Using Single-Cell Transcriptome Sequencing to Infer Olfactory Stem Cell Fate Trajectories

Sandrine Dudoit
Division of Biostatistics and Department of Statistics
University of California, Berkeley
www.stat.berkeley.edu/~sandrine

Intelligent Systems for Molecular Biology (ISMB)
Orlando, FL
July 11, 2016

Version: 10/07/2016, 12:53

Acknowledgments

- Sandrine Dudoit, Division of Biostatistics and Department of Statistics, UC Berkeley.
  - Fanny Perraudeau, Graduate Group in Biostatistics, UC Berkeley.
  - Davide Risso, Division of Biostatistics, UC Berkeley.
  - Kelly Street, Graduate Group in Biostatistics, UC Berkeley.
- John Ngai, Department of Molecular and Cell Biology, UC Berkeley – Principal investigator.
  - Diya Das.
  - Russell Fletcher.
  - David Stafford.
- Elizabeth Purdom, Department of Statistics, UC Berkeley.
- Jean-Philippe Vert, Mines ParisTech and Institut Curie, Paris, France.
  - Svetlana Gribkova.

Acknowledgments

- Nir Yosef, Department of Electrical Engineering and Computer Sciences, UC Berkeley.
  - Michael Cole.
  - Allon Wagner.
- Funded by BRAIN Initiative and California Institute for Regenerative Medicine (CIRM).

Outline

Outline

Methods
Software: slingshot

R. Fletcher, J. Ngai

Olfactory Stem Cells and Neural Regeneration

- The nature of stem cells giving rise to the nervous system is of particular interest in neurobiology, because neural stem cells remain active in certain brain regions for the entire life of an individual.
- We focus on the mouse olfactory epithelium (OE), a site of active neurogenesis in the postnatal animal.
- Adult olfactory stem cells support the replacement of olfactory sensory neurons and non-neuronal support cells (e.g., sustentacular) over postnatal life and can reconstitute the entire OE following injury.
- The OE is a convenient system to study, due to its experimental accessibility (in situ analysis) and its limited number of cell types:
  - olfactory sensory neurons (OSN),
  - sustentacular cells (SUS),

- cells of the Bowman gland,
- microvillous cells (rare).

Olfactory Stem Cells and Neural Regeneration

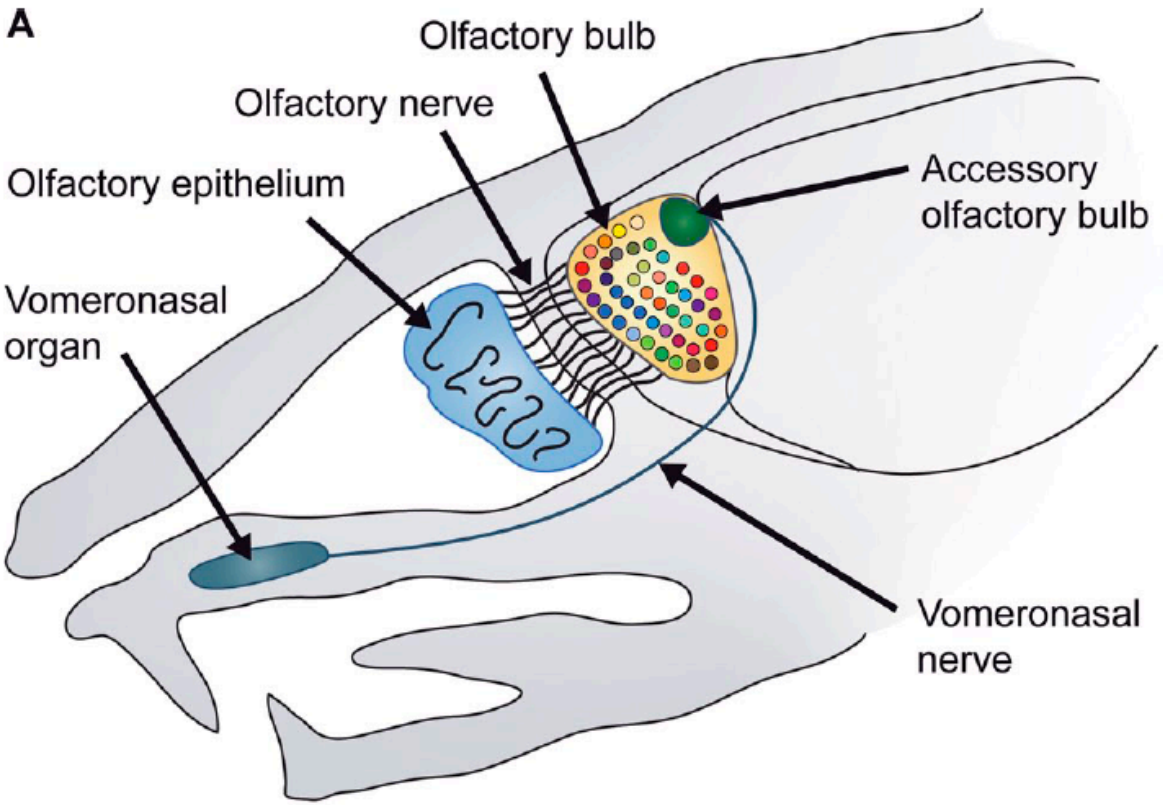
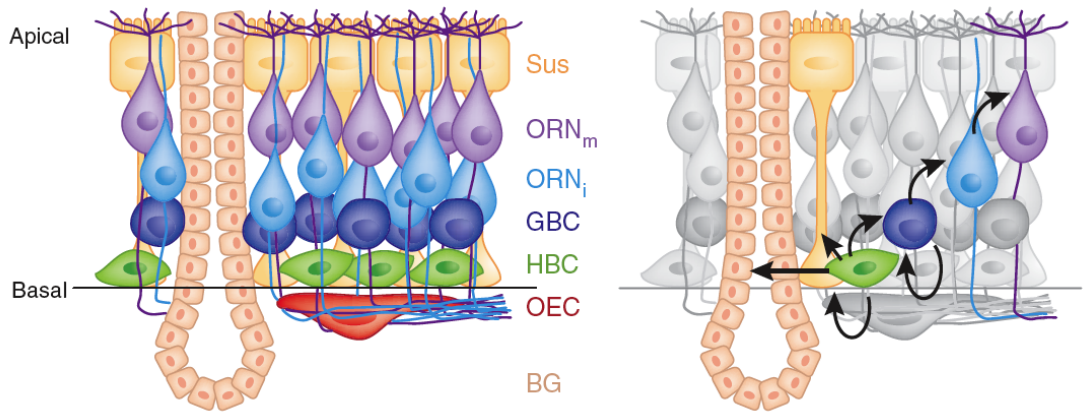


Figure 1: *Mouse olfactory epithelium.*

The Horizontal Basal Cell Is an Adult Tissue Stem Cell



- HBCs: multipotent, quiescent – deep reserve adult tissue stem cell
- GBCs: proliferative progenitor cells + transit-amplifying cells

Figure 2: *Olfactory epithelium cell types.*

Open questions.

- Determine the stage at which the neuronal and non-neuronal lineages bifurcate/diverge.
- Characterize discrete intermediate stages of cell differentiation.
- Identify the genetic networks and signaling pathways that promote self-renewal and regulate the transition to differentiation.

Olfactory Stem Cells and Neural Regeneration

p63 regulation of horizontal basal cells.

- The horizontal basal cell (HBC) is an adult tissue stem cell.
- The p63 protein (tumor protein p63, TP63) promotes self-renewal of HBC by blocking differentiation.
- When p63 is down-regulated, this "brake" is removed, allowing differentiation to proceed at the expense of self-renewal. Thus, p63 can be viewed as a "molecular switch" that decides between the alternate stem cell fates of self-renewal vs. differentiation.
- We use single-cell RNA-Seq to analyze cell fate trajectories from olfactory stem cells (HBC) of p63 conditional knock-out mice.

Olfactory Epithelium p63 Dataset

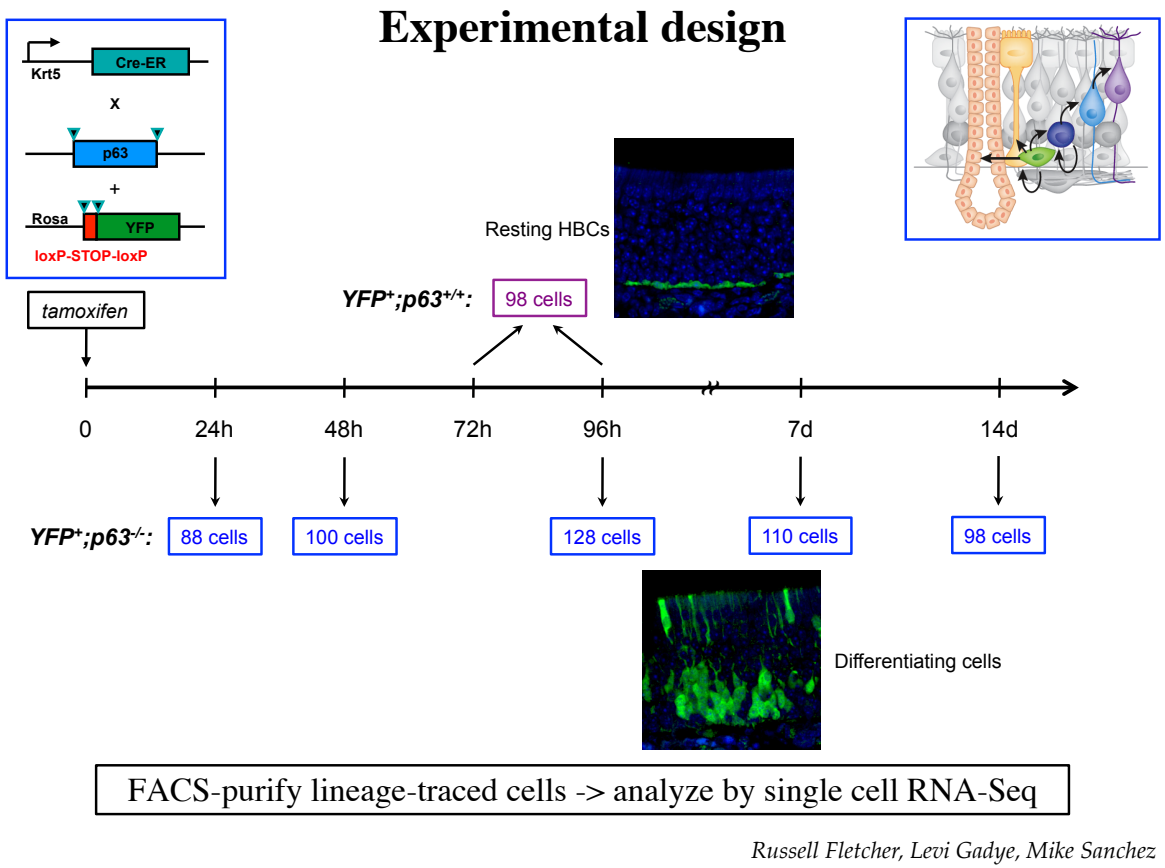


Figure 3: *Experimental design.* Single-cell RNA-Seq for 636 HBC: 102 wild-type and 534 p63 knock-out cells.

