GigaScience, 8, 2019, 1–6

doi: 10.1093/gigascience/giz087
Technical Note

TECHNICAL NOTE

ascend: R package for analysis of single-cell RNA-seq data

Anne Senabouth[1], Samuel W. Lukowski[2], Jose Alquicira Hernandez[1,2], Stacey B. Andersen[2], Xin Mei[2,3], Quan H. Nguyen[2,†] and Joseph E. Powell[1,4,5,*,†]

[1]Garvan Institute of Medical Research, 384 Victoria Street, Darlinghurst, Sydney, Australia 2010; [2]Institute of Molecular Bioscience, 306 Carmody Road, St Lucia, University of Queensland, Brisbane, Australia 4072; [3]South China Botanical Garden, Chinese Academy of Sciences, Guangzhou, China; [4]School of Medical Sciences, 18 High Street, University of New South Wales, Kensington, Sydney, Australia, 2052 and [5]Garvan-Weizmann Centre for Cellular Genomics, Garvan Institute of Medical Research, 384 Victoria Street, Darlinghurst, Sydney, Australia, 2010

[*]Correspondence author. Joseph E. Powell. Garvan Institute of Medical Research, 384 Victoria Street, Darlinghurst, Sydney, Australia, 2010. E-mail: j.powell@garvan.org.au http://orcid.org/0000-0001-9031-6356
[†]Joint senior authors

Abstract

Background: Recent developments in single-cell RNA sequencing (scRNA-seq) platforms have vastly increased the number of cells typically assayed in an experiment. Analysis of scRNA-seq data is multidisciplinary in nature, requiring careful consideration of the application of statistical methods with respect to the underlying biology. Few analysis packages exist that are at once robust, are computationally fast, and allow flexible integration with other bioinformatics tools and methods. **Findings:** *ascend* is an R package comprising tools designed to simplify and streamline the preliminary analysis of scRNA-seq data, while addressing the statistical challenges of scRNA-seq analysis and enabling flexible integration with genomics packages and native R functions, including fast parallel computation and efficient memory management. The package incorporates both novel and established methods to provide a framework to perform cell and gene filtering, quality control, normalization, dimension reduction, clustering, differential expression, and a wide range of visualization functions. **Conclusions:** *ascend* is designed to work with scRNA-seq data generated by any high-throughput platform and includes functions to convert data objects between software packages. The *ascend* workflow is simple and interactive, as well as suitable for implementation by a broad range of users, including those with little programming experience.

Keywords: single cell; scRNA-seq; filtering; clustering; normalization; differential expression; data visualization; R package

Key Points

- *ascend* is a fast and easy-to-use software for thorough and interactive analysis of single-cell RNA sequencing data.
- *ascend*'s streamlined workflow includes filtering, normalization, dimension reduction, clustering, differential expression, and visualization.
- *ascend* optimizes parallelization and algorithms for improving speed of each analysis step (e.g., differential expression analysis).
- *ascend* implements Clustering by Optimal REsolution (CORE) for unsupervised, robust hierarchical clustering.

Received: 25 April 2019; **Revised:** 30 May 2019; **Accepted:** 27 June 2019

Findings

Background

Single-cell RNA sequencing (scRNA-seq) has revolutionized the way we understand the transcriptional programs of cells. Recent advances in barcoding molecular biology techniques, coupled with microfluidics, have yielded platforms such as 10× Genomics Chromium [1] and Drop-seq [2], which are capable of capturing the transcriptomes of tens of thousands of single cells simultaneously. The increased capacity of scRNA-seq has been advantageous as larger sample sizes provide greater statistical power and correspondingly higher resolution to determine differences in cellular features. A consequence has been the increase in the complexity of scRNA-seq data, creating new challenges for data management, statistical methods, data visualization, and computing strategies. A number of scRNA-seq specific methods and toolkits have been developed to address these challenges [3–5], but both the functionality and specific methods implemented vary. It is becoming apparent that for a given scRNA-seq data set, the specific analysis steps need to be carefully considered in light of the underlying biology. For single-cell analysis packages, flexibility in both the choice of methods implemented and arguments passed to functions is therefore important.

