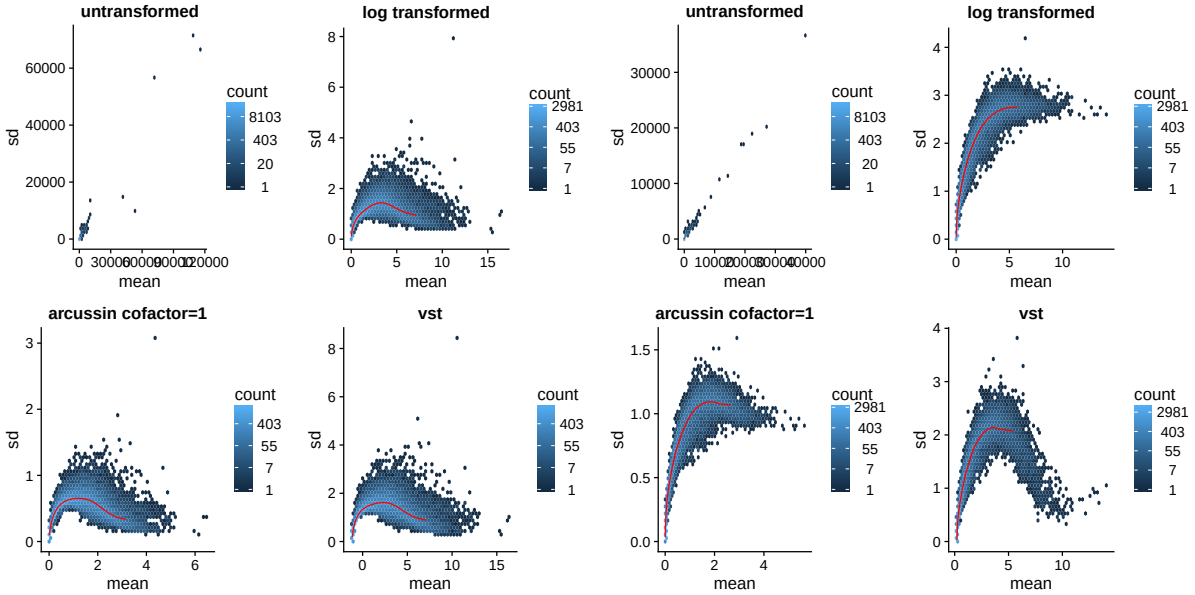
The metadata for the Trapnell dataset contained information about the cell quality. In this dataset, cells that were marked as debris and any single libraries consisting of more than one cell were filtered out. After filtering of the Kumar, Trapnell, Koh, Zheng, simDataKumar and simDataKumar2 datasets, 496, 222, 531, 1996, 496 and 496 cells were retained, respectively. The filtering was less strict in the Koh dataset compared to the original analysis where they retained 498 cells.

Low-abundance genes influence the mean-variance trend. Here low-abundance genes are filtered by their average counts (see Appendix A.2, Figures 16, 17, 18, 19, 20 and 21; d). For the Kumar, Trapnell data and the simulations, genes with average counts less than one are removed. The Zheng data set had a shallower sequencing depth. A different filter is used, and features which are not expressed in at least two cells are excluded.

Another examination of the technical variation was done using the marginal variances (Lun et al., 2016b). For that, a linear model with the expression values per gene as response variables and a chosen explanatory variable is fitted. The correlation coefficient can then be seen as the marginal explained variance for the explanatory variables.

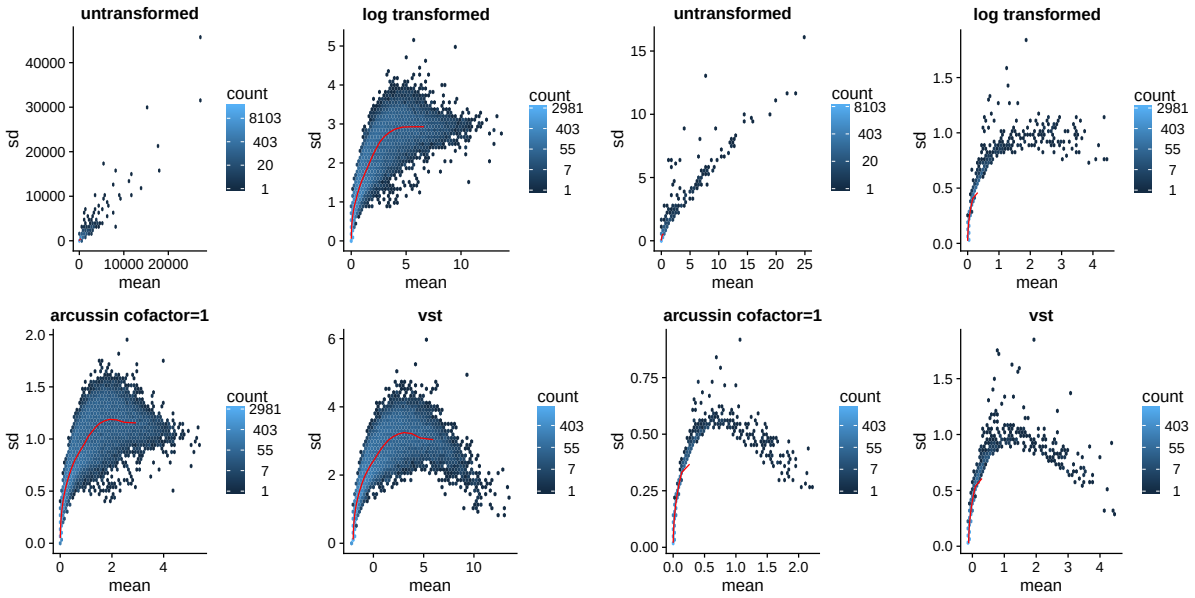
A wide variety of normalisation methods exist based on bulk RNA methods. These methods are usually not designed for dealing with the zero-inflated nature of scRNA-seq data (Lun et al., 2016a). Methods for normalisation of scRNA-seq data are based on spike-ins or RNA counts. Spike-in RNA is added before the library preparation. Any changes in the spike-in coverage are assumed to be due to technical factors. The normalisation is done by scaling the counts to level the spike-in. However, this approach is not feasible as none or only a limited number of spike-in counts were present in the datasets.

Here, normalisation through pooled cells is used, where the problem of excess zero counts is reduced by the pooling of multiple cells (Lun et al., 2016a). The normalisation procedure can briefly be described as follows: (i) Different pools of cells are defined. (ii) The expression values are summed across the cell pools. (iii) The cell pool is normalised against an average of the summed expression values. (iv) This step is repeated several times to construct a linear system. The summed count size is then used to estimate the corrected size factor. The size factors for the pooled cells are then "deconvoluted" into cell-based factors.



(a) Kumar dataset

(b) Koh dataset



(c) Trapnell dataset

(d) Zheng dataset

Figure 3: Shown is the gene-wise standard deviation versus the mean for the datasets Kumar (a), Koh (b), Trapnell (c) and Zheng (d). Different transformations were considered; log, arcus sin and VST transformations.

2.5 Evaluation of the clustering methods using different run modes

Using filtered and normalised data, the methods were operated in the default mode, with the number of clusters given by the ground truth and under a range of parameters. Additionally, the methods were also run with the unfiltered datasets. Many users will use these methods in the default mode, hence it was seen as important that the results were provided without any fine-tuning of the parameters. The run parameters in the default mode were given either according to the packages default settings or by using examples from the package vignettes. If the method was able to detect the number of subpopulations, this auto-detection function was used to infer the number of clusters. When the number of clusters had to be provided, the number of clusters given by the authors of the datasets were used. Clustering results have to be evaluated using some sort of "ground truth". Here, the cell annotation provided by the authors of the datasets or the given truth from the simulations were used. Seurat, TSCAN, RaceID, ZINBWaVE and Linnorm each have their own filtering and normalisation procedures. In order to test these methods preprocessing capabilities, the methods were tested with the unprocessed raw counts. The methods SC3, tSNEkmeans, pcaReduce, SIMLR and CIDR do not include filtering and normalisation steps. For these methods filtered, normalised and log-transformed counts, detailed in section 2.4, were used. An overview of the filtering and normalisation steps used by the methods is given in Table 4.

In a further analysis, the clustering methods were tested for different values of the number of clusters k. Seurat does not allow the number of clusters to be set. Hence, Seurat was run under a range of the parameters of the number of neighbours and the resolution parameter.

To assess the stability of the clustering methods, a random subsample of cells without replacement was drawn from the Kumar dataset. The size of the subsample was 100 and the subsampling was repeated 30 times. The clusterings for each method were then compared using the overlapping samples and the ARI scores.

Table 3: The table summarizes the parameter settings for the different run modes. Shown are the parameters for the datasets Kumar, Trapnell, Zheng, Koh, simDataKumar and simDataKumar2.

Method	Parameter	default	unfiltered	filtered
CIDR	**k**	NULL	3, 3, 4, 10, 4, 4	3, 3, 4, 9, 4, 4
	nPC	4	5, 10, 8, 8, 3, 3	5, 10, 8, 8, 3, 3
	cMethod	Ward.D2	Ward.D2	Ward.D2
tSNEkmeans	**k**	3, 3, 4, 9, 4, 4	3, 3, 4, 10, 4, 4	3, 3, 4, 9, 4, 4
	perplexity	30	30	30
	nPC	50	20	20
pcaReduce	**k**	3, 3, 4, 9, 4, 4	3, 3, 4, 10, 4, 4	3, 3, 4, 9, 4 ,4
	nbt	100	100	100
	method	S	S	S
Linnorm	**k**	range 1 to 20	3, 3, 4, 10, 4, 4	3, 3, 4, 9, 4, 4
	minNonZeroPortion	0.75, 0.75, 0.1, 0.75, 0.75, 0.75	0.75, 0.75, 0.1, 0.75, 0.75, 0.75	0.75, 0.75, 0.1, 0.75 ,0.75, 0.75
	BE_strength	0.5	0.5	0.5
SIMLR	**k**	3, 3, 4, 9, 4, 4	3, 3, 4, 10, 4, 4	3, 3, 4, 9, 4, 4
	normalize	FALSE	TRUE	TRUE
SIMLRlarge	**k**	3, 3, 4, 9, 4, 4	3, 3, 4, 10, 4, 4	3, 3, 4, 9, 4, 4
SC3	**ks**	range from 2 to 15	NULL	NULL
	k	NULL	3, 3, 4, 10, 4, 4	3, 3, 4, 9, 4, 4
	pct_dropout_max	90	90, 90, 99, 99, 90, 90	90, 90, 99, 99, 90, 90
TSCAN	**k**	range from 2 to 10	3, 3, 4, 10, 4, 4	3, 3, 4, 9, 4, 4
	minexpr_percent	0.5	0.5, 0.5, 0.1, 0.5, 0.1, 0.1	0
Seurat	**resolution**	0.6, 0.6, 0.6, 0.7, 0.6, 0.6	0.6, 0.6, 0.6, 0.7, 0.6, 0.6	0.6, 0.6, 0.6, 0.7, 0.6, 0.6
	neighbors	30	10 percent of dataset	10 percent of dataset
	mincell	0	2	0
	mingenes	0	0	0
	dimsuse	NULL	9, 12, 10, 15, 10, 10	9, 12, 10, 15, 10,10
ZINBWaVE	**k**	3, 3, 4, 9, 4, 4	3, 3, 4, 10, 4, 4	3, 3, 4, 9, 4, 4
	number of HVG genes	1000, except Zheng =200	1000, except Zheng = 200	1000 , except Zheng = 200

2.6 Parameter settings

The number of clusters k is the main parameter used for most of the methods. Except for Seurat, the clustering functions for each of the methods allow for the number of subpopulations to be directly controlled. Seurat allows the setting of k only indirectly through a resolution parameter. The other important parameters were the number of kNN, the number or the type of latent space dimensions used for the clustering algorithms and the settings of the filtering and normalisation steps. The methods pcaReduce, SC3, Linnorm, RaceID and TSCAN, can be run in an unsupervised mode, and no parameters have to be provided. Although it is possible to run these the methods unsupervised, fine-tuning of the parameters is often recommended by the authors of the methods. CIDR, tSNEkmeans and SIMLR need the specification of the number of clusters k. For Seurat, the number of PCs or the number of the kNN have to be defined. An overview of the chosen parameter settings is given in Table 3. Next, a brief overview of the chosen parameter setting and the rationale behind it is given.

tSNEkmeans To reduce the run time the Barnes-Hut tSNE implementation from the R package `Rtsne` was used. Different values of the perplexity parameter can give different tSNE representations; however, here the default setting with the perplexity parameter set to 30 was chosen. tSNE is performed on the first 50 dimensions in the PCA latent space in the default mode. Otherwise, 30 dimensions were chosen.

pcaReduce For pcaReduce, the range of clusters cannot be specified. Instead, the number of dimension q in the PCA latent space are to be specified. The results are $q - 1$ different clustering solutions, with $k - 2$ clusters. For all data sets, 30 dimensions were chosen, and the evaluation was based on the respective number of clusters in the subsequent analysis. The method is stochastic and has to be run several times in order to give stable results. Here, 100 samples were chosen, and the merging of clusters was done by sampling that was proportional to the joint probabilities. In order to obtain a consensus from the 100 clusterings, the ensemble clustering methods implemented in the R package `clue` were used.

SC3 A gene-filtering step is implemented in this method. Based on the dropout distribution, genes that are below the 10th and above the 90th percentile are filtered out. However, for the Koh and Zheng datasets, the upper threshold is set to the 99th percentile. Due to the high dropout rate in these datasets, it was otherwise not possible to run the method. When running under the default mode, a range of clusters from 2 to 10 are given, and the number of subpopulations is automatically inferred by the method. Otherwise, k is set to the number of annotated subpopulations.

SIMLR A gene-wise mean normalisation step is implemented by the method. When running in the default mode, no normalisation was used. However, in the other run modes normalisation was included. Without normalisation, the method fails in the spectral decomposition of the similarity matrix. The tuning parameter was set to the default value of 10 on all runs. The number of clusters is set according to the run mode that is being used.