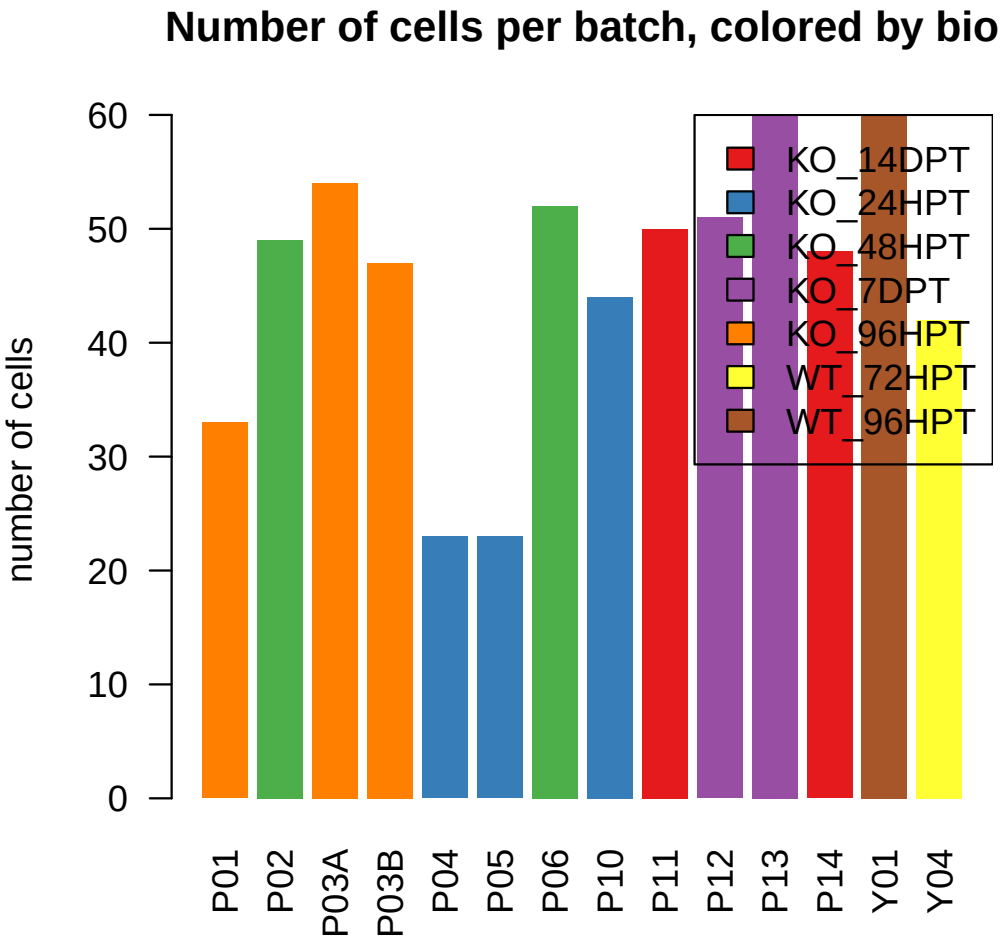


Figure 4: *Experimental design.* Number of cells per batch, colored by biological condition.

Olfactory Epithelium p63 Dataset

- Single-cell RNA-Seq for 636 HBC.
  - 102 wild-type (WT)/resting cells
    534 p63 knock-out (KO) cells, at five timepoints following tamoxifen treatment.
  - Biological replicate: Cells from 1–3 mice.
    At least two replicates per biological condition.
  - One FACS run and one C1 run per biological replicate
    $\Longrightarrow$ 14 batches.
  - 8 HiSeq runs (96 cells/lane, single-end 50-base-pair reads).
- Some confounding between biological and technical effects.
- Batch effects nested within biological effects.

Olfactory Epithelium p63 Dataset

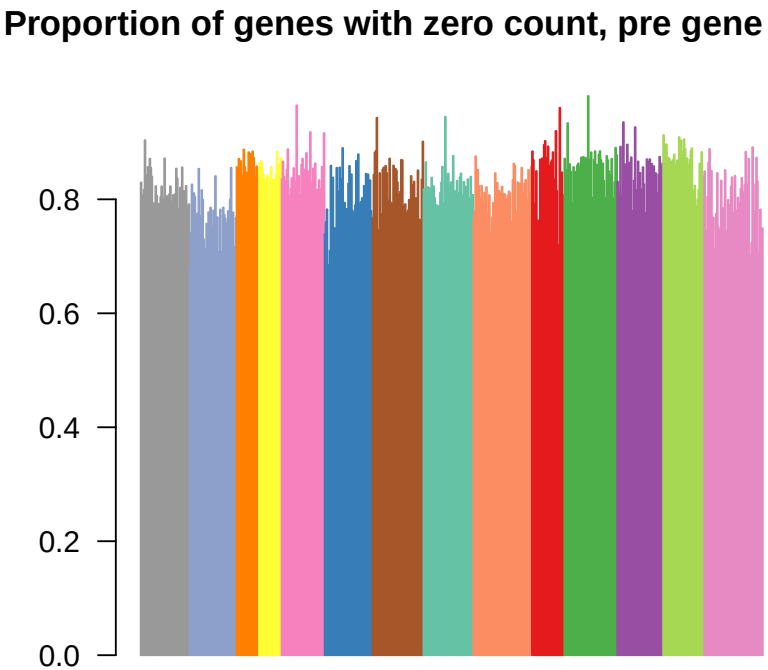
- Marker genes. 94 marker genes, curated from literature and from prior microarray, sequencing, and in situ experiments, e.g., neuronal, progenitor cell markers.
- Housekeeping genes. 715 housekeeping (HK) genes, curated from prior microarray experiments, expected to be constantly and highly-expressed across cells of the OE.
- Gene-level read counts. TopHat2 alignment to RefSeq mm10 genome and `featureCounts` (`bioinf.wehi.edu.au/featureCounts`) counting, with genes defined as union of all isoforms.

- Input/Output (I/O).
  - Input. FASTQ files, gene-level annotation metadata, sample-level annotation metadata.
  - Output. Cluster labels corresponding to cellular types, cell lineages and pseudotimes, list of differentially expressed genes corresponding to markers/transcriptome signatures for these types/lineages.
- Pre-processing. Read alignment, gene-level expression quantitation, quality assessment/control.
- Exploratory data analysis (EDA) and quality assessment/control (QA/QC). Numerical and graphical summaries of gene-level read counts, sample filtering (e.g., based on QC measures), gene filtering (e.g., zero inflation).

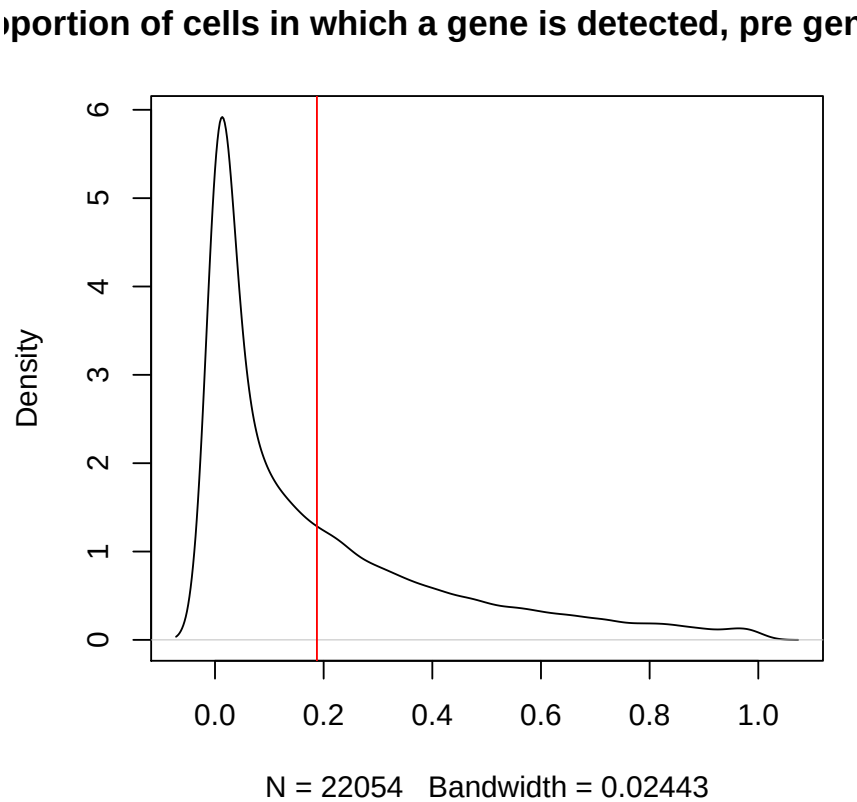
- Normalization and expression quantitation. Adjust for unwanted technical effects (e.g., C1 run effects), account for zero inflation and over-dispersion.
- Cluster analysis. Dimensionality reduction (e.g., based on principal component analysis), distance measure, clustering procedure and tuning parameters, confidence measures/validation of clustering results.
- Cell lineage and pseudotime inference.
- Differential expression (DE) analysis. Identification of marker genes for cell types/lineages.

Computationally Reproducible Research

- Version control system for software development and data analysis: `Git`, `GitHub`.
- R data packages, with standard object structure (S4 classes *ExpressionSet*, *SummarizedExperiment*) for gene-level expression measures and gene-level and sample-level annotation metadata.
- Compendia, i.e., integrated and executable input documents, with text, data, code, and software, that allow the generation of dynamic and reproducible output documents, intermixing text and code output (textual and graphical). Different views of the output document (e.g., PDF, HTML) can be automatically generated and updated whenever the text, data, or code change.
  E.g. For R and LaTeX/markdown, Sweave, knitr, RStudio.