Here we present *ascend*, an R package designed to create a simple and streamlined workflow for the analysis of scRNA-seq experiments. Fig. 1 *ascend* is designed to handle data generated from any single-cell library preparation platform; this can include data from single and paired-end reads and, optionally, with unique molecular identifiers (UMIs). *ascend* imports scRNA-seq data following the generation of an expression matrix consisting of transcript counts from each cell and performs user-friendly quality control, filtering, normalization, dimension reduction, clustering, differential expression, and visualization. It includes functions to leverage multiple CPUs, allowing most analyses to be performed on a standard desktop or laptop.

Data object

The foundation of the *ascend* R package is the **E**xpression and **M**etadata **Set**, a data container class that inherits from the SingleCellExperiment superclass [6]. The SingleCellExperiment class, from the Bioconductor R package of the same name, was introduced as a container class specifically for single-cell genomics data. It is structured in the context of the gene-cell expression matrix and contains slots that can hold data that may be used in scRNA-seq analysis—specifically spike-in information, normalization factors, transformations of the original count data, metadata, and data related to cells and genes Fig. 1.

The EMSet deviates from the SingleCellExperiment in which it is a dynamic element. The object is always accompanied by a set of quality control metrics that is reflective of the data that are currently stored in the counts slot of the object. These values are automatically recalculated by the package whenever changes are made to the count matrix, which occurs during batch normalization and filtering. Another feature of the EMSet is the logging of operations, ensuring analysis is performed in the correct order and allowing users to review changes. As metadata can play a key role in functions such as plotting and differential expression analysis, we have separated cell-related and gene-related metadata from calculated values by storing them in dedicated slots introduced by the EMSet. Additional slots have also been introduced to store objects related to clustering and differential expression analysis.

The EMSet retains the convenient row and column subsetting operations of the SingleCellExperiment and introduces methods to manipulate the object based on conditions defined in the cell metadata slot. To ensure compatibility with other software packages that also use the SingleCellExperiment class, a conversion function is supplied to preserve data stored in EMSet-specific slots. These data can then be retrieved when converting back to an EMSet.

Batch normalization

Typically, samples comprising libraries of thousands of cells are often processed in separate batches, and the resulting data require aggregation before analysis. This can introduce systematic biases due to technical variation in each batch. To address this, *ascend* provides simple and fast methods to normalize between batches (*normaliseBatches*). To perform fast batch-to-batch normalization, we calculate a scaling factor for each batch and multiply the expression values to a batch-specific constant. The batch scale factor is the ratio of the median sequencing reads among all batches to the total reads of a batch. This scaling approach overcomes the limitation of reducing read depths in all libraries to the lowest-depth library, which is commonly applied in the global scaling method. We also introduce more cell-to-cell normalization options in the later section.

Filtering and quality control

Quality control (QC) is an important step of scRNA-seq data analysis, as it can be used to reduce poor-quality data that may mask biologically significant variation [7]. Sources of technical noise include low-quality cells that are generally defined as empty droplets, droplets with multiple cells, and dead or dying cells [8]. The quality of cells is described by a series of metrics, such as total number of reads, number of genes expressed by a cell, mean gene expression of a gene or a cell, and proportion of a gene's expression to total expression. Low-quality cells are identified as outliers in terms of library size and gene expression or with expression dominated by controls that are usually defined as mitochondrial and ribosomal genes. As the EMSet automatically recalculates QC metrics when changes are made to the count matrix, the quality of the data set can be monitored in real time with the aid of QC plots (*plotGeneralQC*) Fig. 2. Users can also review the EMSet log for a record of cells and genes removed by filtering methods. Since these QC steps should allow a user to filter cells or genes based on their own defined metrics, *ascend*'s QC functions allow arguments to be passed using additional metadata.