CIDR CIDR uses three parameter settings: the number of clusters, the number of PCs (nPCs) and the method for hierarchical clustering. By default, Ward linkage is used in the hierarchical clustering. CIDR is able to automatically detect the number of clusters n. By default, n is set to $nPC * 2 + 2$ and the parameter nPC is set to 4 by default. When running in a mode other than the default, the parameter nPC was chosen by a variation of the scree-plot, and the number of clusters was set accordingly to the respective dataset. The number of PCs used for the datasets Kumar, Trapnell, Koh, Zheng, simDataKumar and simDataKumar2 is 5, 10, 8, 8, 3, and 3, respectively.

Seurat Implemented in the method is a normalisation and a gene-filtering step. The filtering criteria are based on how many cells show an expression of a certain gene and the number of total features per cell. By default, no cell-filtering step is included when preprocessed datasets are used.

When running with unfiltered data, genes that are expressed in less than two cells were filtered out. When using the Zheng data, the threshold is set to one, according to the filtering detailed in section 2.4.

The default log normalisation is used, which is currently the only option. The scale factor for cell-level normalisation was set to the default of 10'000. As a default, no explanatory variables were chosen to be regressed out. The experimental batch would be a natural choice as a covariate, but it cannot be used as the datasets containing this information are completely confounded.

The clustering parameters to be defined were a resolution parameter and the number of PCs. The resolution parameter was set to 0.7 for the Koh dataset and 0.6 for the other datasets. The number of PCs was determined according to the methods recommended by the authors. Namely, through the use of a scree plot and a jackknife permutation test the number of PCs was determined. In terms of the datasets, Kumar, Trapnell, Zheng, Koh, simDataKumar and simDataKumar2 were used, with the number of PCs being 9, 12, 10, 15, 10, and 10, respectively. Ten percent of the total cells were used as the number of neighbours in the k-nearest neighbour algorithm.

TSCAN TSCAN adds a pseudo-count of one with the data being log-transformed; this setting is used for all run modes. In the default run-mode, genes that show zero expression in at least half of the cells are filtered out. To be able to run the method using the unfiltered data, this threshold was changed to 0.1 for the Zheng, simDataKumar and simDataKumar2 data.

This filter was switched off when working with the prefiltered datasets. By default, the method infers the number of clusters from a range of 2 to 9 clusters. Here, a range from 2 to 10 was used in the default mode. If run semi-supervised, the respective number of clusters is given. By default, "ellipsoidal, varying volume, shape, and orientation" is used for the model.

RaceID In default mode, cells with a minimum total library size of 3'000 are retained. The gene filter is set to filter out all genes with less than five transcripts in at least one cell. Over-saturated genes, which have over 500 transcripts per cell, are also filtered out. Here, we only use this filter with the Zheng data, as it is the only set that contains UMI counts. Otherwise, the filter is turned off. The gap statistic is used to determine the

number of clusters. The default setting from the clusterboot function is set to a range from 2 to 20 clusters.

When using the run mode with unfiltered datasets, the minimum total library size was set to 1'000, 500, 400, 420, 1'200 and 1'200 for the datasets Kumar, Trapnell, Koh, Zheng, simDataKumar, and simDataKumar2, respectively. These thresholds were chosen so that they corresponded with the thresholds based on the MADs. The filters for over-saturated genes and minimum gene expression were turned off as well, corresponding to the filtering steps in section 2.4. Filtering, based on the original analysis, was done using mean counts. To set the gene filter, we retained those genes that showed at least one transcript for two cells in the count-based datasets. The exception to this was the Zheng dataset, where genes which show at least five counts in two cells are retained.

When the method is run with prefiltered datasets, the filters are turned off, and the appropriate number of clusters is provided. In all run modes, the Pearson metric is used as the distance measure.

Linnorm Except for the Zheng data, the filtering thresholds are set to the default in all run methods. Due to the low sequencing depth of the Zheng dataset, the minimum non-zero expression had to be set to a proportion of 0.1. tSNE and k-means are used for dimension reduction and clustering, respectively. In the default run mode, a range of k from 2 to 20 was provided in order to allow us to infer k.

ZINBWaVE Lowly expressed genes are removed by removing genes that do not show at least five reads in at least five cells. Following the recommendations by the authors only 1'000 highly variable genes are retained, mainly because of computational reasons. The number of dimensions for the low-dimensional space K is set to two and a subsequent clustering on the latent space is done by the use of the k-means algorithm.

2.7 Evaluation metrics

One of the evaluation criteria used was the Hubert-Arabje Adjusted Rand Index (ARI), which is used to compare two partitions (Hubert and Arabie, 1985). The metric is adjusted for chance by subtracting the uncorrected index by its expectation, and divided by a scale factor. Independent clusterings have an expected value of zero and are one if there is full agreement between the partitions. The index can take on negative values. Given there are two partitions X and Y of n elements, they can be summarized in a contingency table with i rows and j columns. The ARI is then defined as

$$ARI = \frac{\sum_{ij}\binom{n_{ij}}{2} - (\sum_i \binom{a_i}{2}\sum_i \binom{b_j}{2})/\binom{n}{2}}{\frac{1}{2}[\sum_i \binom{a_i}{2} + \sum_j \binom{b_j}{2}] - [\sum_i \binom{a_i}{2}\sum_j \binom{b_j}{2}]/\binom{n}{2}} \tag{4}$$

where n_{ij} are the number of elements in agreement between the partitions and a_i and b_j are the row sums and column sums, respectively. Another metric is the F1 score. It is the weighted average mean between precision and recall. The weights are defined as the inverse of the precision and recall. The F1 score is defined as

$$F_1 = 2 * \frac{precision * recall}{precision + recall} \tag{5}$$

F1 scores can take on values between zero and one. The predicted clusters and the "ground truth" were matched by the Hungarian algorithm. Some of the clustering methods are unsupervised and the partitions do not need to have the same sizes (non-bipartite). This causes problems with the Hungarian algorithm. As a solution to this issue, the assignment matrix was augmented with dummy columns that had the maximum matrix value as their entries.

2.8 Software and environment

All analyses were performed in R (version 3.4.2 (2017-09-28)) (R Core Team, 2013), a free software environment for statistical computing and graphics which is available at `http://www.r-project.org/`. The following method-specific R packages were used: `Rtsne` (version 0.13), `ClusterR` (version 1.0.8), `pcaReduce` (version 1.0), `cidr` (version 0.1.5), `Linnorm` (version 2.2.0), `Seurat` (version 2.1.0), `SIMLR` (version 1.4.0), `TSCAN` (version 1.16.0), `zinbwave` (version 1.0.0). These can be downloaded from the Bioconductor repository `https://bioconductor.org`. The package RaceID can be downloaded from the repository `https://github.com/dgrun/RaceID`. In addition the base packages `scater` (version 1.6.0), `ggplot2` (2.2.1), `cluster` (version 2.0.6), `pheatmap` (version 1.0.8), `clValid` (version 0.6-6), `clue` (version 0.3-54) and `splatter` (version 1.2.0) were used in the analysis. The computing environment had the following specifications: macOS Sierra (Version 10.12.6) and 2.7 GHz Intel Core i5 Processor with 8 GB RAM.

Table 4: Overview of filtering and normalisation steps by method.

Method	Cell filtering	Gene filtering	Normalisation	Autodetect	Expression values
CIDR	no	no	no	yes	log-norm counts
Linnorm	no	yes	yes	no	counts
pcaReduce	no	yes	no	no	log-norm counts
RaceID	yes	yes	yes	yes	counts
SC3	no	yes	no	yes	log-norm counts
Seurat	yes	yes	yes	no	counts
SIMLR	no	no	no	no	log-norm counts
TSCAN	no	yes	yes	no	counts
tSNEkmeans	no	no	no	no	log-norm counts
ZINBWaVE	no	yes	yes	no	counts

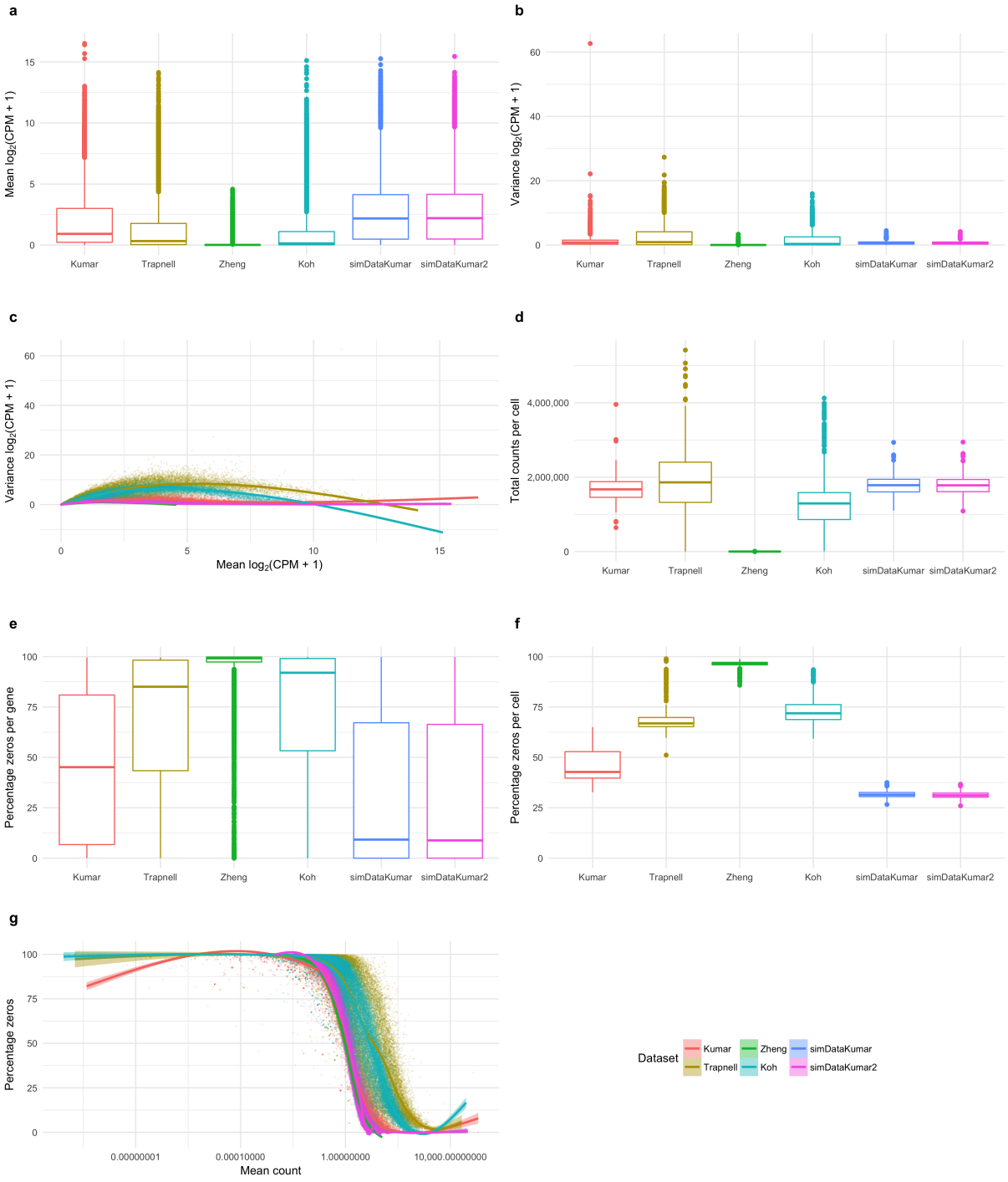


Figure 4: Comparison of the datasets and the simulations. Shown are the gene means (a), variances (b), the mean-variance relationship (c), library size (d), the distribution of zeros per gene (e) and per cell (f) and the relationship between the mean expression of a gene and the percentage of zeros (g). The plots are based on the compareSCEs function from the R-package `Splatter`.

3 Results

3.1 Evaluation of the performances of the methods

For this study, the methods were evaluated using six different datasets and three different run modes. By running the clustering methods using the unfiltered and filtered datasets, we can (to some extent) investigate the effect of the filtering and normalisation steps on the clustering. RaceID, Linnorm, TSCAN and Seurat each have their cell or gene-wise filters implemented; these were included when the unfiltered datasets were used.

By running the methods under the default mode, we can investigate the performance with a minimum user effort. Also, some methods are able to auto-detect the number of clusters, and this setting was included when running in the default mode. The methods were then run with a refined parameter setting and the annotated number of clusters. Note that the number of parameters varied considerably between the methods; in this study, we chose only the parameter that is regarded as the most important.

The clusterings of the methods were assessed by the use of the ARI and F1 metrics. The ARI scores for the different run modes are shown in Figure 5 and 6. According to the dataset used, Figure 8 shows the differences in ARI scores between the method with the highest ARI score and the other methods. The differences between the ARI scores of the different run modes are shown in Figure 7. The comparison of the different run modes is strictly speaking, only possible for deterministic methods such as CIDR, and is not possible for stochastic methods (e.g. tSNEkmeans). The F1 scores for the filtered dataset are shown in Figure 9. The results for the run mode with the default setting and the unfiltered data are shown in the Appendix (see Figure 14, 15).

The datasets and simulations varied in the number of cells, the library sizes, the number of subpopulations, the zero-fractions per gene and the type of the expression values (see Table 2 and Figure 4). The datasets Kumar, Trapnell and Koh show a similar distribution in the gene means, variances and the library sizes. The Zheng dataset had a lower sequencing depth and shows a high dropout rate. The parameter estimation for the simulated dataset is based on the Kumar dataset. When comparing the simulations with the Kumar datasets they are comparable by their mean, variance and library size distributions, but their gene and cell-wise zero fractions are lower. In order to assess the accuracies of the methods, some ground truth of the type of subpopulations is needed. Here, we used the annotation given by the authors of the datasets. This may not be correct; the cell types could be wrongly annotated, or the annotated clusters might consist of more refined unknown subpopulations. However, here we used this annotation because it was seen as the best information available. The datasets were chosen such that there was a range in clustering difficulty. As an objective measure of the clustering difficulty, the average silhouette coefficient is used.