(a) Proportion of genes with zero count



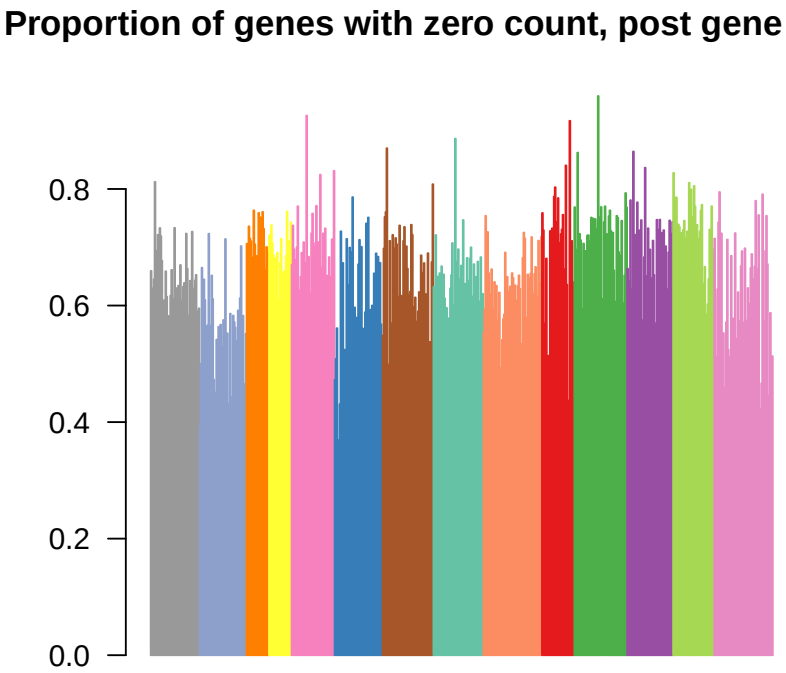
(b) Proportion of cells in which a gene is detected

Figure 5: *Zero inflation.* Pre gene filtering.

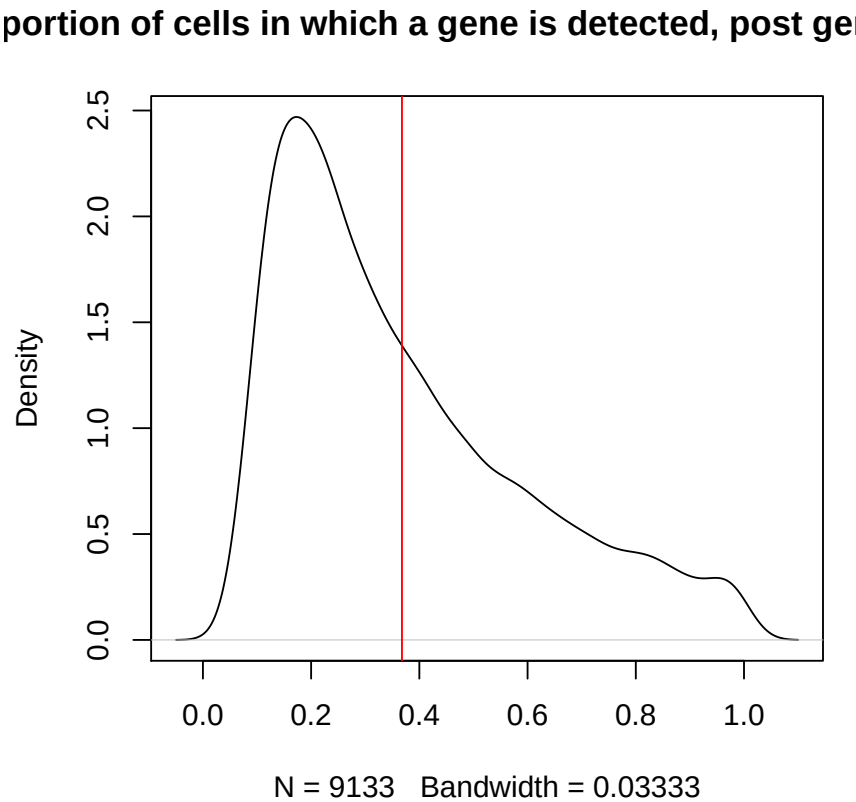
Zero Inflation

- Single-cell RNA-Seq data have many more genes with zero read counts than bulk RNA-Seq data.
- This zero inflation could occur for biological reasons (i.e., the gene is simply not expressed) or technical reasons (e.g., low capture efficiency).
- Zero-count gene filtering is advisable for normalization and downstream analyses.
- Most RNA-Seq normalization methods involve scaling and perform poorly when many genes have zero counts.
- In particular, the global-scaling method of Anders and Huber (2010), implemented in the Bioconductor R package DESeq, discards any gene having zero count in at least one sample. In practice, the scaling factors are therefore estimated based on only a handful of genes, e.g., 5/22,054 genes for OE dataset.

- Full-quantile (FQ) normalization also doesn't behave properly due to ties from the large number of zeros.
- We apply the following zero-count gene filtering to the OE dataset: Retain only the genes with at least $n_r = 20$ reads, in at least $n_s = 40$ samples.
  This yields 9,133/22,054 genes.



(a) Proportion of genes with zero count



(b) Proportion of cells in which a gene is detected

Figure 6: *Zero inflation.* Post gene filtering.

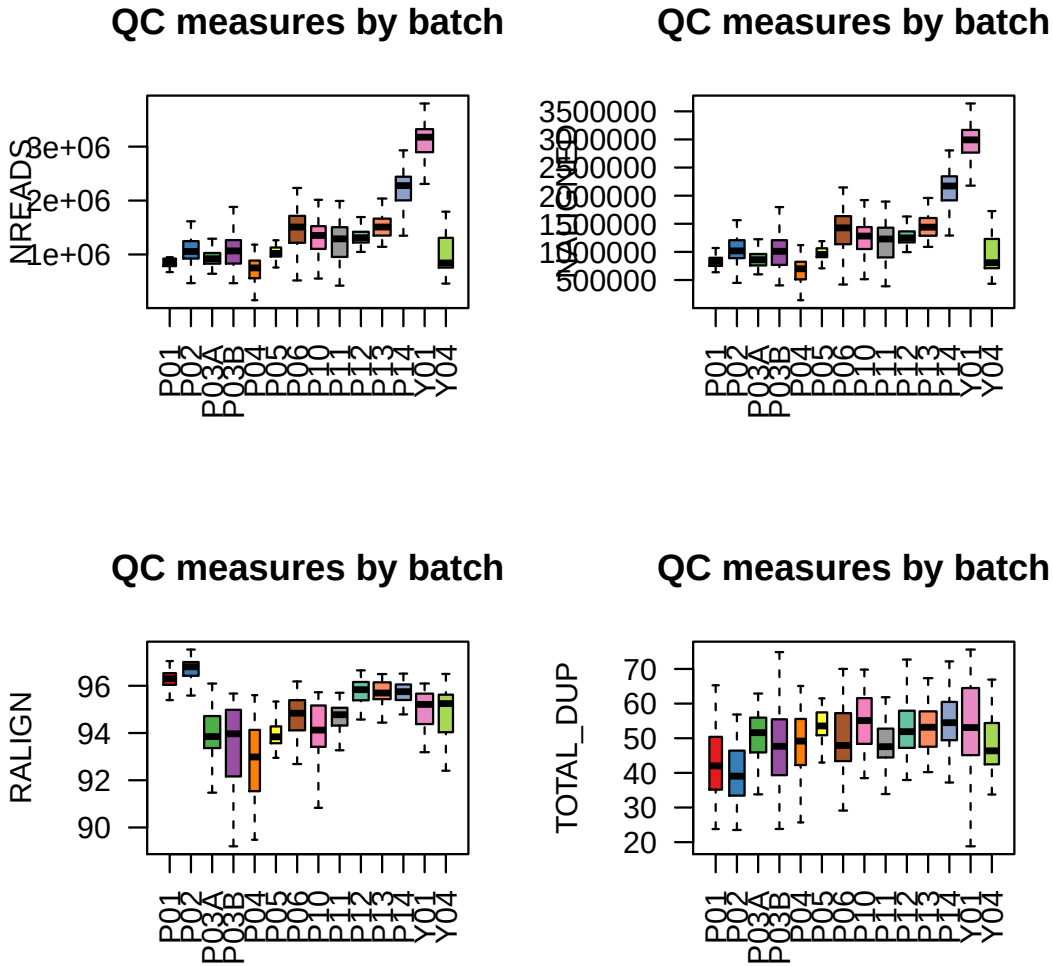


Figure 7: *Sample-level QC.* Boxplots of QC measures, by batch.