Cell-cell normalization

Cell-cell normalization is another crucial step to remove technical variation between individual cells. The normalization of scRNA-seq data is complicated by the zero-inflated count distributions of genes, which may be due to either biological or technical factors. *ascend* addresses this issue by adapting the normalization by the relative log expression (RLE) method [9] for zero-inflated data by estimating size factors from the geometric means based on true count values that are greater than zero. For each gene, a gene-specific geometric mean is estimated across all cells, not including cells with zero values. The cell-specific size factor is then calculated based on the expression of the gene in a cell relative to the geometric mean of that gene. The size factors for all genes in a cell are used for calculating the

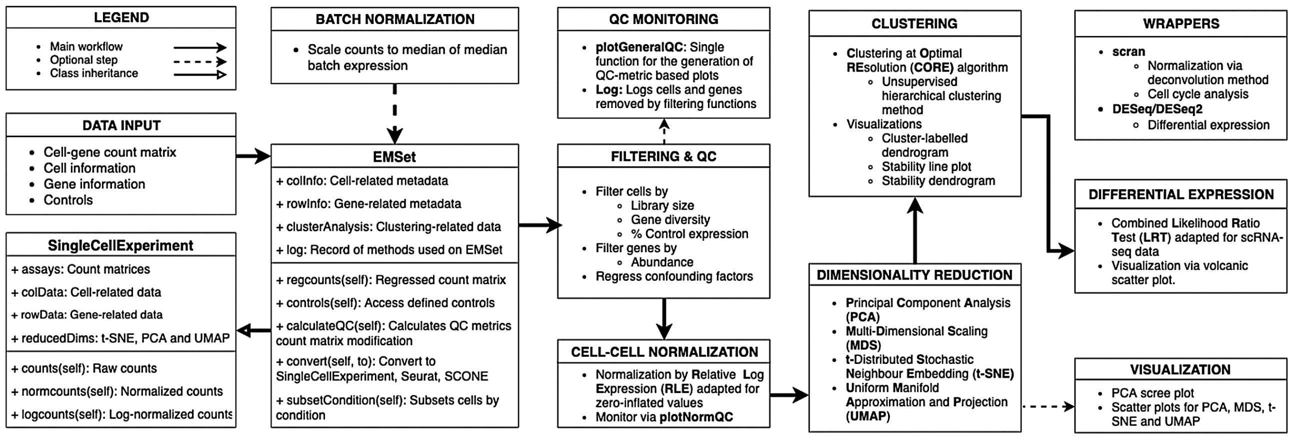
LEGEND
Main workflow
Optional step
Class inheritance
BATCH NORMALIZATION
Scale counts to median of median batch expression
QC MONITORING
plotGeneralQC: Single function for the generation of QC-metric based plots
Log: Logs cells and genes removed by filtering functions
CLUSTERING
Clustering at Optimal REsolution (CORE) algorithm
Unsupervised hierarchical clustering method
Visualizations
Cluster-labelled dendrogram
Stability line plot
Stability dendrogram
WRAPPERS
scran
Normalization via deconvolution method
Cell cycle analysis
DESeq/DESeq2
Differential expression
DATA INPUT
Cell-gene count matrix
Cell information
Gene information
Controls
EMSet
+ colInfo: Cell-related metadata
+ rowInfo: Gene-related metadata
+ clusterAnalysis: Clustering-related data
+ log: Record of methods used on EMSet
+ regcounts(self): Regressed count matrix
+ controls(self): Access defined controls
+ calculateQC(self): Calculates QC metrics count matrix modification
+ convert(self, to): Convert to SingleCellExperiment, Seurat, SCONE
+ subsetCondition(self): Subsets cells by condition
FILTERING & QC
Filter cells by
Library size
Gene diversity
% Control expression
Filter genes by
Abundance
Regress confounding factors
DIFFERENTIAL EXPRESSION
Combined Likelihood Ratio Test (LRT) adapted for scRNA-seq data
Visualization via volcanic scatter plot.
SingleCellExperiment
+ assays: Count matrices
+ colData: Cell-related data
+ rowData: Gene-related data
+ reducedDims: t-SNE, PCA and UMAP
+ counts(self): Raw counts
+ normcounts(self): Normalized counts
+ logcounts(self): Log-normalized counts
DIMENSIONALITY REDUCTION
Principal Component Analysis (PCA)
Multi-Dimensional Scaling (MDS)
t-Distributed Stochastic Neighbour Embedding (t-SNE)
Uniform Manifold Approximation and Projection (UMAP)
CELL-CELL NORMALIZATION
Normalization by Relative Log Expression (RLE) adapted for zero-inflated values
Monitor via plotNormQC
VISUALIZATION
PCA scree plot
Scatter plots for PCA, MDS, t-SNE and UMAP