The average silhouette width of the Kumar dataset is with 0.53 the highest for all datasets. It consists of three distinct cell populations and is the simplest of all the datasets. A high proportion of the variances are explained by the treatment or the batch effect, as these two are not separable (see Figure 16, e). This dataset can be seen as a benchmark for the dataset as no method should have problems in clustering this dataset. Using the filtered datasets, SC3, pcaReduce, SIMLR, CIDR and ZINBWaVE, all achieved a correct

partition of the cells. The other methods also achieved high accuracies with ARI scores between 0.97 and 0.99. The F1 scores give a more in-depth view of the actual partitioning, and for the filtered data we have similarly high F1 scores for each of the subpopulations, showing that no method failed to cluster one of the subpopulations. In contrast to the uniform results with the filtered datasets, the results were more variable when running in default mode and with the unfiltered datasets. Running with the default setting, and automatically detecting the clusters, methods RaceID, SC3 and TSCAN failed to detect the correct number of clusters. RaceID and TSCAN partitioned the cells into four clusters and SC3 into five clusters. Additionally, TSCAN failed to correctly partition the cells into three subgroups. Note that for SC3 and TSCAN, the extra cluster consisted only of a few cells and had only a marginal effect on the final clustering. ZINBWaVE, tSNEkmeans, pcaReduce and Linnorm failed in clustering the three populations correctly. With the unfiltered datasets, the methods show similar results where most of the methods again achieved ARI scores close to one. There are two exceptions that had a drop in the performance: Linnorm and ZINBWaVE. These two methods failed by clustering one particular subpopulation.

The simDataKumar simulation has an average silhouette width of 0.15 and is one of the simpler datasets for clustering. Three out of four subpopulations in the dataset are distinct, with a high proportion of DE genes. Only 5 % of the subpopulation Group 1 are DE genes and based on the tSNE representation of the dataset, the two subpopulations Group1 and Group2 are not distinguishable (see Figure 2). For the filtered data and run with the annotated number of clusters, the methods SC3, Seurat, pcaReduce, SIMLR and CIDR had high ARI scores between 0.95 and 1.00. Also, Linnorm and TSCAN showed a somewhat lower performance with an ARI score of 0.87 and 0.90, respectively. However, tSNEkmeans, TSCAN and RaceID failed to correctly cluster the dataset with ARI scores between 0.31 and 0.65. The low ARI scores are due to failing to partition the subpopulation Group 1. TSCAN failed in clustering the results due to the high threshold for the zero expression. CIDR, RaceID, SC3 and Seurat, the methods with an auto-detect function, were all able to correctly identify the number of clusters. Running the data in default mode had no impact on the clusterings for pcaReduce, SC3, CIDR, RaceID, tSNEkmeans and SIMLR. However, Seurat and ZINBWaVE had a different partition. Using unfiltered data affected the methods Linnorm, RaceID and TSCAN. Note that for Linnorm and RaceID this could be due to the stochasticity of the method. For the other methods, it had no impact, and the ARI scores were stable.

The Zheng dataset is a mixture of four populations of PBMCs. The two subpopulations $CD19^{+}$B and $CD14^{+}$monocytes are distinct cell populations, whereas the naive cytotoxic and regulatory T cells are overlapping populations. The tSNE representations show that $CD19^{+}$B and $CD14^{+}$monocytes form two separate clusters, with a third cluster that consists of the two nested populations naive cytotoxic and regulatory T cells. The dataset has a medium difficulty with an average silhouette width of 0.1. The dataset has a low sequencing depth and a high dropout rate. The performances given by the methods were highly variable on the different run modes. With the filtered data SC3, pcaReduce and tSNEkmeans had ARI scores near one. Also, Seurat, Linnorm and SIMLR had high accuracies with ARI scores between 0.88 and 0.92. CIDR dropped in performance when

compared to its performance for other datasets. The Zheng datasets consist of UMI counts, for which CIDR is not designed. When running in the default mode CIDR, SC3 and Seurat detected an extra cluster, which explains the lower performance for Seurat and SC3 in the default run mode. With the exception of method SIMLR (for large-scale data) and ZINBWaVE, using filtered data had no impact on the performance of the methods.

The more difficult simulation simDataKumar2 has an average silhouette width of 0.03. For this dataset, all the proportions of DE genes are relatively low with five to eight percent. The subpopulation Group 3 is distinguishable from the other three in the tSNE representations. Group 1, Group 2 and Group 4 form a single non-separable cluster in the tSNE representation. SC3, SIMLR, Seurat, CIDR and pcaReduce were mostly able to correctly cluster the cells when using the filtered datasets and run with the annotated number of clusters. The ARI scores were between 0.9 and 1.00. The other methods showed profoundly lower performances. When running under the default mode, SC3 and RaceID detected only three of the clusters when using the auto-detect function of the methods, explaining the drop in the ARI scores for these methods. The methods Linnorm, tSNEkmeans, TSCAN and to a lesser extent SIMLR (large scale) showed a decrease in the ARI scores. For the other methods using the unfiltered did not affect the clusterings.

The Trapnell dataset is not a mixture of distinct cell populations, and the development of the populations followed a time-dependent trajectory. In tSNE space, the non-differentiated cells form a distinct cell cluster. However, the more differentiated cells later on the time axis are, at least in tSNE space, non-separable. It is also notable that the batch explains a higher proportion of the variance than the cell type (see Appendix: Figure 17). The average silhouette width is 0.04, and it is one of the more difficult datasets.

The methods showed the lowest ARI scores for this dataset, and the maximum ARI score achieved SC3 with 0.55, showing the difficulty to cluster this dataset. The other methods all had ARI scores below 0.5. TSCAN is specially developed for this scenario (Ji and Ji, 2015). However, the method did not perform any better than the other methods. To improve the clustering results for TSCAN, it is possible to provide a starting point for the trajectory. However, this was not done in this study. By running the dataset in the default mode, the ARI scores varied considerably compared to the run mode with the filtered and annotated number of clusters. Indeed, depending on the method, the scores were better or worse or stable. SC3 detected an additional three clusters. RaceID detected only one single cluster. CIDR and Seurat found three subpopulations but failed to correctly label the cells. Except for TSCAN, the filtering led to an increase in the ARI scores. Without filtering, most of the methods failed to cluster the dataset as they had scores around zero.

Koh was the dataset that had the highest number of subpopulations; ten clusters were annotated by the authors of the original study. During the cell filtering of the Koh dataset, one subpopulation was wrongly detected as outlier cells and was removed during the filtering steps (D3GARPpCrdcM). The average silhouette width is -0.04 and the dataset is one with the highest difficulty. In the tSNE representation, some cell types are easily distinguishable, whereas other are mixed. On average, 10 % of the variance is explained by the cell type. Similar to the Zheng dataset, the Koh data has a low

sequencing depth and a high dropout rate. For the filtered data, only SC3 and pcaReduce were able to correctly identify all nine subpopulations with the highest ARI score of 0.96. CIDR, SIMLR, pcaReduce and Seurat also showed good performances with ARI scores between 0.88 and 0.92. In the middle field are the methods tSNEkmeans, Linnorm and ZINBWaVE with ARI scores between 0.62 and 0.8. RaceID failed for this data with an ARI score of 0.26. Compared to other run modes, the clustering was the overall worst for each method when running with the unfiltered dataset, indicating the importance for pre-filtering this dataset. Except for the unstable methods, Linnorm and tSNEkmeans, running the methods in default mode only had a small impact on the ARI scores. However, the number of detected clusters varied greatly between the methods with an auto detect function. CIDR, TSCAN and RaceID missed one or more clusters, and SC3 detected an additional three clusters.

Overall, Kumar provided no difficulties for most of the methods. Most methods also achieved high scores for the SimDataKumar2 and the Zheng datasets. The Trapnell dataset was a challenge for the methods, and only low accuracies were achieved. When looking at the F1 scores and the size of the clusters, wrongly assigned clusters tend to be smaller in size. The investigated methods either used PCA or tSNE for dimension reduction. For the clustering, either graph-based, K-means, hierarchical clustering or combinations of these are used. No connection between the performance and the type of the dimension reduction or clustering approach can easily be seen.

3.2 Evaluation of the clustering results

Based on the average scores, SC3 was the best performing method when running with the filtered datasets. Exceptions are seen in the default and unfiltered setting when using the Trapnell and the Zheng datasets. A pre-filtering step and a fine-tuning of the parameters for this method is recommended, as this improved the accuracy of the method. It is noteworthy that the method was able to correctly classify more than 95% percent of the cells for the Kumar, Koh, simDataKumar, simDataKumar2 and Zheng datasets. Even if it had low scores for the Trapnell data, it achieved the highest accuracies of all methods in this dataset. SC3 can detect the numbers of clusters automatically and did so correctly for the simulated datasets. For the other datasets, the number of clusters was higher than in the annotation. It is unknown whether this is due to the existence of more refined subpopulations. The method pcaReduce achieved the highest average ARI scores in the default mode and when run with the unfiltered datasets. Filtering of the Koh and Trapnell datasets lead to an improvement of the performance. Although pcaReduce showed a high performance, a drawback is the instability of the method.

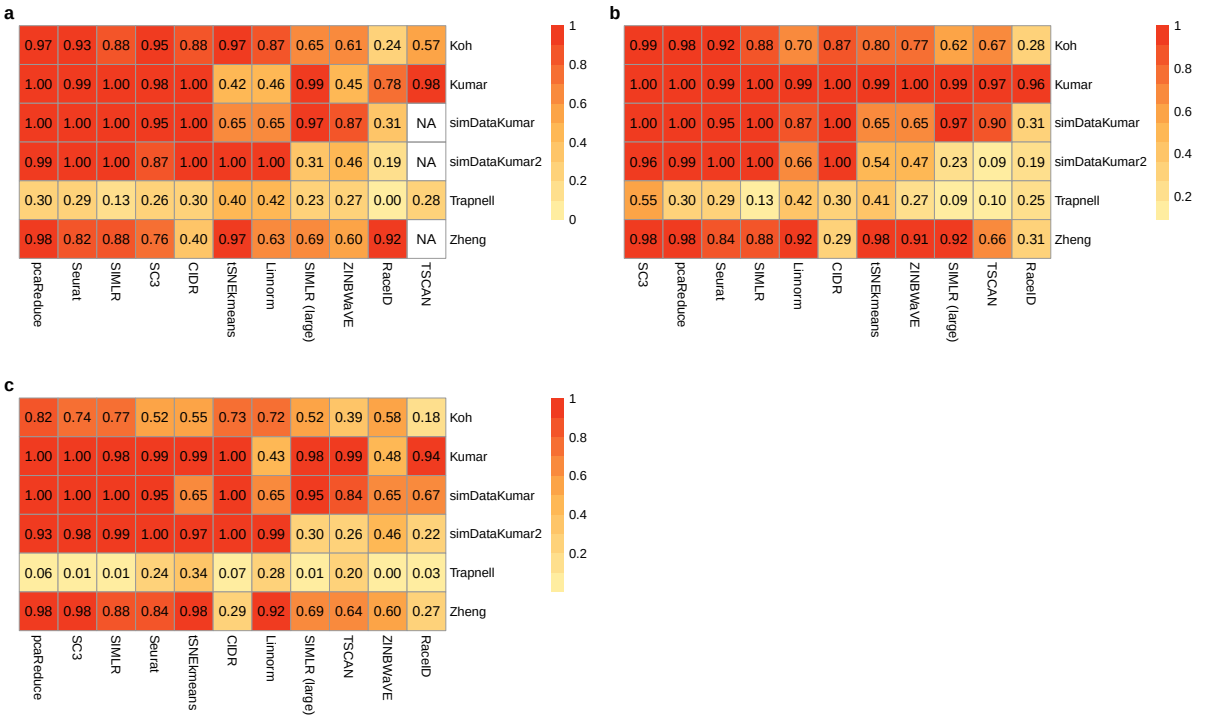


Figure 5: ARI scores for the datasets Koh, Kumar, Trapnell, Zheng and the simulations simDataKumar and simDataKumar2. Shown are the ARI scores for the run mode (a) default, (b) filtered and (c) unfiltered. The methods are ordered by their average ARI score.

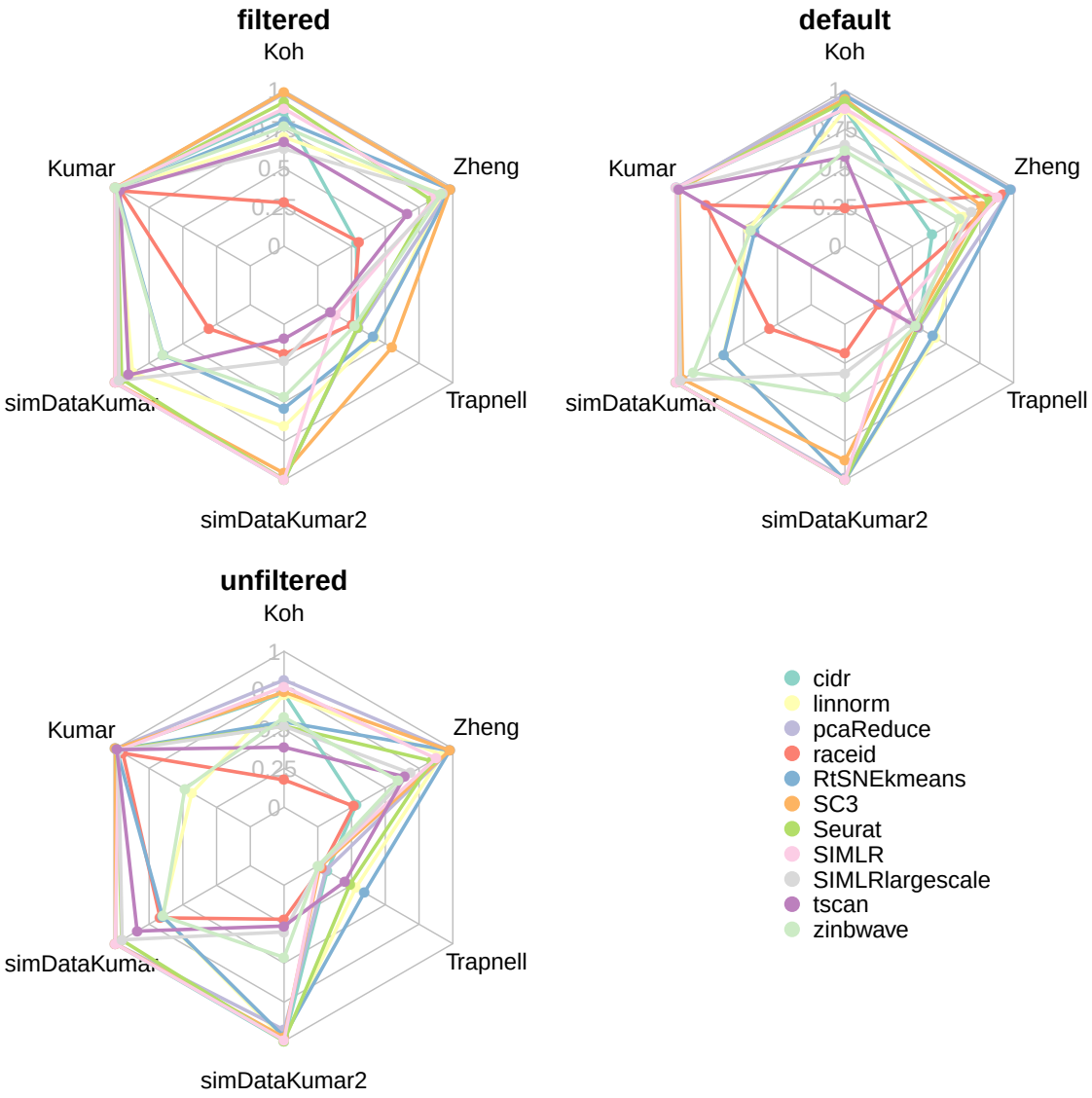


Figure 6: Radar plots showing the ARI scores for the filtered data run with the annotated number of clusters, in the default run mode and with the unfiltered data.