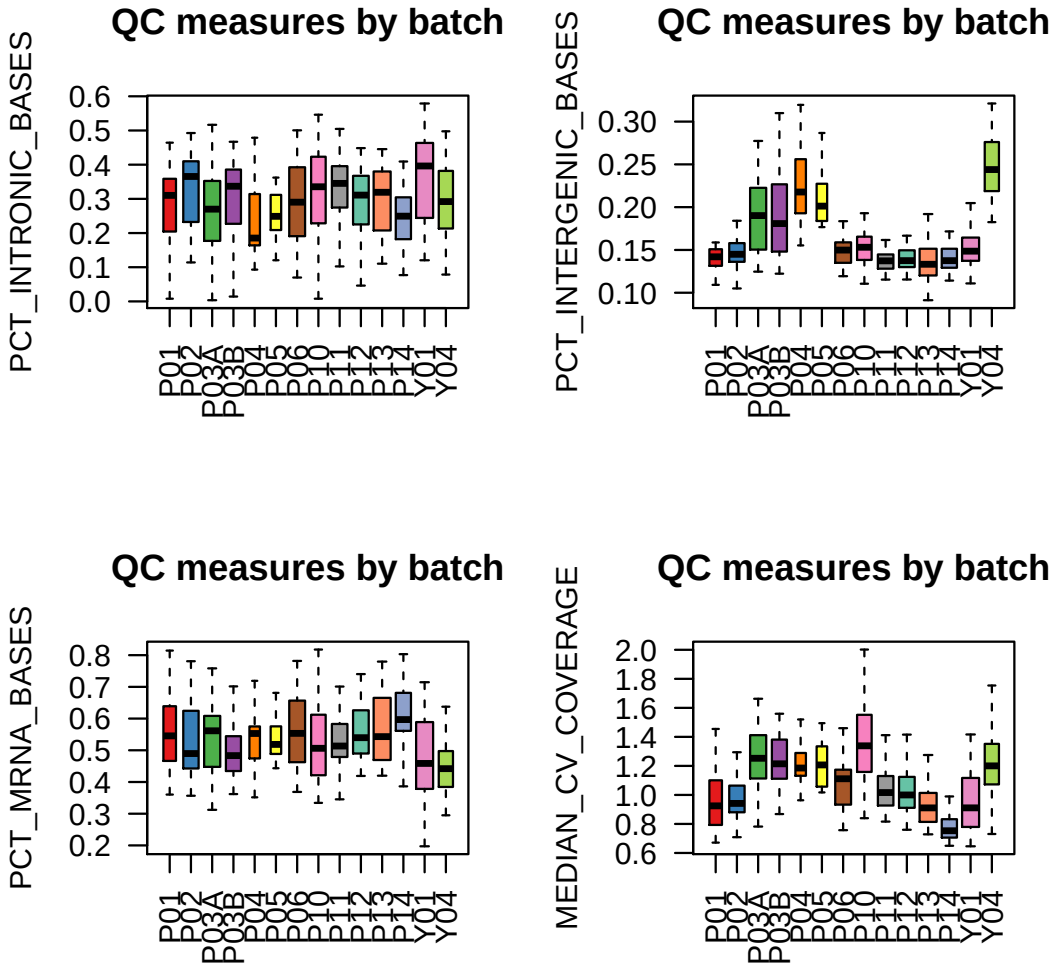
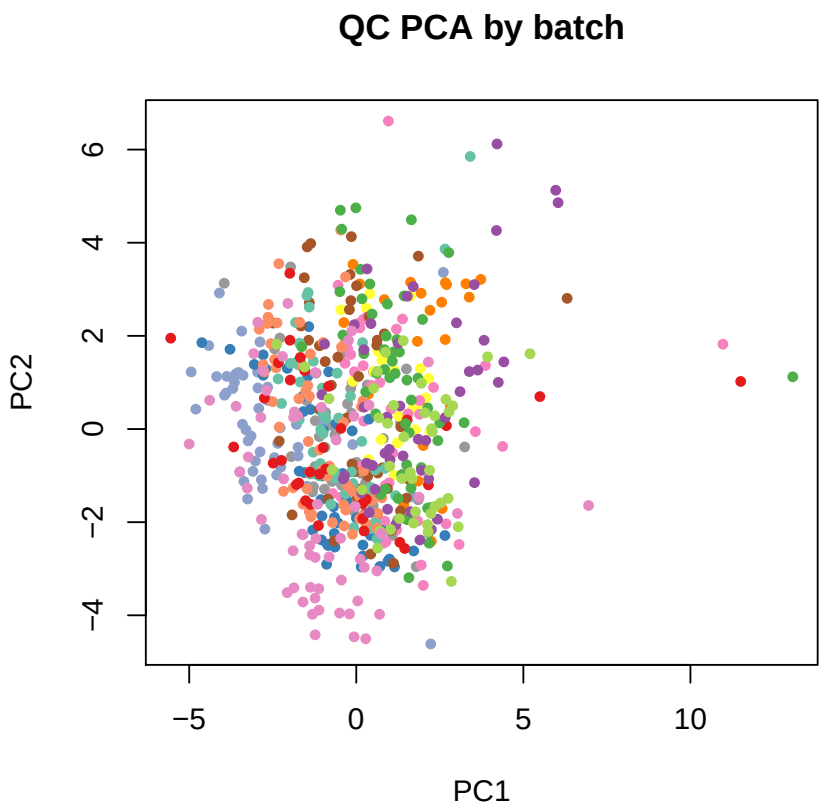
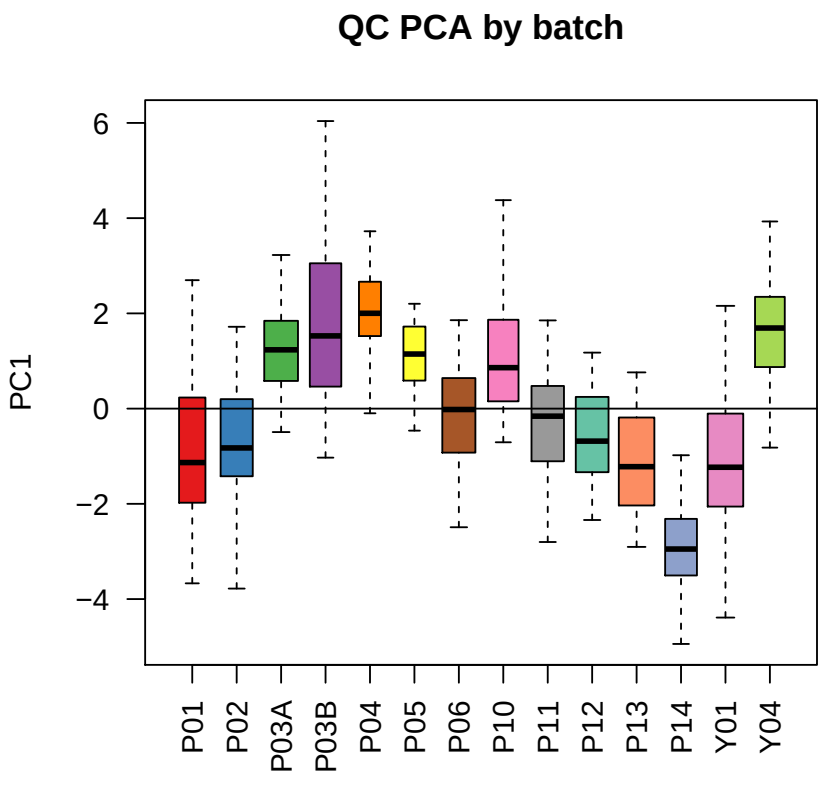


Figure 8: *Sample-level QC.* Boxplots of QC measures, by batch.

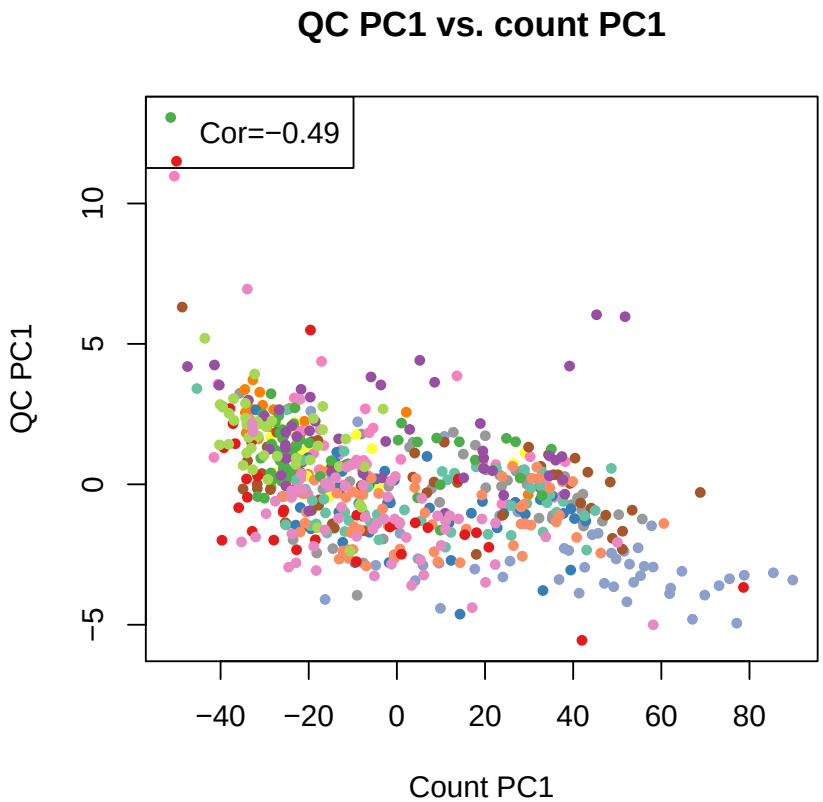


(a) QC PC2 vs. PC1, colored by batch

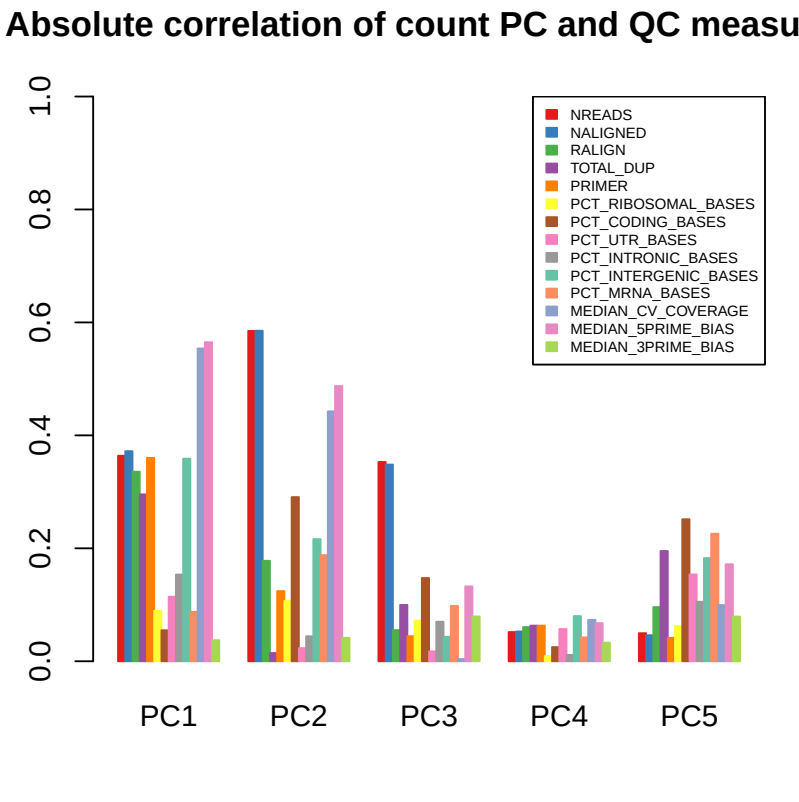


(b) QC PC1, by batch

Figure 9: *Sample-level QC.* Principal component analysis (PCA) of sample-level QC measures.