Figure 1 A summary of the typical analysis workflows and major function groups available in *ascend*.

cell-specific size factor. We introduce the use of a *scran* wrapper function as the default normalization method and recommend this method if the computation time and memory are not limiting factors. Alternatively, the RLE approach introduced here is the faster and more memory-efficient option for cell-to-cell normalization. Users can review the impact of normalization on the counts by generating a series of plots with the *plotNormQC* function Fig. 2, which compares prenormalized and normalized library sizes and individual gene counts.

Reduction of high-dimensional space

Since scRNA-seq data are typically multiple orders of magnitude larger than bulk RNA-seq data (n-cells $\times$ m-genes), dimensionality reduction is vital. Moreover, the expression levels of many genes are likely to be correlated, and therefore the problem of collinearity is common, while additional factors such as dropout rate and high expression variation increase noise in the data [10]. *ascend* contains functions to perform principal component analysis (PCA) to reduce the dimensions of the normalized count data and preserve the data structure (i.e., explain the majority of the variance between cells) Fig. 2. t-Distributed stochastic neighbor embedding and multidimensional scaling are only used to visualize cells in a low-dimensional space, supplemented by information supplied by the user or generated by downstream analysis.

Clustering

Clustering cells into subpopulations or subtypes provides structure to the data set by grouping transcriptionally similar cells. *ascend* implements our previously published Clustering by Optimal REsolution (CORE) method [11], which identifies the most stable clustering identity. First, a Euclidean distance matrix between cells is calculated from the first 20 principal components of the PCA-reduced normalized count matrix. An unsupervised dendrogram is then constructed by applying hierarchical clustering. Outlier cells identified by this initial round of clustering are removed from the data set, although their identifiers are retained in the EMSet logs. The dendrogram is then dynamically reclustered by a top-down split and merging process over multiple iterations with changing tree-height thresholds. This approach merges smaller clusters into larger consensus clusters and uses an adjusted Rand index to compare different clustering results to identify the most stable number of clusters. The method is fast and scalable, enabling the analysis of small clusters at high resolution or larger clusters for more general classification with simpler downstream analysis.

Differential expression

In a heterogeneous data set, such as scRNA-seq data, analyzing the differentially expressed (DE) genes between 1 cluster and the combined remaining clusters can reveal signature genes that can be used to assign identity to a population of cells or to more clearly understand cell transition states. After decomposing the data into subpopulations, *ascend* provides a combined likelihood ratio test (LRT) to compare these subpopulations by finding biological signatures that distinguish them, taking into account subpopulation-size imbalance and high dropout rates. Introduced as a method for single-cell qPCR data [12], the combined LRT has been adapted in *ascend* such that it takes into account genes with zero variance. LRT is particularly suitable for the cases where the number of cells in 2 clusters is very different. In these cases, most dispersion estimation methods, such as those in DESeq, do not result in a convergence. The imbalance issue becomes exaggerated for the cases of smaller clusters, where the high dropout rates have a higher impact. LRT uses a combined distribution assumption consisting of both discrete (on/off) and continuous (low/high expression) components, which helps overcome the issues in dropout and small number of cells. LRT applies chi-square approximation for likelihood differences and thus is fast and less memory intensive. A similar LRT test approach that optimizes a 2-part general linearized model to estimate parameters that account for bimodality and stochastic dropout (cell detection rate) implemented in the MAST package is more computationally intensive, especially for data sets with large cell numbers [13]. The LRT applied in *ascend* does not model cell detection rate to use as a covariate when comparing subpopulations. The resulting implementation is fast, scalable, and robust, even in situations where standard DE methods fail. Wrapper functions are also provided for DE analysis based on negative binomial tests from DESeq [9]. We introduced several modifications that allow (i) more accurate estimation of fold change (adjusted fold change) and (ii) more efficient multiprocessing, using a divide-and-conquer approach, to handle large data sets and substantially reduce computational time.