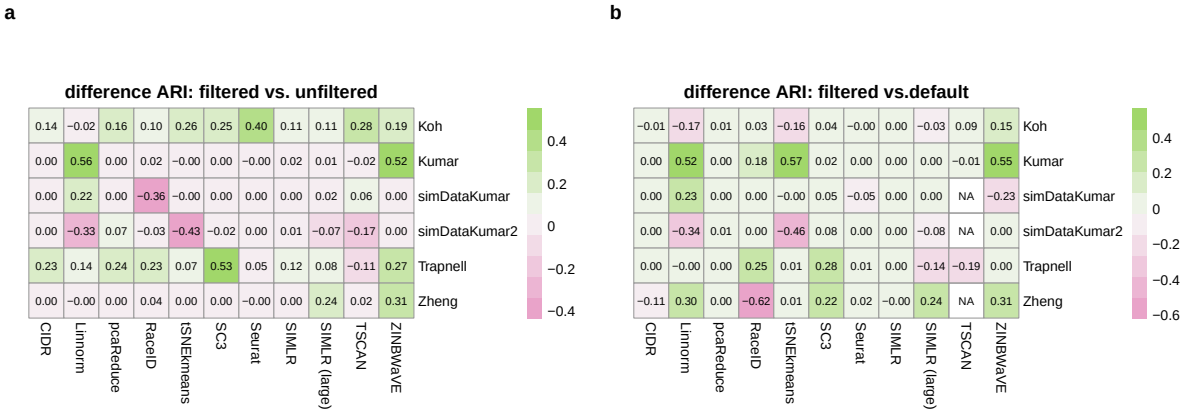


Figure 7: Differences in ARI scores for the datasets Kumar, Trapnell, Zheng, Koh, simDataKumar and simDataKumar2 for the run mode default versus annotated k and filtered versus unfiltered. Note that there were no differences in the parameter setting in the default and filtered run mode for pcaReduce.

Similar to SC3, Seurat achieved high accuracies. The method had the highest average ARI score when running under the default settings. This is particularly interesting if the number of clusters is unknown, as Seurat determines the number of clusters automatically (albeit, it is possible to adjust the granularity of the clusterings). However, Seurat's performance levels dropped when used with the Koh and Zheng datasets. The filtering of the Koh dataset led to an improvement in the accuracies, but otherwise, when using the filtered datasets, only small changes were detected. According to Butler and Satija (2017), the resolution parameter for the function is crucial to the ability to determine the number of clusters, and it is recommended that the method is tested using different values of this parameter. In this study, we were only able to run the methods on a small range of values in the parameter, as else it was not possible to run the method. However, Seurat was able to detect the correct number of subpopulations in most of the datasets and run modes. The exceptions are in the Zheng data, where an additional cluster was detected, and the Koh datasets were the method missed one subpopulation. The default for the number of neighbours for the kNN is set to 30, and no or only small differences in the clustering are achieved when this parameter was set to 10 percent of the dataset (see Figure 7).

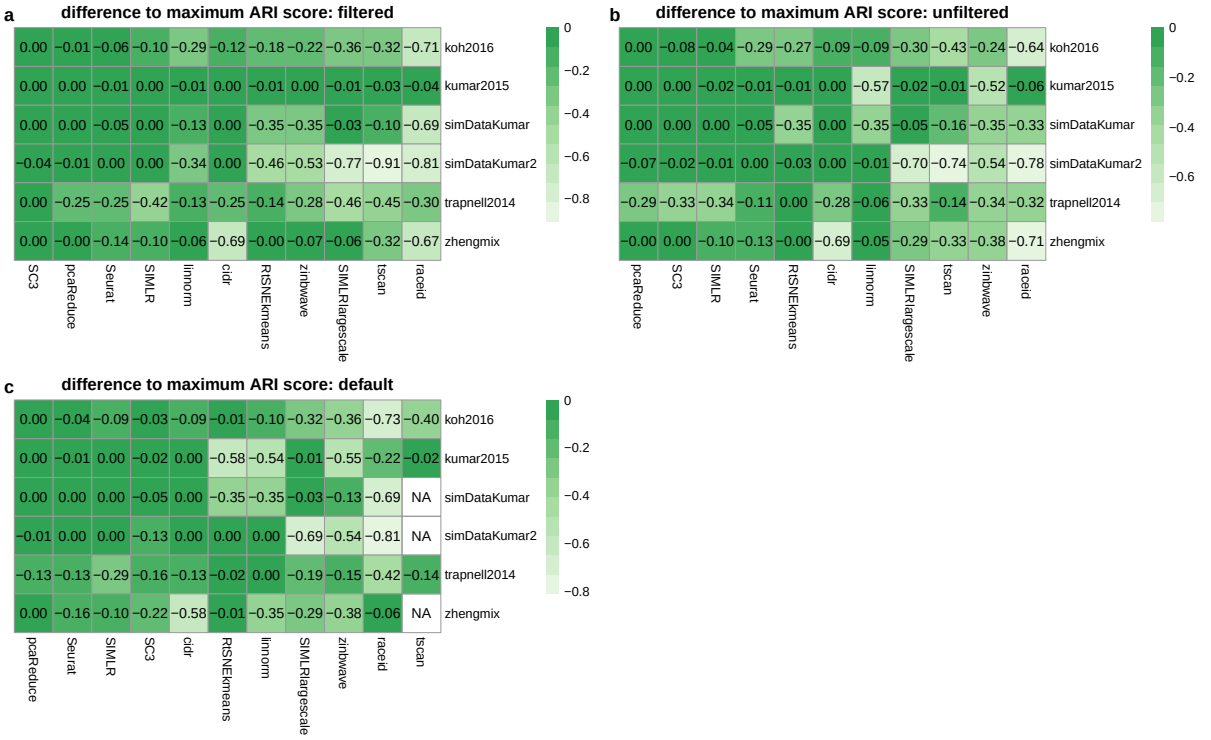


Figure 8: Differences between the method with the maximum ARI scores and the other methods. Shown are the differences in ARI scores for the run mode (a) default, (b) filtered and (c) unfiltered.

Although CIDR showed an overall high level of accuracy, it had one of the lowest performances for the Zheng dataset of all the tested methods. A possible reason for its low performance is that the expression values are UMI counts and the data has a low sequencing depth, leading to a poor model fit in the imputation procedure. Filtering improved this method's performance for two out of the six different datasets; otherwise, it had no impact on the performance of the method. It was attempted to further improve CIDR performance by selecting an appropriate number of PCs; however, no improvements in accuracy were achieved. Except for the Koh and Zheng dataset, the method was able to identify the correct number of subpopulations.

In comparison to the other well-performing methods, SIMLR had similar or higher scores for Koh, Kumar, and the simulated datasets. It dropped in performance for the Trapnell and Zheng data. Filtering the datasets led to an improvement for the Trapnell and Koh datasets. Mean scaling of the dataset, one of the parameter settings, had no impact on the performances. However, it is necessary to do so as else the method is not able to perform the eigenvalue decomposition of the learned similarity matrix.

The method tSNEkmeans achieved similar accuracies compared to the high-performing methods when used with the simple Kumar, simDataKumar, and Zheng datasets, but it showed consistently low accuracies for the Koh dataset. The k-means algorithm assumes spherical clusters which, in the tSNE representation of the dataset Koh, are not given. The influence of a reduced number of PCs as input for the tSNE dimension reduction was dependent on the datasets. However, though this improved the performance of the Kumar data, it had a negative impact on the Koh dataset. Nevertheless, no changes in ARI scores were detected for the other datasets. It is unclear whether gene filtering had a positive or negative influence on the clustering, as it varied highly between the datasets.

The method is unstable, and the changes in the performances could also be due to the stochasticity of the method.

Linnorm's performance is in the medium range. It was able to cluster the Kumar and the Zheng datasets correctly, but was in medium range for the other dataset when compared to well-performing methods as SC3 and Seurat. Changes in the filtered parameters seemed crucial and led to an improvement in the clustering results for the Zheng, Kumar and simDataKumar datasets. However, the score for the Koh dataset was worse. The minimum zero fraction was set to 0.75, which is probably too high if the high dropout rate in the Koh data is considered. However, note that the method was highly unstable, and the results for the different run modes may not be comparable. Linnorm had relatively high accuracies for the unfiltered Trapnell and Zheng datasets. This could be due to the filtering functions that are implemented in the method.

The methods ZINBWaVE and TSCAN only had high accuracies for the Kumar datasets. Filtering of the data improved the accuracies for ZINBWaVE, whereas for TSCAN it only improved the performance for the Koh dataset. The method TSCAN is designed to find trajectories. This type of data is given in the Trapnell data. Here, the method achieved ARI scores between 0.6 and 0.7, which is not better than other methods. In this study, a start and end point of the trajectory could be given which, according to the authors, could improve the results. In this study, however, this was not done. It was not possible to run TSCAN in the default mode for the Zheng and the simulated datasets.

RaceID had the lowest performance and returned only a high ARI score run modes when used with the simple Kumar data. It performed well for the Zheng dataset with the default setting but failed for the other two run modes. If this is due to a poor choice of the parameter settings, or it was due to the stochasticity of the method remains unclear. The method is based on absolute transcript counts which can explain the bad performance in the non-UMI based datasets.

Overall, SC3, pcaReduce, Seurat and SIMLR are the methods with high accuracies. Also, CIDR performed comparable to the before mentioned methods, except when running with UMI counts. Due to the stochasticity of pcaReduce and tSNEkmeans, the results are unstable, and the performance was dependent on the actual run. Linnorm, TSCAN, ZINBWaVE and, in particular, RaceID showed overall low ARI scores. Using filtered data improved the results for the Koh and the Trapnell dataset, with only two exceptions, namely TSCAN and Linnorm. For the other datasets, the gene-wise filtering had different effects on the clustering results. Mostly the ARI scores were stable. The methods Linnorm, RaceID, tSNEkmeans, TSCAN and ZINBWaVE showed strong changes in the ARI scores in at least one of these datasets. Whether these changes are due to the stochasticity of the methods, is unclear.

All methods showed differing ARI scores for at least one dataset when comparing the default mode and when run with the annotated number of clusters. It is not clear whether the methods perform better under the default settings or with a changed parameter setting. As stated above, especially for tSNEkmeans, RaceID and Linnorm, any changes in the clustering could be due to the stochasticity of the methods. It was also highly dependent on the dataset, and for most methods, it only had a slight impact, negative or positive, on the ARI score. However, we note that for SC3, the ARI scores were higher when the number of clusters is given.

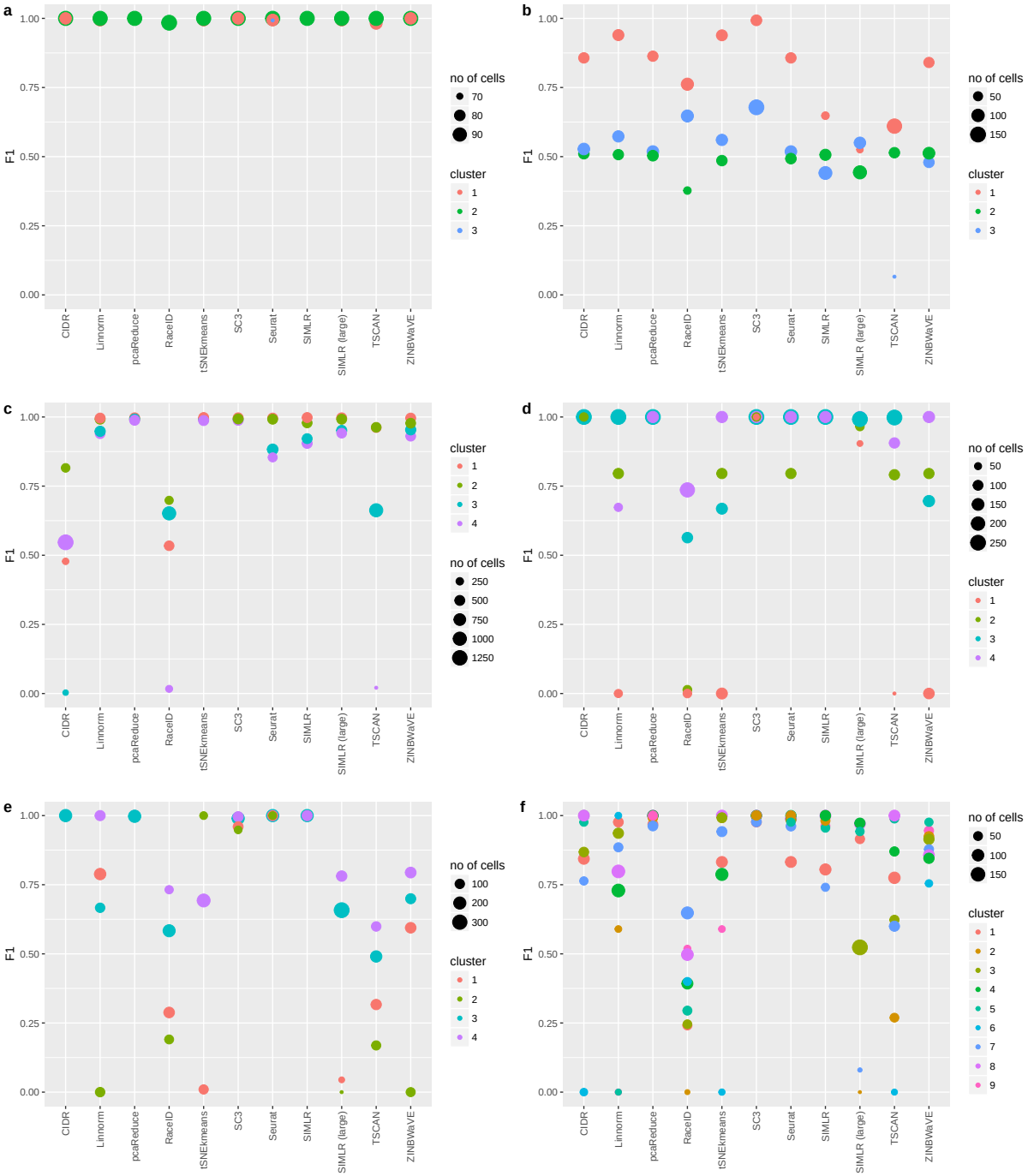


Figure 9: F1 scores for the filtered datasets and the annotated number of clusters. Shown are the datasets Kumar (a), Trapnell (b), Zheng (c), Koh (d), simDataKumar (e) and simDataKumar (f). Each method has a dot for a cluster. The number of cells is indicated by the size of the dots.