(a) QC PC1 vs. count PC1



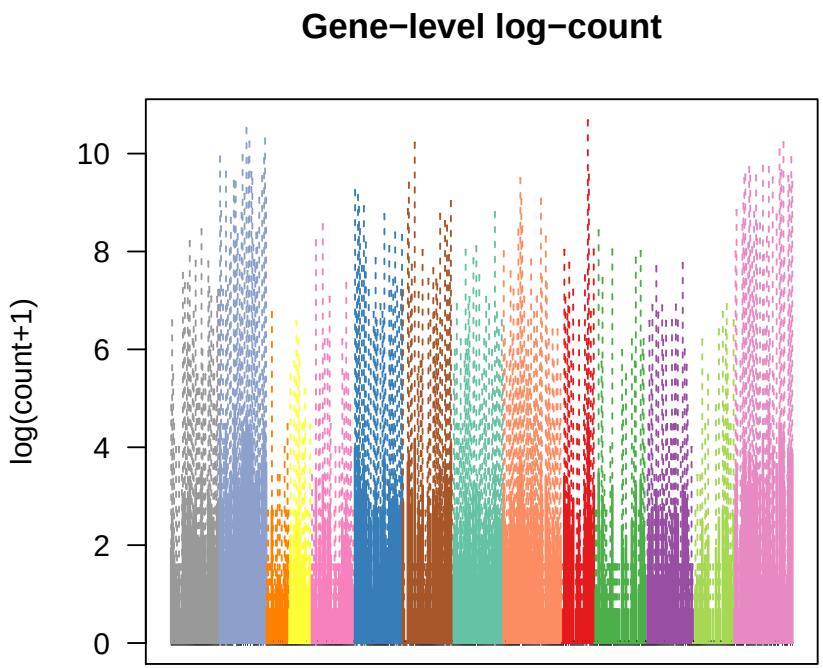
(b) Correlation of count PC and QC measures

Figure 10: *Sample-level QC.* Association of counts and sample-level QC measures.

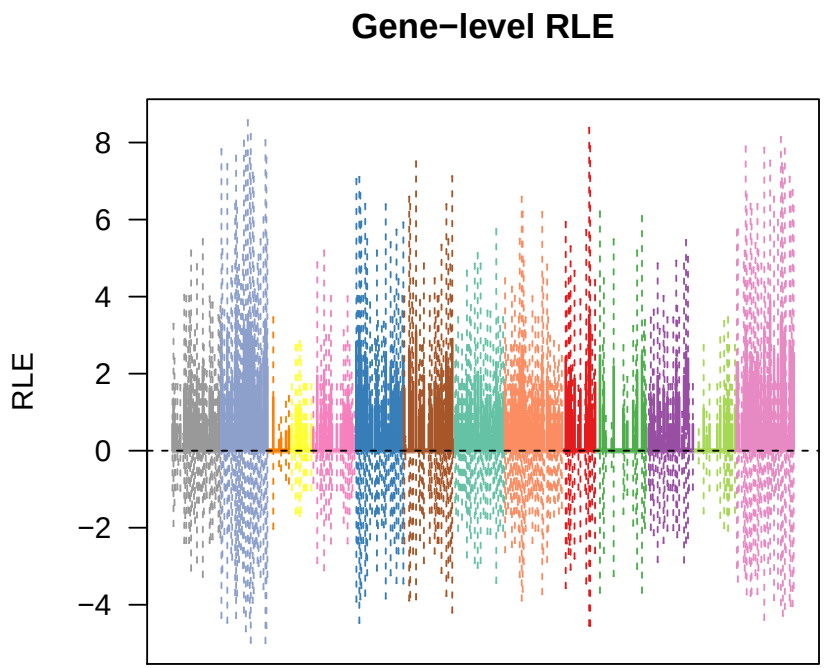
Sample-Level QC: Summary

- The distribution of QC measures can vary substantially within and between batches.
- Some QC measures clearly point to low-quality samples, e.g., low percentage of mapped reads (RALIGN).
- There can be a strong association between QC measures and read counts (cf. PCA).
- Filtering samples based on QC measures is advisable, as normalization procedures may not be able to adjust for QC and some samples simply have low quality.
- Normalization procedures based on QC measures (e.g., regression on first few PC of QC measures) should also be considered.

Gene-Level Counts



(a) Log-count



(b) RLE

Figure 11: *Gene-level counts.* Gene-level log-count and relative log expression (RLE = log-ratio of read count to median read count across samples).

Gene-Level Counts: Summary

- After gene and sample filtering and before normalization, there are large differences in gene-level count distributions within and between batches (cf. RLE, housekeeping genes).
- The counts are still zero-inflated.
- There can be substantial association of counts and sample-level QC measures.
- Normalization is essential before any clustering or differential expression analysis, to ensure that observed differences in expression measures between samples and/or genes are truly due to differential expression and not technical artifacts.

D. Risso, M. Cole, N. Yosef

SCONE: Single-Cell Overview of Normalized Expression. A general framework for the normalization of scRNA-Seq data.

- Range of normalization methods.
  - Global-scaling, e.g., DESeq, TMM.
  - Full-quantile (FQ).
  - Unknown factors of unwanted variation: Remove unwanted variation (RUV).
  - Known factors of unwanted variation: Regression-based normalization on, e.g., QC PC, C1 run.
- Normalization performance metrics.
- Numerical and graphical summaries of normalized read counts and metrics.
- R package `scone`, to be released through the Bioconductor Project: `github.com/yoseflab/scone`.

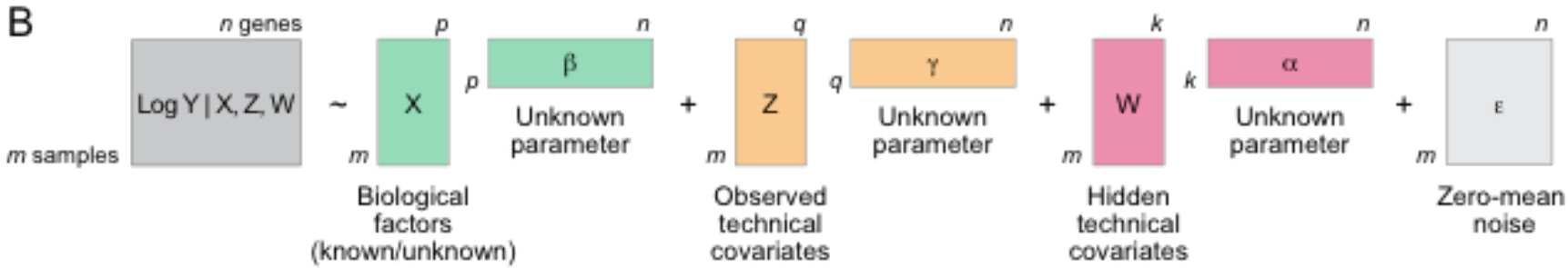


Figure 12: *scone*. Regression model.

Performance metrics. (Green: Good when high; Red: Good when low.)

- BIO_SIL: Average silhouette width by biological condition.
- BATCH_SIL: Average silhouette width by batch.
- PAM_SIL: Maximum average silhouette width for PAM clusterings, for a range of user-supplied numbers of clusters.
- EXP_QC_COR: Maximum squared Spearman correlation between count PCs and QC measures.
- EXP_UV_COR: Maximum squared Spearman correlation between count PCs and factors of unwanted variation (preferably derived from other set of negative control genes than used in RUV).
- EXP_WV_COR: Maximum squared Spearman correlation between count PCs and factors of wanted variation (derived from positive control genes).

- RLE_MED: Mean squared median relative log expression (RLE).
- RLE_IQR: Mean inter-quartile range (IQR) of RLE.

Application to OE p63 dataset.

- Apply and evaluate 172 normalization procedures using main `scone` function.
  - `scaling_method`: None, DESeq, TMM, FQ.
  - `uv_factors`: None, RUVg $k = 1, \cdots, 5$, QC PC $k = 1, \cdots, 5$.
  - `adjust_biology`: Yes/no.
  - `adjust_batch`: Yes/no.
- Select a normalization procedure based on (function of) the performance scores.
  Unweighted mean score $\Longrightarrow$ `none,fq,qc_k=4,bio,no_batch`
  Weighted mean score $\Longrightarrow$
  `none,fq,qc_k=2,no_bio,no_batch`