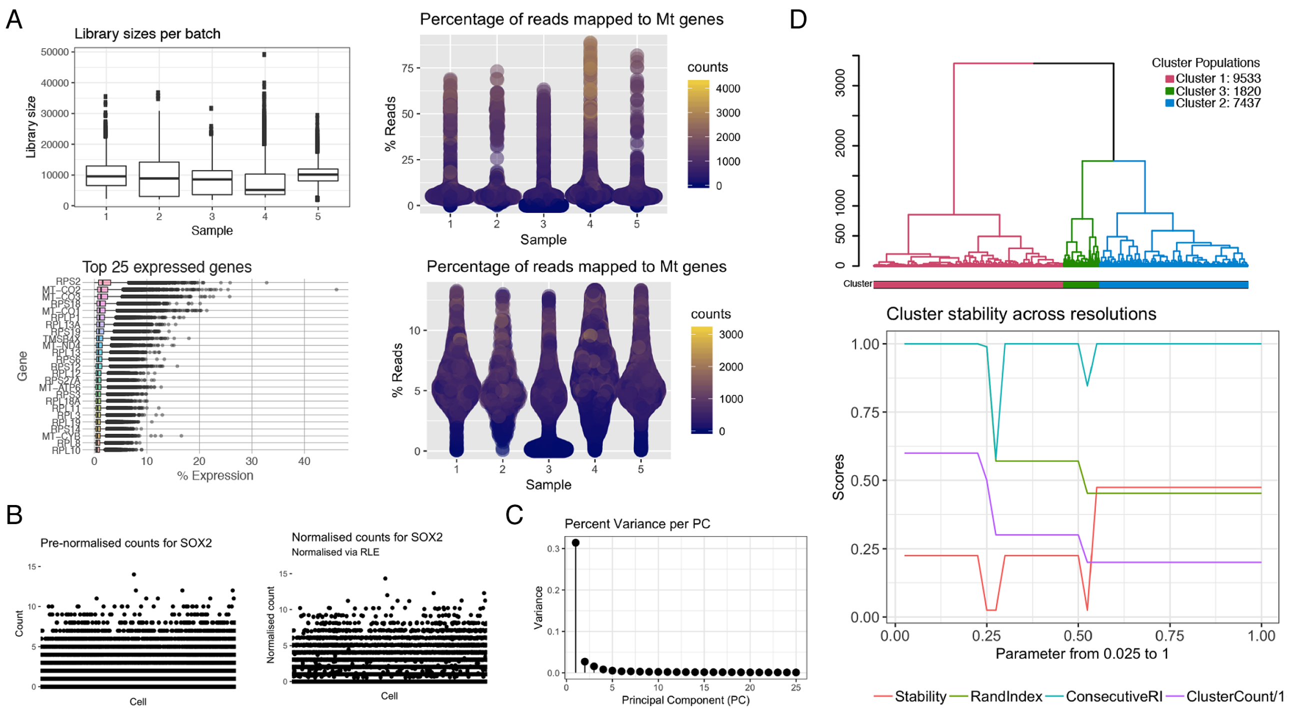

Figure 2 Graphics generated by *ascend* during different stages of analysis. (A) Quality control plots include a boxplot representing distribution of library sizes across each batch, a boxplot representing the expression of the top 25 most abundant transcripts, and violin plots representing proportion of mitochondrial-related transcripts to total expression per sample. (B) Normalization quality control plot represents the expression of the *SOX2* gene before and after RLE normalization. (C) Scree plot related to principal component dimensionality reduction. (D) Clustering plots include a cluster-labeled dendrogram and a line plot depicting the relationships between cluster numbers and stability.