3.3 Range of clusters

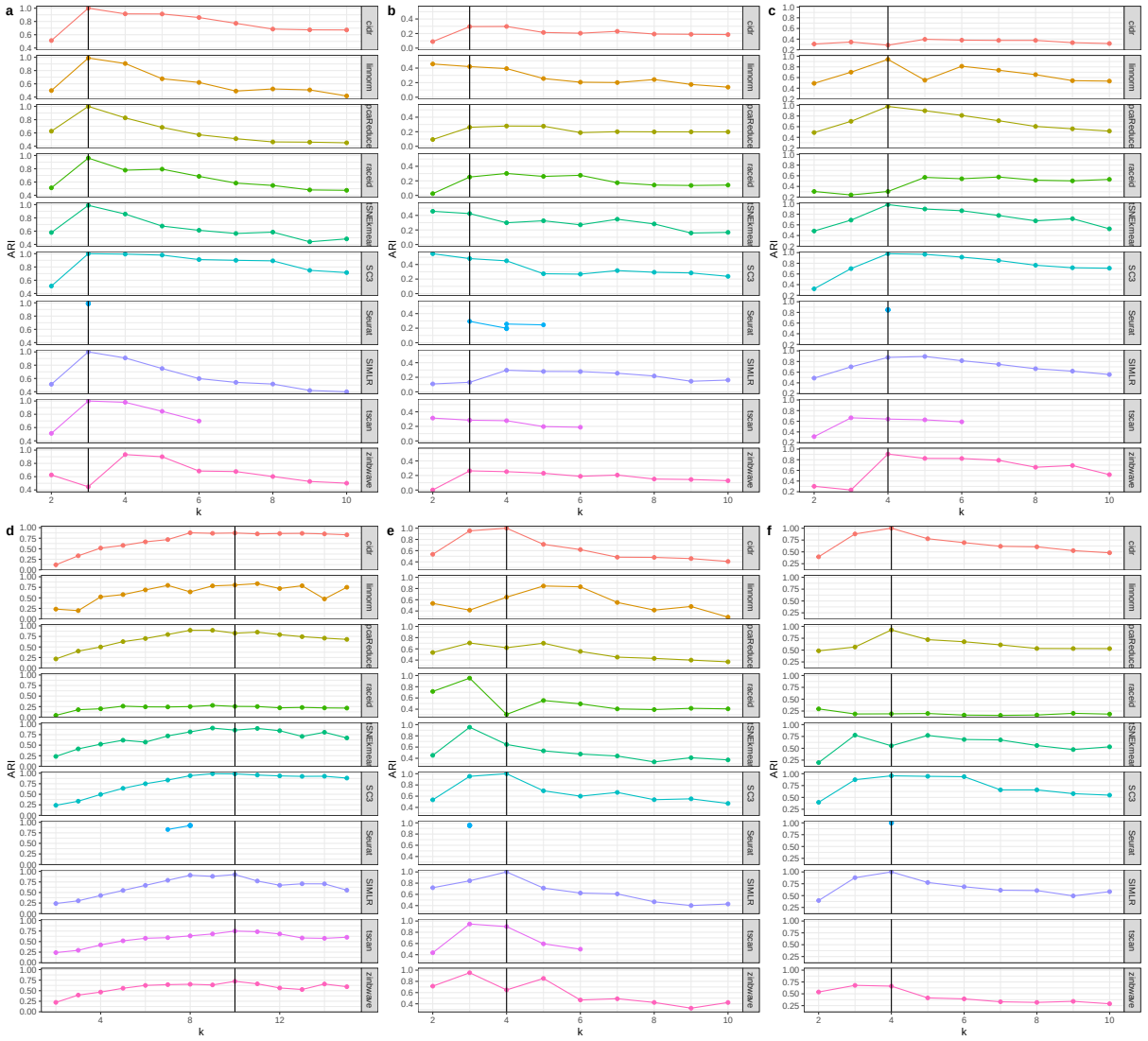


Figure 10: ARI scores for range of parameters for the datasets Kumar (a), Trapnell (b), Zheng (c), Koh (d), simDataKumar (e) and simDataKumar (f). Shown are the methods where the number of cluster could be defined. Seurat didn't allow the direct control of the number of clusters. The Kumar, Trapnell, Zheng, Koh and simDataKumar and simDataKumar2 had 3, 3, 4, 10, 4 and 4 clusters, respectively. The vertical line indicates the annotated number of clusters.

The methods were run under a range of the number of clusters, and for each clustering result, the ARI score was computed. Seurat did not allow for direct control of the number of clusters. Instead, the resolution parameter was used. However, it was only possible to run the method on a small range of the resolution parameter. The results are shown in Figure 10. The methods behaved differently depending on the difficulty of the datasets and the number of subpopulations. For example, most of the methods had clear maximums when used with the simple Kumar dataset, whereas, when used with the more difficult Koh dataset, the methods showed a monotonic increase in the ARI and reached a plateau within five to ten clusters. For the Trapnell data, SC3, Linnorm, TSCAN and tSNEkmeans had clear maximum values with two clusters and then had a decrease in

the scores for a higher number of clusters. The methods CIDR, pcaReduce, RaceID and SIMLR had no clear maximums, and all had a plateau between two to five clusters. For the Zheng dataset, tSNEkmeans, SC3, Linnorm, ZINBWaVE and SIMLR showed a similar behavior with a clear maximum with four clusters. The other methods had no such clear maximum values, or the maximum score is not at the annotated four clusters. The methods tSNEkmeans and SC3 showed a similar behavior for the six different datasets.

3.4 Stability analysis

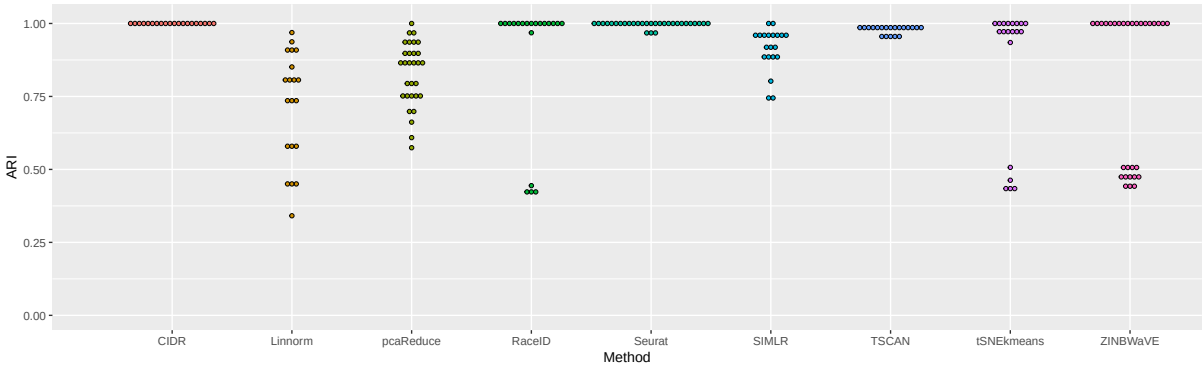


Figure 11: Stability analysis results with 20 subsamples (n=100) for the Kumar dataset. The method SC3 is missing as it was not possible to run the method due to the high run time.

Subsampling without replacement was used to assess the stability of the methods. Based on the wide range of algorithms for the methods, the methods showed results with varying levels of stability (see Figure 11). The deterministic method CIDR is stable. Seurat and TSCAN were mostly stable and showed some outlying runs with a slight decrease in the ARI scores. This is in contrast to RaceID and tSNEkmeans which had some strong outlying runs. pcaReduce and Linnorm are both very unstable, and the assignment of the cells to the respective cluster varied greatly. Also, ZINBWaVE is unstable, notably not because of the method itself, but due to the use of the k-means algorithm for clustering. For this study, the simple Kumar dataset is used, and it can be expected that the methods will behave even more unstable with a more difficult dataset.

3.5 Runtime

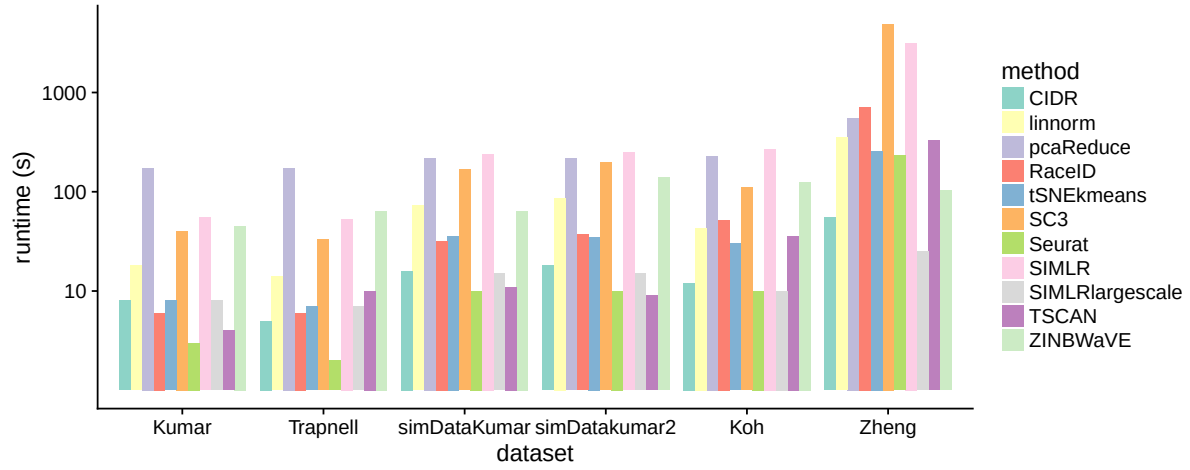


Figure 12: Runtime (s) on a log-scale for the methods on the datasets. Shown are the datasets Kumar, Trapnell, Zheng, Koh, simDataKumar and simDataKumar2 ordered by the number of cells in each dataset. The datasets are filtered and the number of clusters based on the ground truth is used.

The run times for each method can be seen in Figure 12. They were highly different, ranging across two magnitudes for the studied methods. The fastest methods were Seurat, CIDR, and SIMLR (large-scale), whereas the methods pcaReduce, SIMLR, RaceID, and SC3 showed the highest run times. It is notable that SC3 and SIMLR both have a non-linear increase in run time, making their use unfeasible with a larger dataset consisting of thousands of cells. For an improved run time, Kiselev et al. (2017) recommend the use of support vector machines when using SC3 and bigger data sets ($>$ 5'000 cells). For small-scale datasets, Seurat is one of the fastest methods. However, when used with the larger Zheng dataset (with 2'000 cells), its run time lies in the middle ground. Figure 13 shows the average run time and ARI scores for the datasets. SC3 and pcaReduce showed high accuracies, but suffered from high run times. If the run time is included as an evaluation criterion, Seurat shows a good combination of high accuracies and a fast run time.

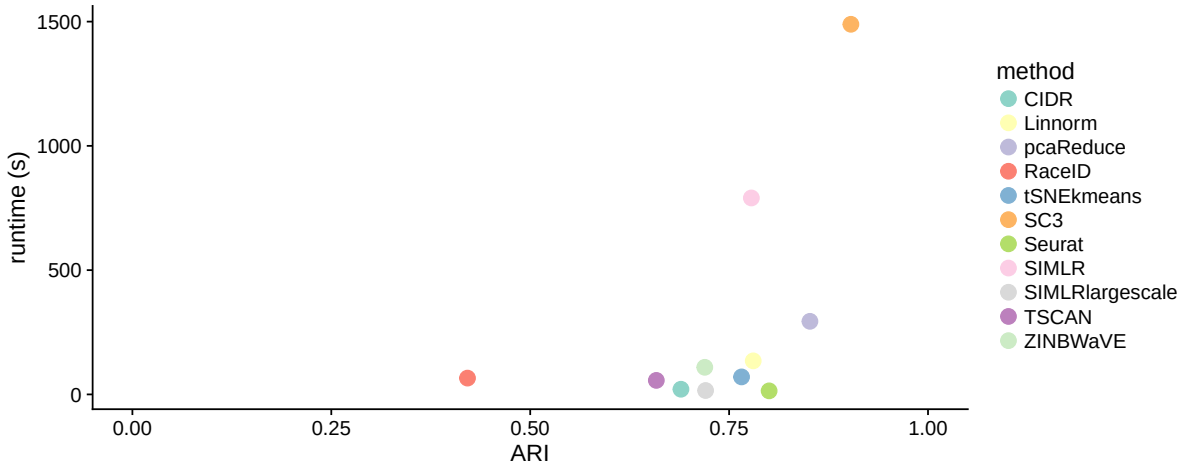


Figure 13: Average runtime and ARI scores for the filtered datasets. Shown is the average runtime and average ARI score for the datasets Kumar, Koh, Trapnell, Zheng and the simulated datasets.