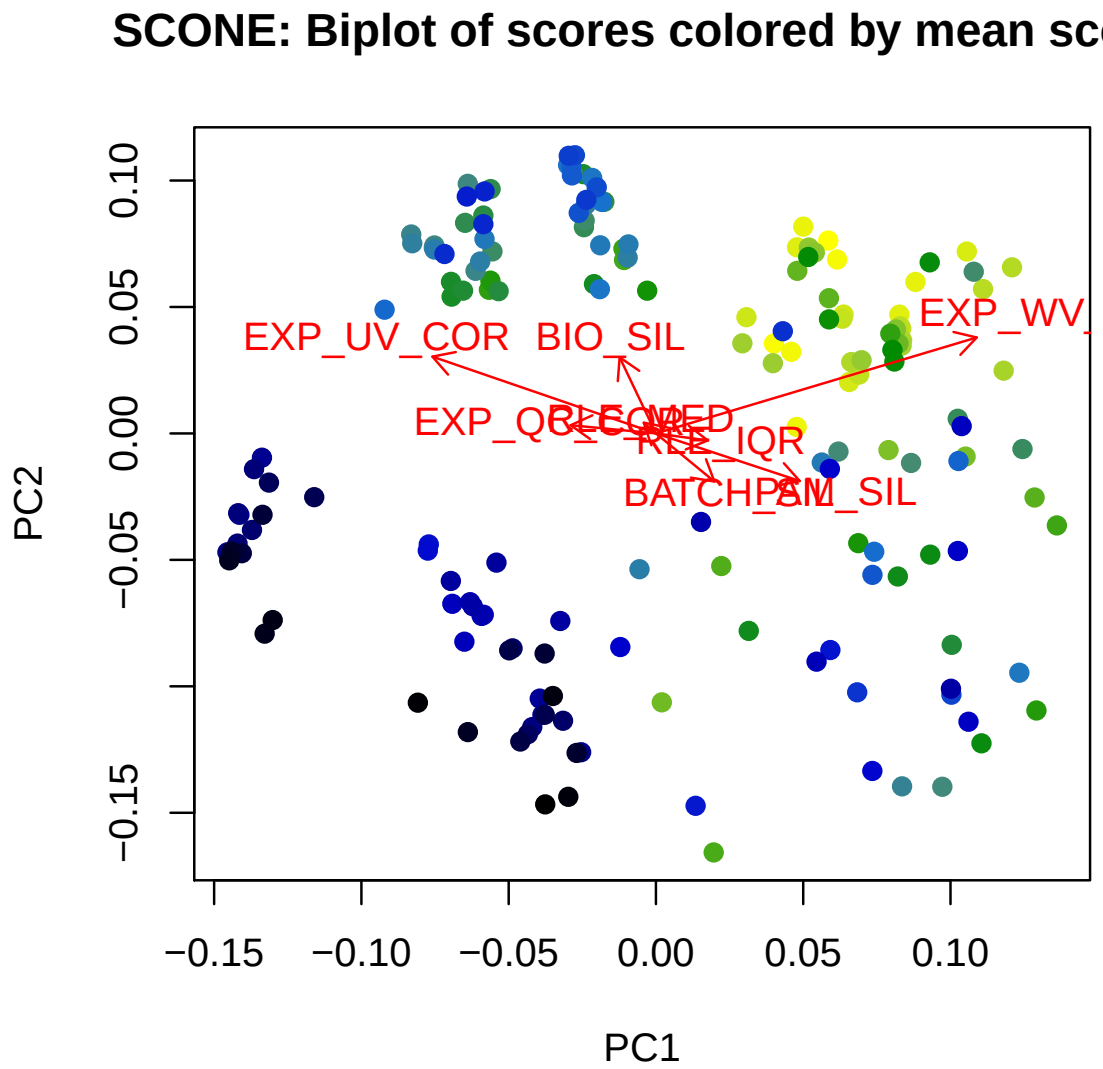


Figure 13: *scone*. Biplot of performance scores, colored by mean score (yellow high/good, blue low/bad).

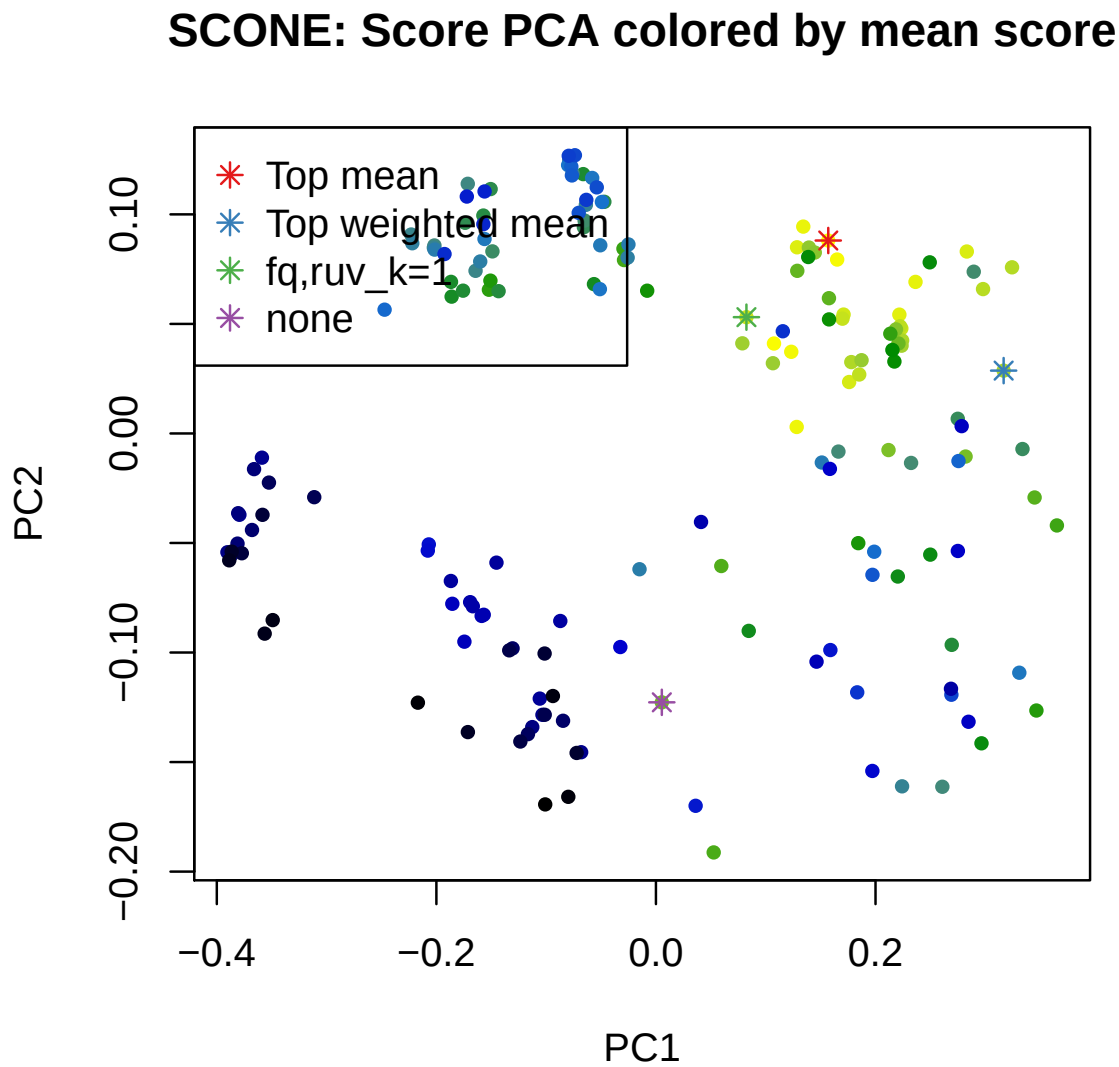


Figure 14: *scone*. PCA of performance scores, colored by mean score.

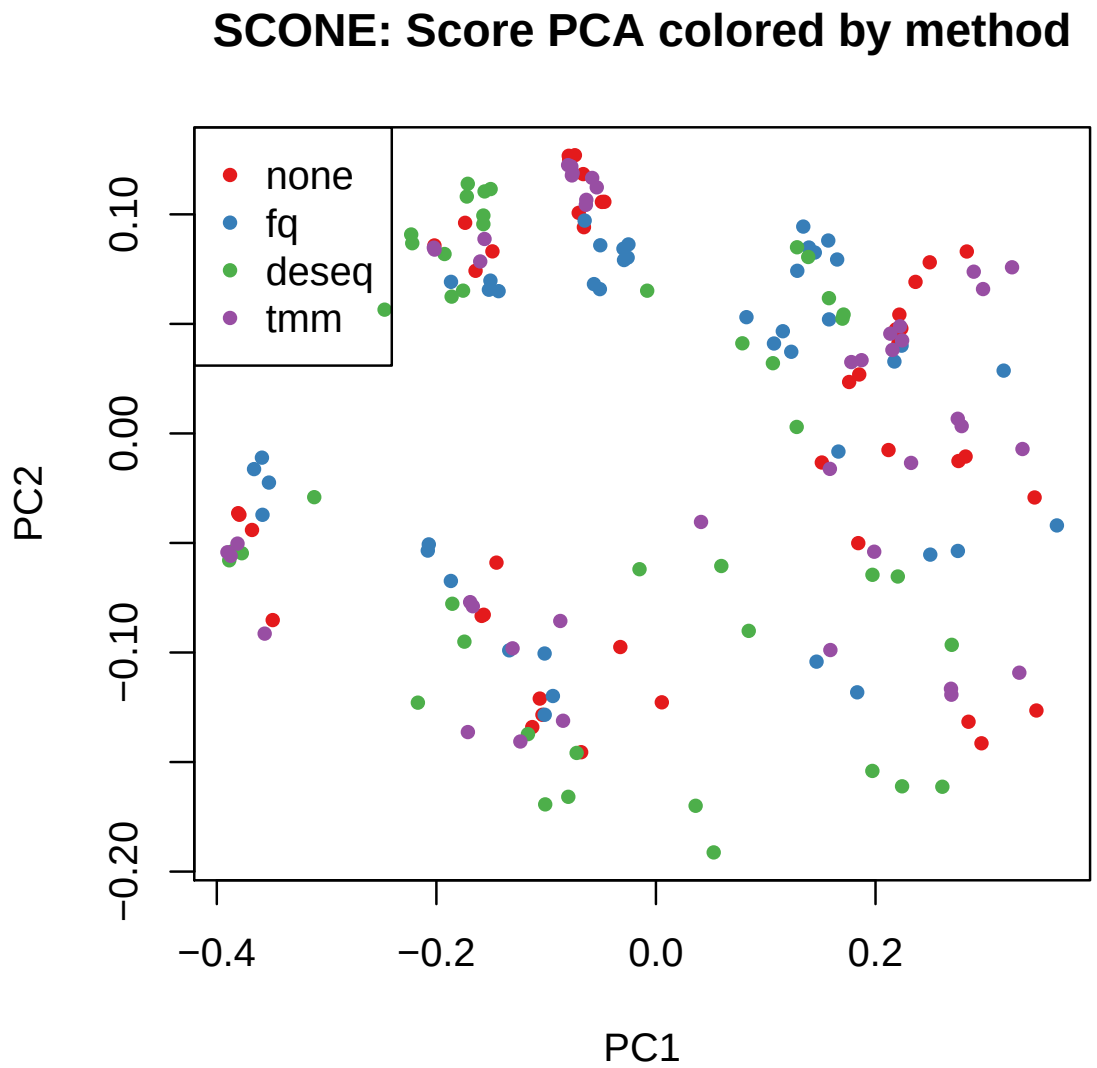


Figure 15: *scone*. PCA of performance scores, colored by method – `scaling_method`.