Benchmarking

The CPU time of the *ascend* package was compared to 2 other toolkits developed for scRNA-seq analysis: *Seurat* [14] and *scater* [4]. Using a data set that comprises 1,272 retinal ganglion cells from the study by Daniszewski et al. [15], these packages were used to perform quality control, normalization, dimensionality reduction, clustering, and differential expression using equivalent methods. As shown in Supplementary File 1, *ascend*'s processing time is comparable to *Seurat* [14] and *scater* [4].

Conclusion

In summary, *ascend* is a user-friendly and computationally efficient package for analyzing scRNA-seq data from all experimental platforms. *ascend* implements quality control and filtering approaches that are highly customizable, as well as a unsupervised clustering method (CORE), and optimizes speed for implementing established analysis techniques for normalization and differential gene expression. Statistical methods for single-cell analysis are constantly evolving. Here we have implemented a series of current cutting-edge approaches, although the flexibility of *ascend* allows it to adapt as future methods are developed. The *ascend* package and context-specific tutorials addressing a range of analytical scenarios are available at https://github.com/powellgenomicslab/ascend. We expect that *ascend* is especially useful for biologists who wish to explore their own data sets using expert domain knowledge and an easy-to-use and complete toolkit.

Methods

Data

Here we present an application case study of *ascend* using scRNA-seq data from undifferentiated human induced pluripotent stem cells generated as described by Nguyen et al. [11]. The raw 10× Chromium Single Cell 3′ Gene Expression data set consists of 20,448 cells that are divided into 5 samples. Raw (FASTQ or aggregated count matrix) and processed data can be downloaded from ArrayExpress (accession number: E-MTAB-6687).

Preprocessing of scRNA-seq data set

The raw expression data from each sample were combined into a single data set using Chromium's Cell Ranger 1.2.0 *aggr* function. This function performs 2 tasks: batch normalization and transcript count aggregation. Cell Ranger first normalizes the sequencing depth between the 5 samples by subsampling reads for each sample until their median depth equals the sample with the shallowest read depth. Once normalized, the transcript counts from each sample are combined into a single matrix. The rows of this matrix were labeled with ENSEMBL gene identifiers; to simplify analysis, these were replaced with corresponding gene names that were stored in the Cell Ranger outputs.

ascend analysis

ascend was run in RStudio (R Version 3.5.0), and analysis of data from quality control to differential expression (LRT) took 95 minutes on a MacBook Pro laptop with a dual-core Intel Core i5 2.7 GHz and 8 GB of RAM. To minimize memory use, the raw expression matrix was converted into a sparse matrix using the Matrix R Package [16]. The sparse matrix and accompanying cell-related metadata were loaded into an *EMSet*.

```
library(ascend)
library(Matrix)

# Read in file downloaded from ArrayExpress
counts <- read.csv(''raw_expMat_cellbarcodes.tsv'',
sep = ''\t'')

# Convert counts to a sparseMatrix
counts <- as(as.matrix(counts), ''dgCMatrix'')

# Create colInfo dataframe
batches <- sapply(strsplit(colnames(counts),
''[.]''), function(x) x[2])

colInfo <- S4Vectors::DataFrame(
cell_barcode = colnames(counts),
batch = batches)

# Define controls
mt_genes <- grep(''M̂t−″,
rownames(counts),
ignore.case = TRUE,
value = TRUE)

rb_genes <- grep(''R̂ps|R̂pl″,
rownames(counts),
ignore.case = TRUE,
value = TRUE)

controls <- list(Mt = mt_genes,
 Rb = rb_genes)

# Create a new EMSet
EMSet <- EMSet(counts,
colInfo = colInfo,
controls = controls)
```

The quality of the expression data was assessed with the aid of QC figures generated by the *plotGeneral*QC function from the *ascend* package.

```
raw_qc_plots <- plotGeneralQC(EMSet)
```