4 Conclusion and Outlook

SC3 showed, on average, one of the best performance in clustering scRNA-seq data, but also the highest run time. Despite this, the method is still well suited for datasets with hundreds of cells. For bigger datasets, it is possible to improve runtime with an alternative approach. However, this was not tested in this study, and the investigation of its performance would be interesting. Seurat reached similar performance levels to SC3 and had the advantage of a faster run time. It is not possible to define the number of clusters, which, depending on the research question, can be disadvantageous. Other high performing methods were pcaReduce, SIMLR and CIDR. CIDR is not as flexible regarding the transcript counts and fails if UMI counts are used, whereas pcaReduce suffers from instability. Gene filtering can have an impact on performance and is recommended, especially for datasets with more complex clustering structures. This also applies to methods that already include gene filtering functions (i.e. SC3 and Seurat).

In this study, the high-performing methods achieved almost perfect results for many of the datasets. This hinders a more rigorous comparison of these methods. For future analysis, the inclusion of more difficult datasets could give a better insight into the performance of these methods. The simulations included in the study used non-overlapping differentially expressed gene populations. More realistic simulations could include subpopulations with overlapping DE genes between the populations. Of possible interest is also the impact of different types of expression values.

The instability of tSNEkmeans, pcaReduce, ZINBWaVE, and Linnorm is an obstacle for the comparison of the methods. When used in a comparison study and the annotation of the cells is known, it is suggested to rerun the method several times and report the average as the evaluation metric. Also, note that, for tSNEkmeans and ZINBWaVE, clustering via k-means was used; for future evaluation studies, a different clustering approach should be considered. For example, Perraudeau et al. (2017) recently suggested k-means clustering with resampling for the method ZINBWaVE. The field of scRNA-seq is a rapidly developing discipline, and there remain several promising clustering methods worth investigating (van Dijk et al., 2017; Sinha et al., 2018; Sun et al., 2017; Yang et al., 2017).

References

Anders, S. and Huber, W. (2010). Differential expression analysis for sequence count data. *Genome biology*, 11(10):R106.

Andrews, T. S. and Hemberg, M. (2017). Identifying cell populations with scRNASeq. *Molecular aspects of medicine*.

Butler, A. and Satija, R. (2017). Integrated analysis of single cell transcriptomic data across conditions, technologies, and species. *bioRxiv*, page 164889.

Grün, D., Lyubimova, A., Kester, L., Wiebrands, K., Basak, O., Sasaki, N., Clevers, H., and van Oudenaarden, A. (2015). Single-cell messenger RNA sequencing reveals rare intestinal cell types. *Nature*, 525(7568):251.

Hartigan, J. A. and Wong, M. A. (1979). Algorithm as 136: A k-means clustering algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 28(1):100–108.

Hotelling, H. (1933). Analysis of a complex of statistical variables into principal components. *Journal of educational psychology*, 24(6):417.

Hubert, L. and Arabie, P. (1985). Comparing partitions. *Journal of classification*, 2(1):193–218.

Ji, Z. and Ji, H. (2015). *TSCAN: Tools for Single-Cell ANalysis*. R package version 1.16.0.

Jolliffe, I. T. (1986). Principal component analysis and factor analysis. In *Principal component analysis*, pages 115–128. Springer.

Kiselev, V. Y., Kirschner, K., Schaub, M. T., Andrews, T., Yiu, A., Chandra, T., Natarajan, K. N., Reik, W., Barahona, M., Green, A. R., et al. (2017). SC3: consensus clustering of single-cell RNA-seq data. *Nature methods*.

Koh, P. W., Sinha, R., Barkal, A. A., Morganti, R. M., Chen, A., Weissman, I. L., Ang, L. T., Kundaje, A., and Loh, K. M. (2016). An atlas of transcriptional, chromatin accessibility, and surface marker changes in human mesoderm development. *Scientific data*, 3:160109.

Kumar, R. M., Cahan, P., Shalek, A. K., Satija, R., DaleyKeyser, A. J., Li, H., Zhang, J., Pardee, K., Gennert, D., Trombetta, J. J., et al. (2014). Deconstructing transcriptional heterogeneity in pluripotent stem cells. *Nature*, 516(7529):56–61.

Lin, P., Troup, M., and Ho, J. W. (2017). CIDR: Ultrafast and accurate clustering through imputation for single-cell RNA-seq data. *Genome biology*, 18(1):59.

Lun, A. T., Bach, K., and Marioni, J. C. (2016a). Pooling across cells to normalize single-cell RNA sequencing data with many zero counts. *Genome biology*, 17(1):75.

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

Perraudeau, F., Risso, D., Street, K., Purdom, E., and Dudoit, S. (2017). Bioconductor workflow for single-cell RNA sequencing: Normalization, dimensionality reduction, clustering, and lineage inference. *F1000Research*, 6.

Picelli, S., Björklund, Å. K., Faridani, O. R., Sagasser, S., Winberg, G., and Sandberg, R. (2013). Smart-seq2 for sensitive full-length transcriptome profiling in single cells. *Nature methods*, 10(11):1096–1098.

R Core Team (2013). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Risso, D., Perraudeau, F., Gribkova, S., Dudoit, S., and Vert, J.-P. (2017). ZINB-WaVE: A general and flexible method for signal extraction from single-cell RNA-seq data. *bioRxiv*, page 125112.

Sinha, D., Kumar, A., Kumar, H., Bandyopadhyay, S., and Sengupta, D. (2018). dropClust: Efficient clustering of ultra-large scRNA-seq data. *Nucleic Acids Research*.

Soneson, C., Love, M. I., and Robinson, M. D. (2015). Differential analyses for RNA-seq: transcript-level estimates improve gene-level inferences. *F1000Research*, 4.

Sun, Z., Wang, T., Deng, K., Wang, X.-F., Lafyatis, R., Ding, Y., Hu, M., and Chen, W. (2017). DIMM-SC: a dirichlet mixture model for clustering droplet-based single cell transcriptomic data. *Bioinformatics*, 34(1):139–146.

Tang, F., Barbacioru, C., Wang, Y., Nordman, E., Lee, C., Xu, N., Wang, X., Bodeau, J., Tuch, B. B., Siddiqui, A., et al. (2009). mRNA-Seq whole-transcriptome analysis of a single cell. *Nature methods*, 6(5):377–382.

Trapnell, C., Cacchiarelli, D., Grimsby, J., Pokharel, P., Li, S., Morse, M., Lennon, N. J., Livak, K. J., Mikkelsen, T. S., and Rinn, J. L. (2014). The dynamics and regulators of cell fate decisions are revealed by pseudotemporal ordering of single cells. *Nature biotechnology*, 32(4):381–386.

van der Maaten, L. (2013). Barnes-hut-SNE. *arXiv preprint arXiv:1301.3342*.

van der Maaten, L. and Hinton, G. (2008). Visualizing data using t-SNE. *Journal of machine learning research*, 9(Nov):2579–2605.

van Dijk, D., Nainys, J., Sharma, R., Kathail, P., Carr, A. J., Moon, K. R., Mazutis, L., Wolf, G., Krishnaswamy, S., and Pe'er, D. (2017). MAGIC: A diffusion-based imputation method reveals gene-gene interactions in single-cell RNA-sequencing data. *BioRxiv*, page 111591.

Wagner, A., Regev, A., and Yosef, N. (2016). Revealing the vectors of cellular identity with single-cell genomics. *Nature biotechnology*, 34(11):1145–1160.

Wang, B., Zhu, J., Pierson, E., Ramazzotti, D., and Batzoglou, S. (2017). Visualization and analysis of single-cell RNA-seq data by kernel-based similarity learning. *Nature Methods*, 14(4):414–416.

Yang, Y., Huh, R., Culpepper, H. W., Love, M. I., and Li, Y. (2017). SAFE-clustering: Single-cell aggregated (from ensemble) clustering for single-cell RNA-seq data. *bioRxiv*, page 215723.

Yip, S. H., Wang, P., Kocher, J.-P. A., Sham, P. C., and Wang, J. (2017). Linnorm: improved statistical analysis for single cell RNA-seq expression data. *Nucleic acids research*.

Zappia, L., Phipson, B., and Oshlack, A. (2017). Splatter: simulation of single-cell rna sequencing data. *Genome biology*, 18(1):174.

Zheng, G. X., Terry, J. M., Belgrader, P., Ryvkin, P., Bent, Z. W., Wilson, R., Ziraldo, S. B., Wheeler, T. D., McDermott, G. P., Zhu, J., et al. (2017). Massively parallel digital transcriptional profiling of single cells. *Nature communications*, 8:14049.

Žurauskienė, J. and Yau, C. (2016). pcaReduce: hierarchical clustering of single cell transcriptional profiles. *BMC Bioinformatics*, 17(1):140.

Zwiener, I., Frisch, B., and Binder, H. (2014). Transforming RNA-Seq data to improve the performance of prognostic gene signatures. *PloS one*, 9(1):e85150.

A Additional information

A.1 F1 scores for the for the run modes with the unfiltered datasets and in the default mode

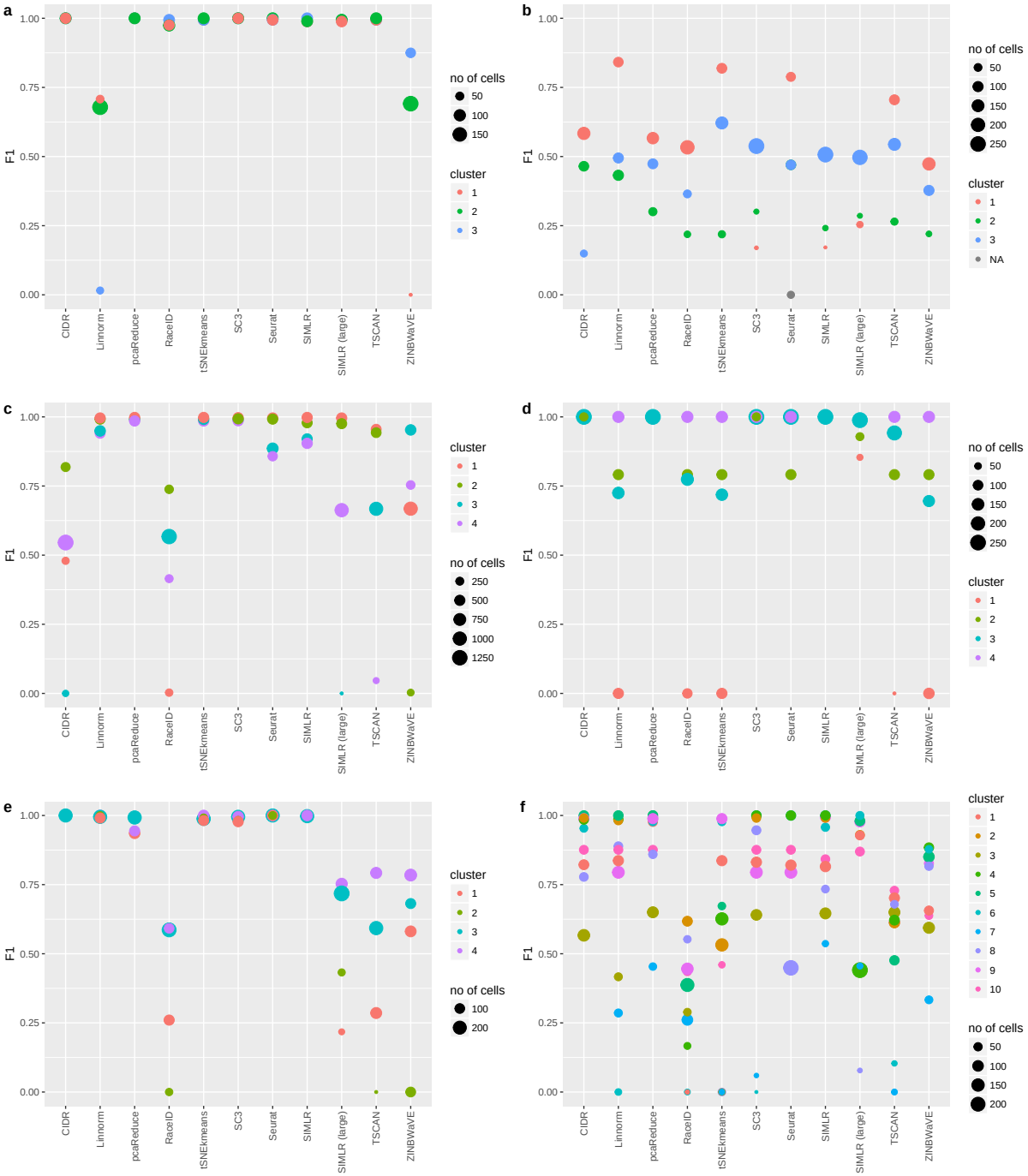


Figure 14: F1 scores for the unfiltered datasets and the annotated number of clusters. Shown are the datasets Kumar (a), Trapnell (b), Zheng (c), Koh (d), simDataKumar (e) and simDataKumar2 (f). Each method has a dot for a cluster. The number of cells is indicated by the size of the dots.