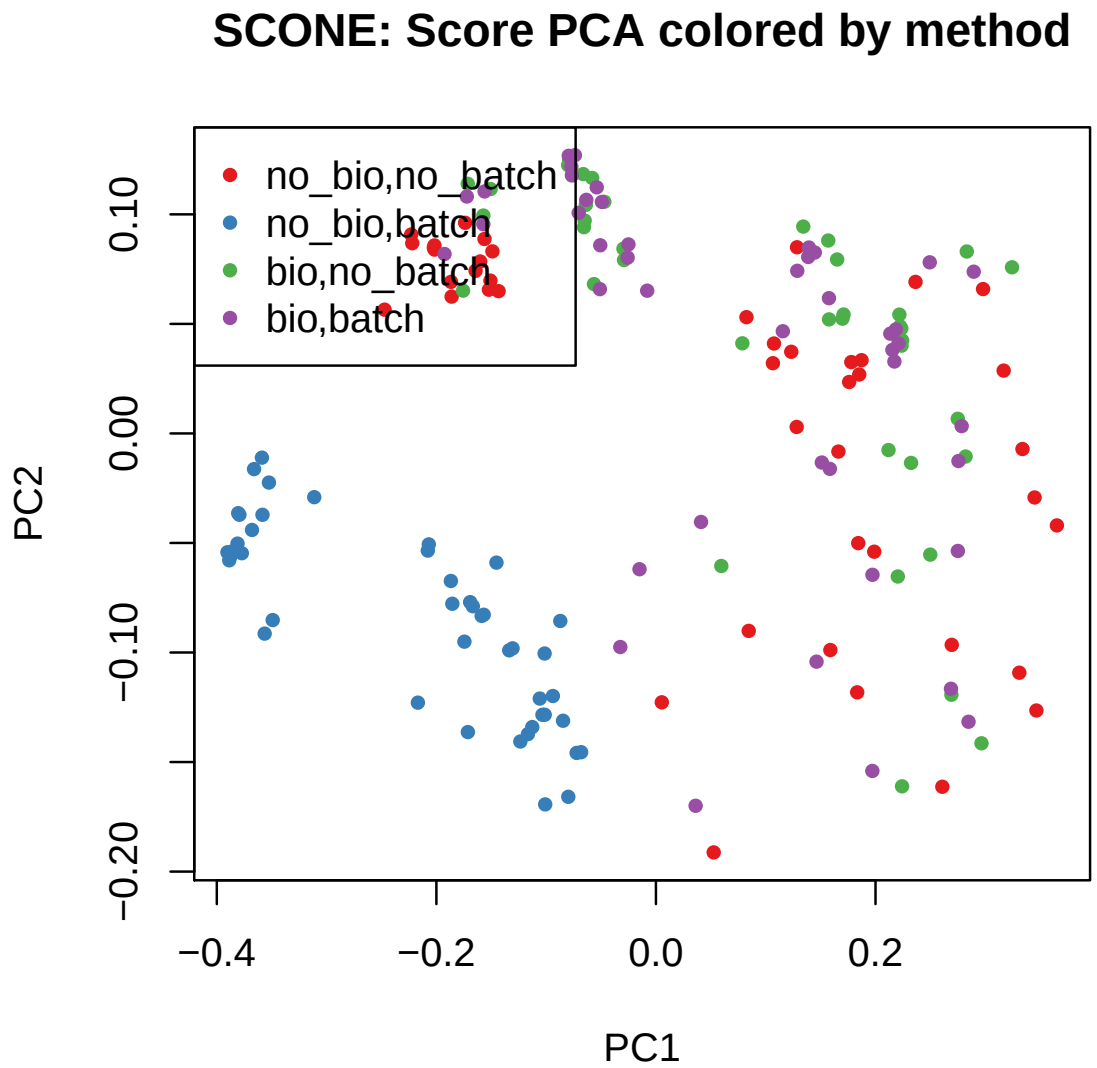
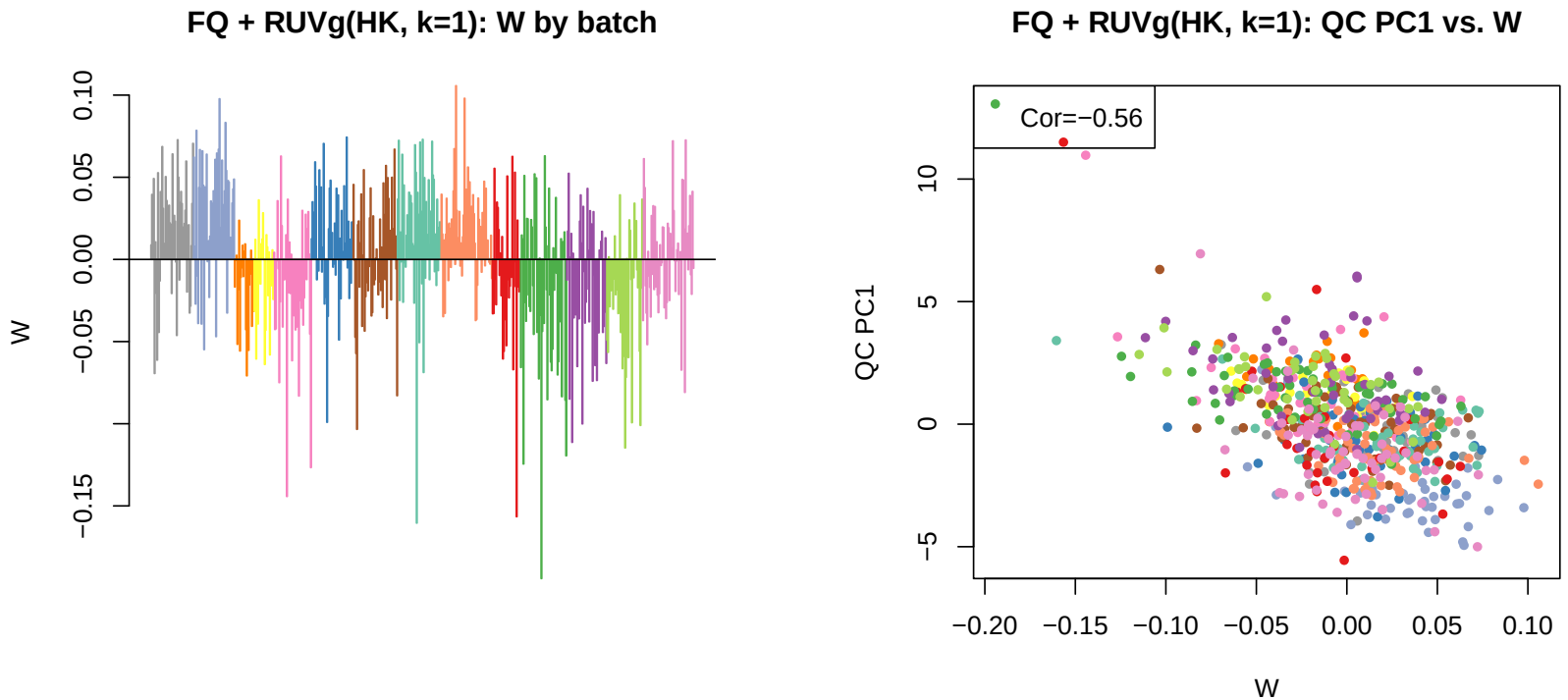


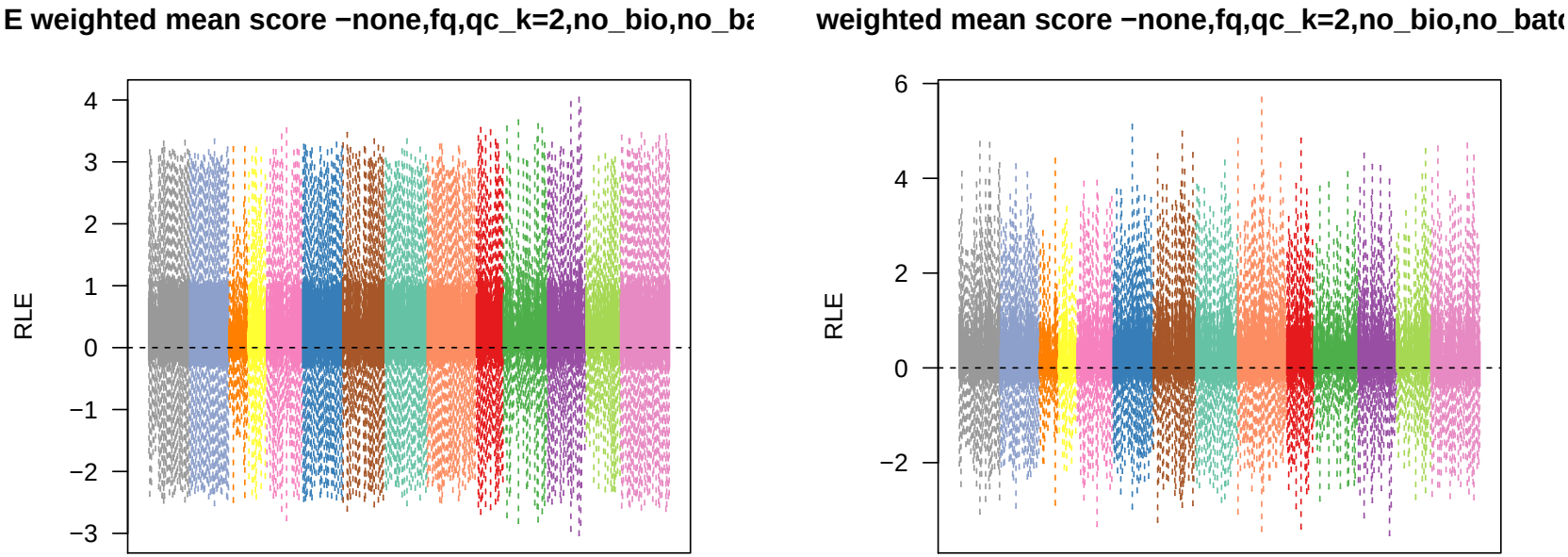
Figure 16: *scone*. PCA of performance scores, colored by method – `adjust_biology`, `adjust_batch`.