The data then underwent quality control. First, cells were filtered based on library size, number of detected genes, and reads mapped to mitochondrial and ribosomal genes using the default threshold of 3 × Median Absoulte Deviation (MAD) range. Next, cells were removed if 20% of reads were mapped to mitochondrial transcripts and 50% of reads were mapped to ribosomal transcripts. Finally, genes were removed if they were expressed in less than 0.1% of the cell population.

```
# Remove cells that are outliers
EMSet <- filterByOutliers(EMSet,
cell.threshold = 3,
control.threshold = 3)

# Remove cells where mitochondrial-related
# transcripts account for at least 20% of reads
EMSet <- filterByControl(EMSet,
control = ''Mt'',
pct.threshold = 20)

# Remove cells where ribosomal-related
# transcripts account for at least 50% of reads
EMSet <- filterByControl(EMSet,
control = ''Rb'',
pct.threshold = 50)
# Remove genes that are expressed in less than
# 0.1% of the cell population
EMSet <- filterLowAbundanceGenes(EMSet,
pct.threshold = 0.1)
```

QC removed 1,683 cells and 16,272 genes, leaving 18,765 cells and 16,466 genes for further analysis. The UMI counts for the remaining cells and genes were normalized with the *normaliseByRLE* function. The effectiveness of the normalization method was assessed with the aid of figures generated by the *plotNormQC* function. Mitochondrial and ribosomal gene transcripts were removed from the data before proceeding with further analysis.

```
# Normalize dataset using RLE
EMSet <- normaliseByRLE(EMSet)

# Plot normalisation quality control plots
norm_qc <- plotNormQC(EMSet,
gene_list = c(''GAPDH'', ''MALAT1''))

# Remove controls from dataset
EMSet <- excludeControl(EMSet,
control = c(''Mt'', ''Rb''))
```

To reduce the dimensions of the data, the normalized UMI count matrix was reduced using the *ascend* function *runPCA*. This function is a wrapper for the *prcomp* function from the *irlba* R package.

```
   # Reduce dataset with PCA
   EMSet <- runPCA(EMSet, ngenes =
1500, scaling = TRUE)
```

The scree plot generated by *ascend*'s *plotPCAVariance* function revealed the first 10 principal components explained 88.70% of the variance in these data. These 10 principal components were passed to the CORE algorithm function to build a cell distance matrix and subsequently a dendrogram that was used to identify clusters.

```
EMSet <- runCORE(EMSet,
conservative = FALSE,
remove.outliers = TRUE,
nres = 40,
dims = 10)
```

Using the default arguments, the CORE method generated clustering results for 40 different resolutions, and based on the Rand index, the function identified 3 clusters of cells that represent the most stable result. Clusters 1, 2, and 3 comprised 9,046, 6,436, and 3,283 cells, respectively.

To characterize the biological properties of the 3 clusters, differential expression was performed using *ascend*'s *runDiffExpression* function. The expression of each cluster was compared to the expression of the other clusters.

```
# Comparison of cluster 1 vs other clusters
cluster1_vs_all <- runDiffExpression(EMSet,
 group = ''cluster'',
 condition.a = 1,
 condition.b = c(2, 3))

# Comparison of cluster 2 vs other clusters
cluster2_vs_all <- runDiffExpression(EMSet,
 group = ''cluster'',
 condition.a = 2,
 condition.b = c(1, 3))
```

```
# Comparison of cluster 3 vs other clusters
cluster3_vs_all <- runDiffExpression(EMSet,
 group = ''cluster'',
 condition.a = 3,
 condition.b = c(1, 2))
```

Using a Bonferroni-corrected P-value threshold ($P < 3.1 \times 10^{-7}$) and an absolute $\log_2$ fold change greater than 2, differential expression analysis revealed clusters 1, 2, and 3 had 230, 7, and 47 DE genes, respectively.

Availability of source code and requirements

- Project name: *ascend*
- Project home page: https://github.com/powellgenomicslab/ascend
- Operating system(s): Platform independent
- Programming language: R
- Other requirements: R 3.5, Bioconductor 3.7
- License: GPL 3.0
- RRID: SCR_017257

Availability of supporting data and materials

The data supporting the results of this article are available in ArrayExpress at accession E-MTAB-6687 [17]. An archival copy of the code and supporting data is also available via the *GigaScience* repository, GigaDB [18].