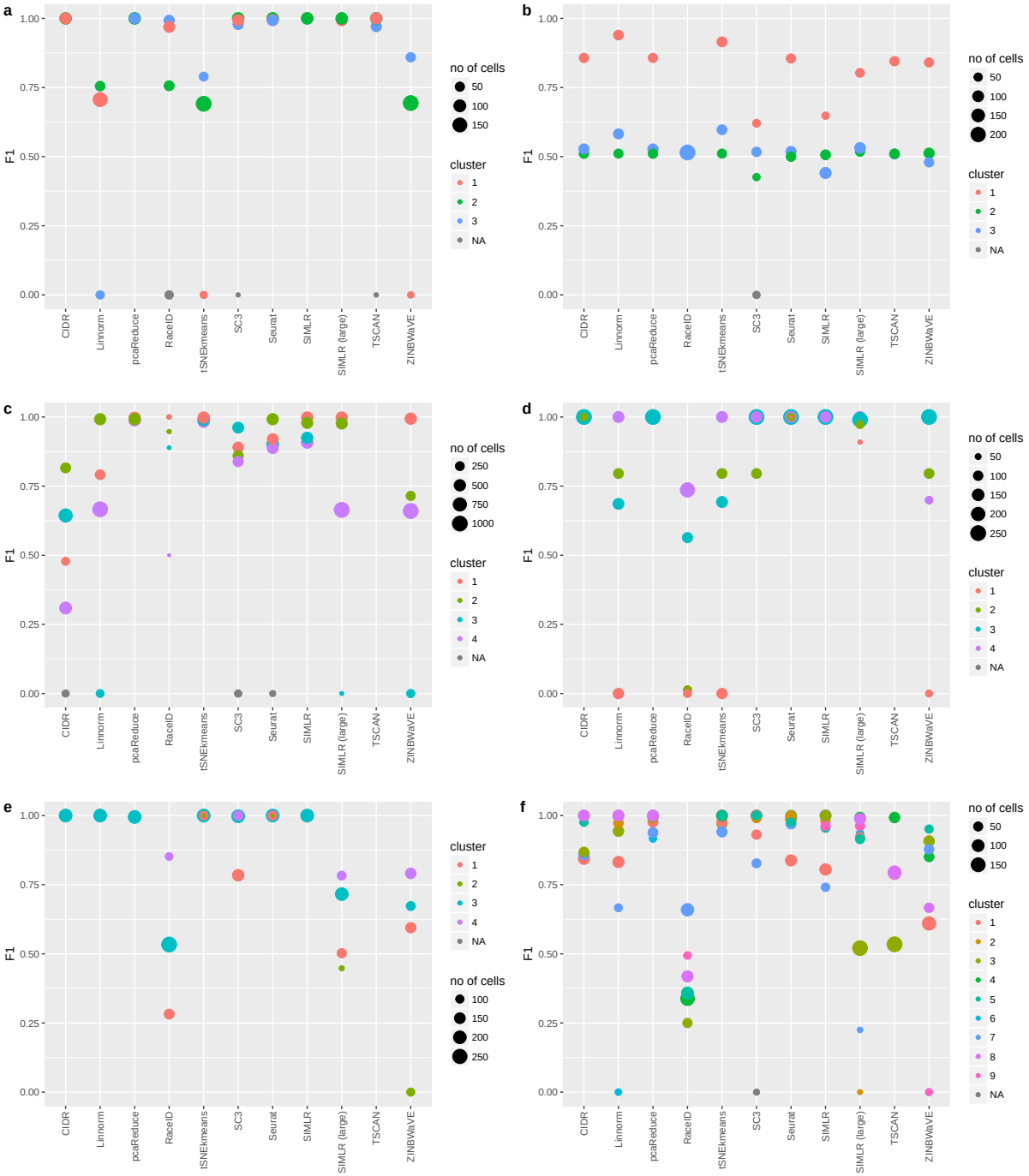


Figure 15: F1 scores for the datasets run in the default mode. Shown are the datasets Kumar (a), Trapnell (b), Zheng (c), Koh (d), simDataKumar (e) and simDataKumar2 (f). Each method has a dot for a cluster. The number of cells is indicated by the size of the dots.

A.2 QC plots for the datasets

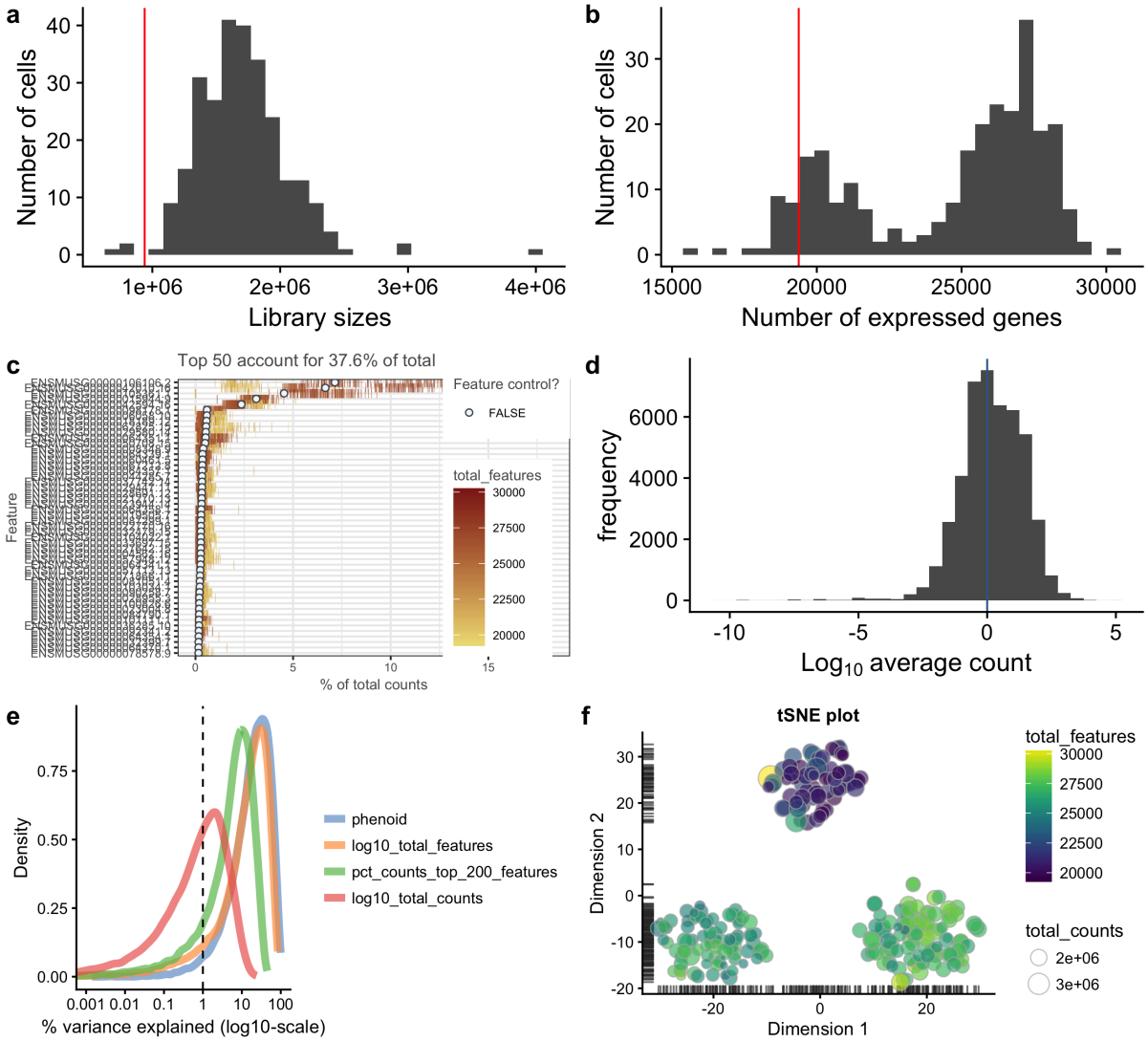


Figure 16: QC summary of Kumar. Shown are the histograms of library size (a) and the number of expressed genes (b). (c) Percent of total counts of the highest expressed genes. (d) Histogram of the average counts per gene on a log-scale. (e) Density plot of the percentage of variance explained by different factors. (f) t-SNE plots of the log-normalised counts, indicated are the total counts and total features. The vertical lines are the filter thresholds.

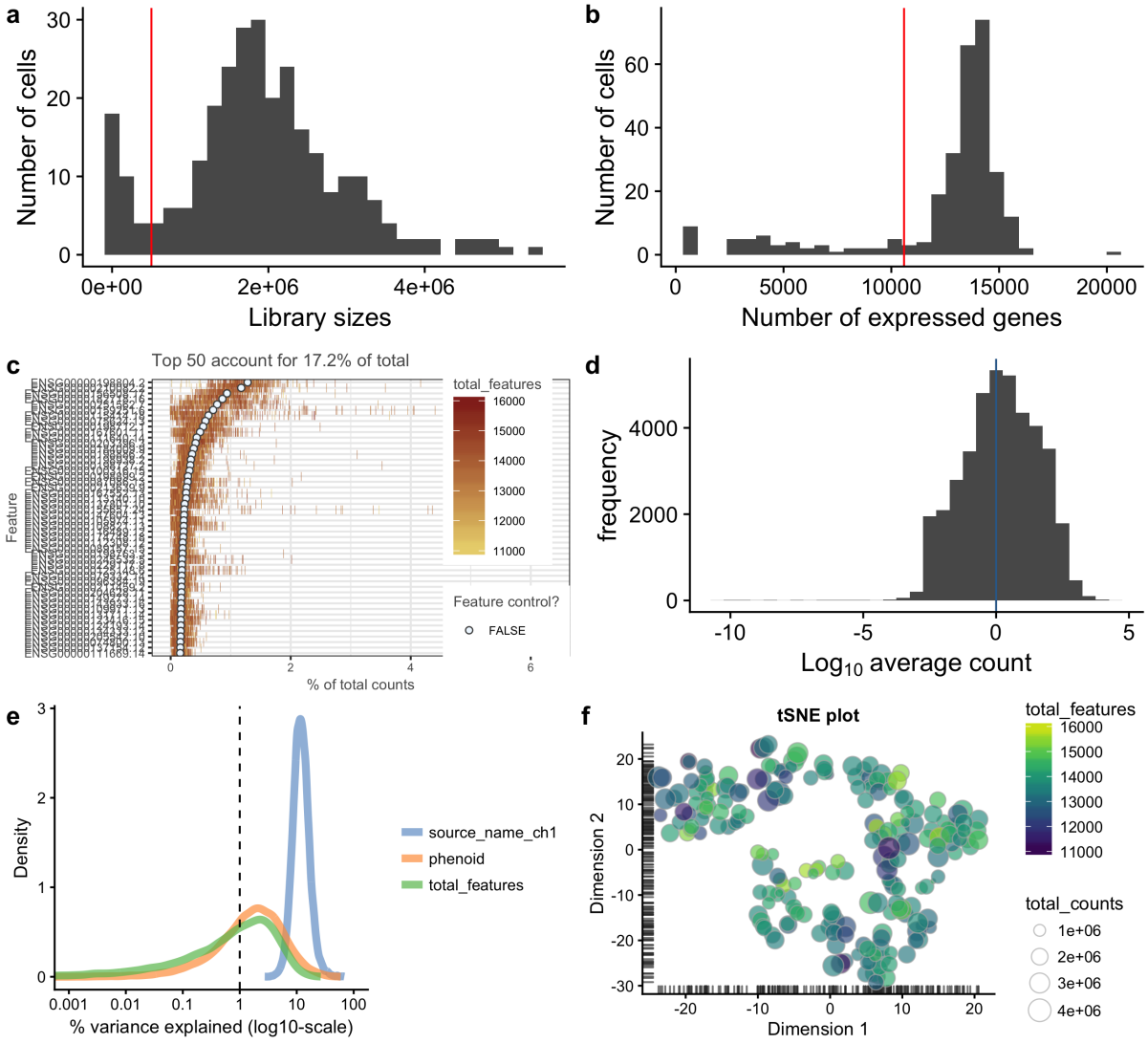


Figure 17: QC summary of Trapnell. Shown are the histograms of library size (a) and the number of expressed genes (b). (c) Percent of total counts of the highest expressed genes. (d) Histogram of the average counts per gene on a log-scale. (e) Density plot of the percentage of variance explained by different factors. (f) t-SNE plots of the log-normalised counts, indicated are the total counts and total features. The vertical lines are the filter thresholds.

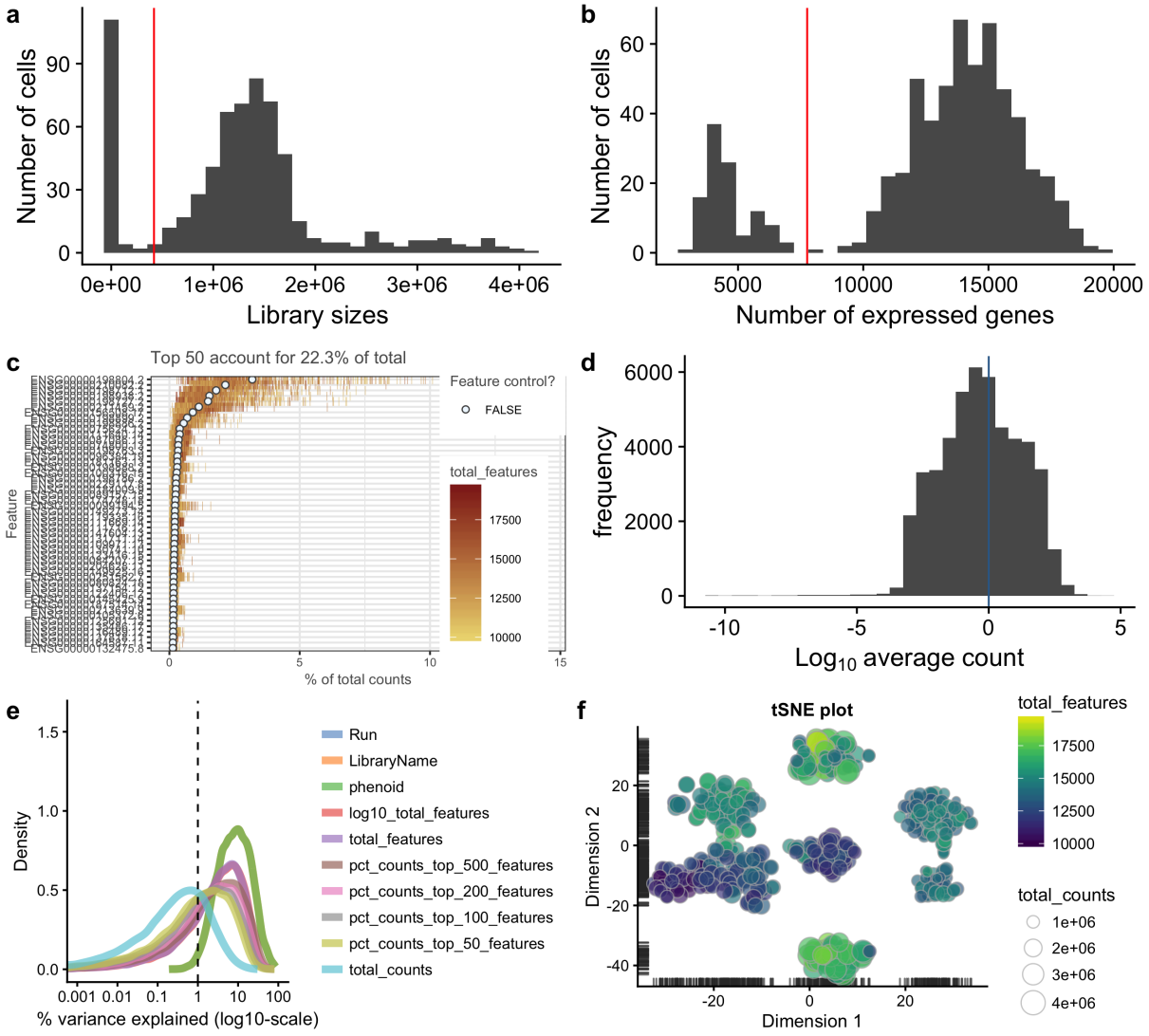


Figure 18: QC summary of Koh. Shown are the histograms of library size (a) and the number of expressed genes (b). (c) Percent of total counts of the highest expressed genes. (d) Histogram of the average counts per gene on a log-scale. (e) Density plot of the percentage of variance explained by different factors. (f) t-SNE plots of the log-normalised counts, indicated are the total counts and total features. The vertical lines are the filter thresholds.