Declarations

None declared.

List of abbreviations

CORE: Clustering at Optimal REsolution; DE: differentially expressed; LRT: likelihood ratio test; PCA: principal component analysis; QC: quality control; RLE: relative log expression; scRNA-seq: single-cell RNA-sequencing; UMI: unique molecular identifier.

Ethical approval

Not applicable.

Consent for publication

Not applicable.

Competing interests

The authors declare that they have no competing interests.

Funding

This work was supported by the National Health and Medical Research Council grants 1107599 and 1083405.

Authors' contributions

A.S. wrote the software; all authors contributed to software development; A.S., S.W.L., Q.H.N., and J.E.P. wrote the manuscript. Q.H.N. and J.E.P. oversaw the project.

References

1. Zheng GXY, Terry JM, Belgrader P, et al. Massively parallel digital transcriptional profiling of single cells. Nat Commun 2017;**8**(206):14049.
2. Macosko EZ, Basu A, Satija R, et al. Highly parallel genome-wide expression profiling of individual cells using nanoliter droplets. Cell 2015;**161**(5):1202–14.
3. Butler A, Hoffman P, Smibert P, et al. Integrating single-cell transcriptomic data across different conditions, technologies, and species analysis. Nat Biotechnol 2018;**36**(5):411–20.
4. McCarthy DJ, Campbell KR, Lun ATL, et al. Scater: pre-processing, quality control, normalization and visualization of single-cell RNA-seq data in R. Bioinformatics 2017;**33**(8):1179–86.
5. Lun ATL, Bach K, Marioni JC. Pooling across cells to normalize single-cell RNA sequencing data with many zero counts. Genome Biol 2016;**17**:75.
6. Lun A, Risso D. SingleCellExperiment: S4 classes for single cell data. 2018, R package version 1.3.6. https://doi.org/10.18129/B9.bioc.SingleCellExperiment
7. Bacher R, Kendziorski C, Auer P, et al. Design and computational analysis of single-cell RNA-sequencing experiments. Genome Biol 2016;**17**(1):63.
8. Ilicic T, Kim JK, Kolodziejczyk AA, et al. Classification of low quality cells from single-cell RNA-seq data. Genome Biol 2016;**17**(1):29.
9. Anders S, Huber W. Differential expression analysis for sequence count data. Genome Biol 2010;**11**(10): R106.
10. Hicks SC, Townes FW, Teng M, et al. Missing data and technical variability in single-cell RNA-sequencing experiments. Biostatistics 2018;19(4):562–78.
11. Nguyen Q, Lukowski S, Chiu H, et al. Single-cell RNA-seq of human induced pluripotent stem cells reveals cellular heterogeneity and cell state transitions between subpopulations. Genome Res 2018;**28**(7):1053–66.
12. McDavid A, Finak G, Chattopadyay PK, et al. Data exploration, quality control and testing in single-cell qPCR-based gene expression experiments. Bioinformatics 2013;**29**(4):461–7.
13. Finak G, McDavid A, Yajima M, et al. MAST: a flexible statistical framework for assessing transcriptional changes and characterizing heterogeneity in single-cell RNA sequencing data. Genome Biol 2015;**16**:278.
14. Satija R, Farrell JA, Gennert D, et al. Spatial reconstruction of single-cell gene expression data. Nat Biotechnol 2015;**33**(5):495–502.
15. Daniszewski M, Senabouth A, Nguyen Q, et al. Single cell RNA sequencing of stem cell-derived retinal ganglion cells. bioRxiv 2017; http://biorxiv.org/content/early/2017/09/22/191395.
16. Bates D, Maechler M. Matrix: sparse and dense matrix classes and methods. 2018. https://CRAN.R-project.org/package=Matrix.
17. Supporting ArrayExpress scRNA-seq data https://www.ebi.ac.uk/arrayexpress/experiments/E-MTAB6687/.
18. 5525/100615, Senabouth A, Lukowski S, et al. ascend: R package for analysis of single cell RNA-seq data. GigaScience Database 2019. http://dx.doi.org/10.5524/100615.