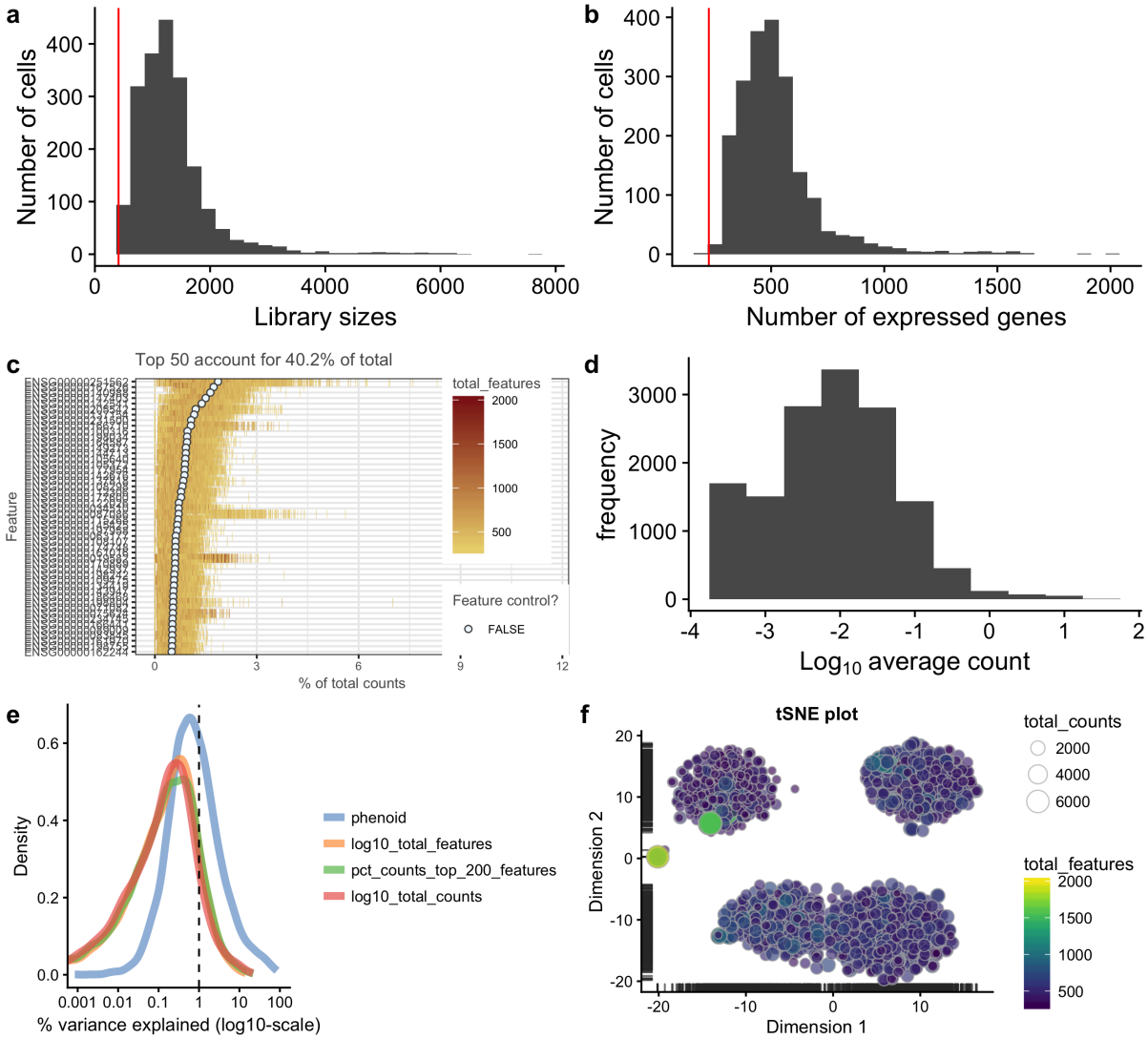


Figure 19: QC summary of Zheng. Shown are the histograms of library size (a) and the number of expressed genes (b). (c) Percent of total counts of the highest expressed genes. (d) Histogram of the average counts per gene on a log-scale. (e) Density plot of the percentage of variance explained by different factors. (f) t-SNE plots of the log-normalised counts, indicated are the total counts and total features. The vertical lines are the filter thresholds.

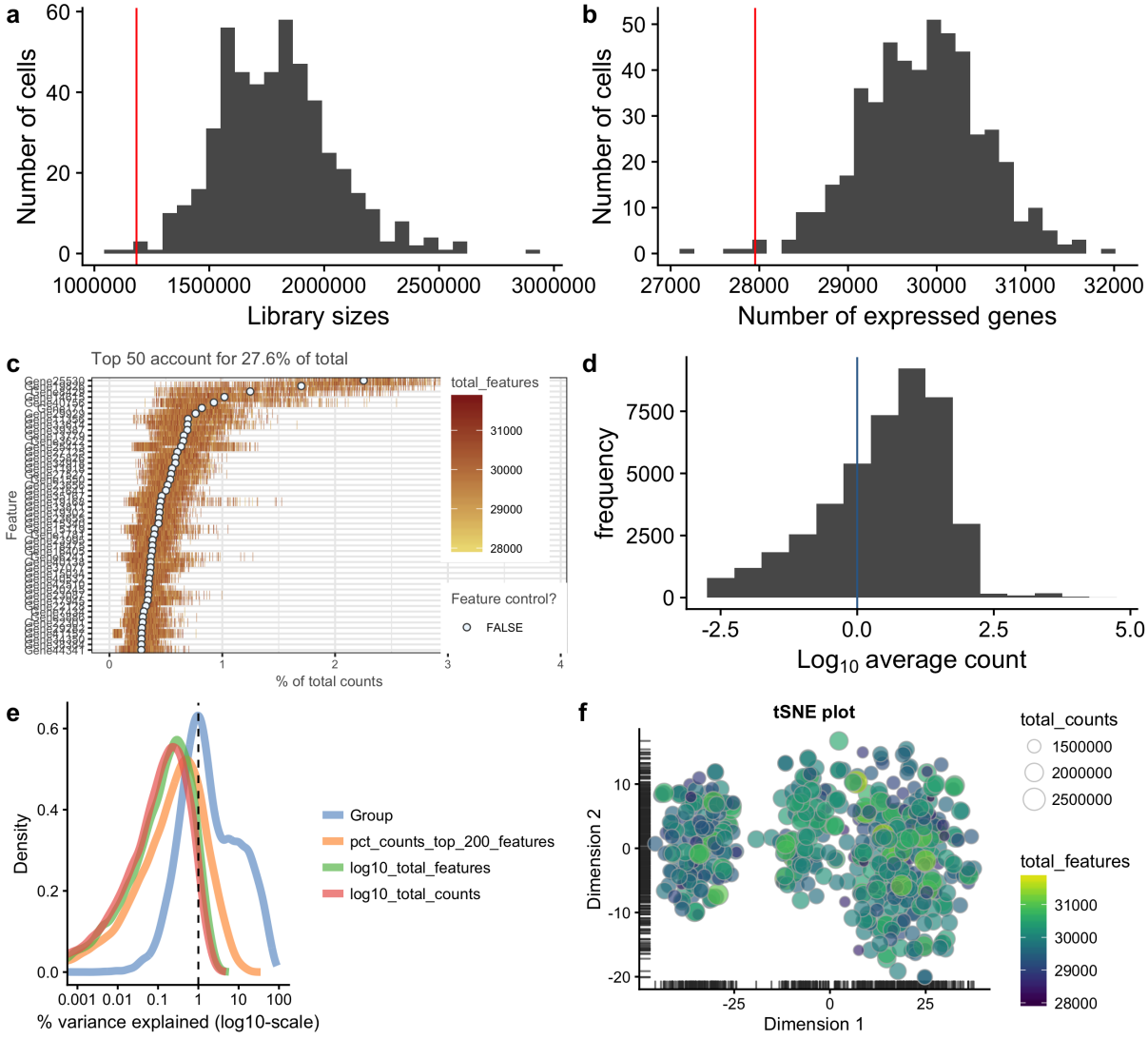


Figure 20: QC summary of simDataKumar. Shown are the histograms of library size (a) and the number of expressed genes (b). (c) Percent of total counts of the highest expressed genes. (d) Histogram of the average counts per gene on a log-scale. (e) Density plot of the percentage of variance explained by different factors. (f) t-SNE plots of the log-normalised counts, indicated are the total counts and total features. The vertical lines are the filter thresholds.

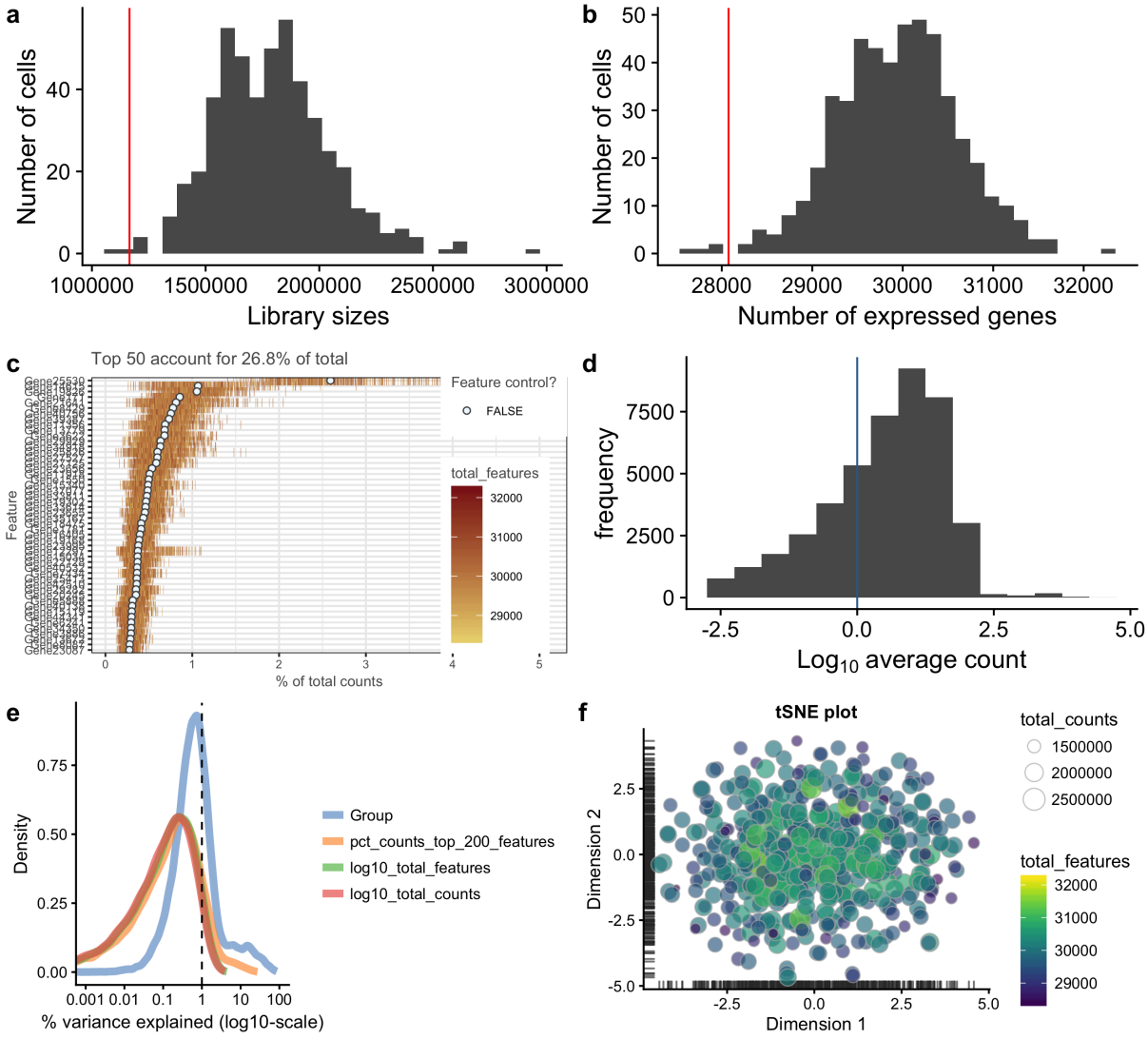


Figure 21: QC summary of simDataKumar2. Shown are the histograms of library size (a) and the number of expressed genes (b). (c) Percent of total counts of the highest expressed genes. (d) Histogram of the average counts per gene on a log-scale. (e) Density plot of the percentage of variance explained by different factors. (f) t-SNE plots of the log-normalised counts, indicated are the total counts and total features. The vertical lines are the filter thresholds.