(a) Unwanted factor *W*

(b) QC PC1 vs. *W*

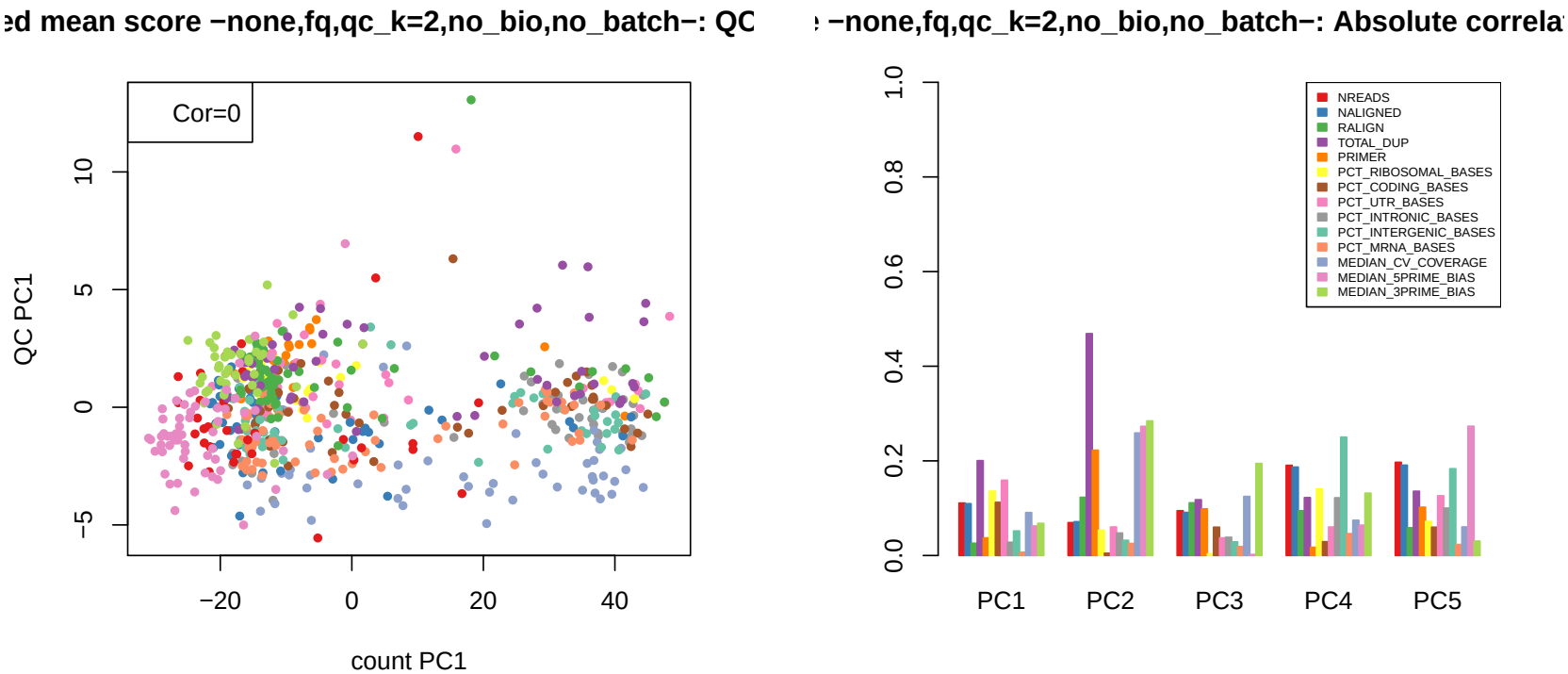
Figure 17: *scone*. Association of RUVg unwanted factor *W* and QC measures for `none,fq,ruv_k=1,no_bio,batch`.



(a) All genes

(b) Housekeeping genes

Figure 18: *scone.* Gene-level relative log expression (RLE = log-ratio of read count to median read count across samples) for method with top weighted mean score `none,fq,qc_k=2,no_bio,no_batch`.



(a) QC PC1 vs. count PC1

(b) Correlation of count PC and QC measures

Figure 19: *scone*. Association of counts and sample-level QC measures, `none,fq,qc_k=2,no_bio,no_batch`.

- Unnormalized gene-level counts exhibit large differences in distributions within and between batches and association with sample-level QC measures.
- Different normalization methods vary in performance according to SCONE metrics and lead to different distributions of gene-level counts, hence clustering and DE results.
- Global-scaling normalization. Not aggressive enough to handle potentially large batch effects and association of counts and QC measures. Biological effects are dominated by nuisance technical effects. Additionally, for DESeq, the scaling factors are computed based on only a handful of genes with non-zero counts in all cells (5/22,054).

- Batch effect normalization. Adjusting for batch effects without properly accounting for the nesting of batch within biological effects (`no_bio,batch`) in the regression model is problematic, as this removes the biological effects of interest (e.g., empirical Bayes framework of ComBat).
- FQ followed by QC-based or RUVg normalization. Seems effective: Similar RLE distributions between samples, lower association of counts and QC measures. The first unwanted factor of RUVg is correlated with the first QC PC.
- The remaining analyses are based on `none,fq,qc_k=2,no_bio,no_batch`, the best method according to weighted mean score.

- Interpretation of performance metrics. Some metrics tend to favor certain methods over others, e.g., EXP_UV_COR (correlation between count PCs and factors of unwanted variation) naturally favors RUVg, especially when the same set of negative controls are used for normalization and evaluation. Hence, a careful, global interpretation of the metrics is recommended.
- Negative controls. The selection of proper, distinct sets of negative controls is important, as these are used for both normalization (RUVg) and assessment of normalization results (EXP_UV_COR).
- Ongoing efforts.
  - Zero-inflated negative binomial (ZINB) model.
  - User-supplied factors unwanted and wanted variation (UV and WV, respectively).
  - Other methods (e.g., ComBat/sva).

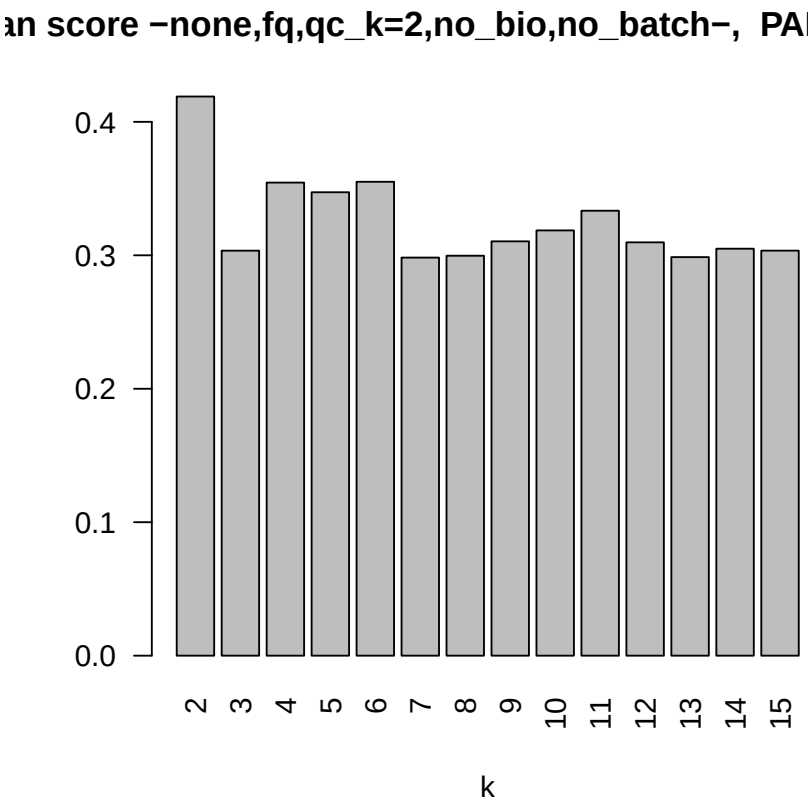
- Other performance metrics.
- Visualization.
- `Shiny` app for interactive web interface.

D. Risso, E. Purdom

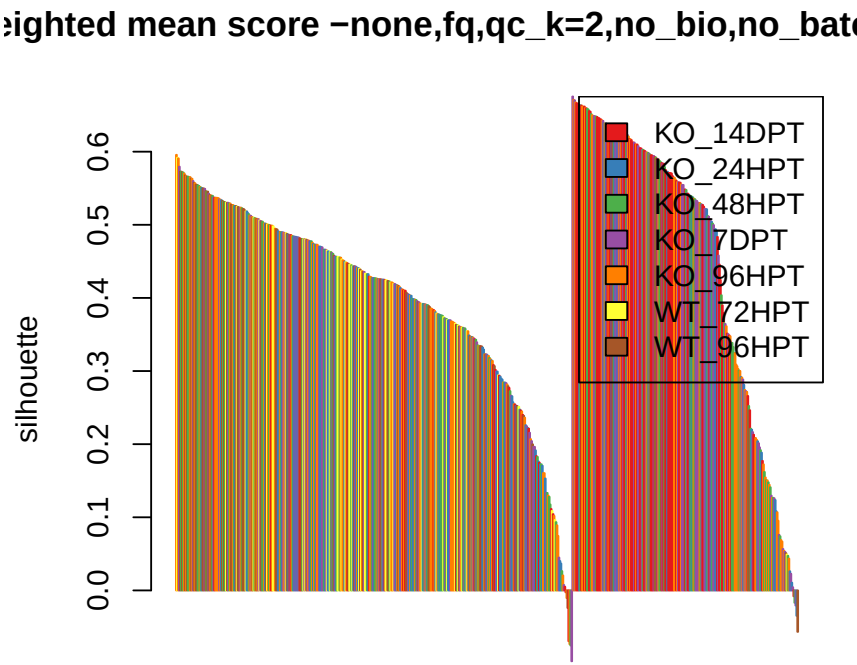
- Robustness to choice of samples. Both hierarchical and partitioning methods tend to be sensitive to the choice of samples to be clustered. Outlying samples/clusters (e.g., glia) are common in scRNA-Seq and mask interesting substructure in the data, often requiring the successive pruning out of dominating clusters to get to the finer structure.
- Robustness to clustering algorithm and tuning parameters. Clustering results are sensitive to pre-processing steps such as normalization and dimensionality reduction, as well as to the choice of clustering algorithm and associated tuning parameters (e.g., distance function, number of clusters).

- Not focusing on the number of clusters. A major tuning parameter of partitioning methods such as partitioning around medoids (PAM) and k-means is the number of clusters k. Methods for selecting k (e.g., silhouette width) are sensitive to the choice of samples, normalization, and other tuning parameters. They tend to be conservative (low k), i.e., capture only the coarse clustering structure and mask interesting substructure in the data. Additionally, the number of clusters k is often not of primary interest.
  E.g. Silhouette width with PAM selects only $k = 2$ clusters for the OE p63 dataset.
- Not forcing samples into clusters. Some samples may be outliers, that do not really belong to any clusters. Leaving them out can improve the quality and interpretability of the clustering as well as downstream analyses (e.g., identification of cluster marker genes).

- Cluster gene expression signatures. Common differential expression statistics are not well-suited for finding marker genes for the clusters, especially for finer structure in a hierarchy.



(a) Average silhouette width



(b) Silhouette widths, $k = 2$

Figure 20: *Partitioning around medoids.* 20 PC, one-minus-correlation distance.

Resampling-Based Sequential Ensemble Clustering

- We have developed a resampling-based sequential ensemble clustering approach, with the aim of obtaining stable and tight clusters.
- Ensemble clustering, i.e., aggregating multiple clusterings obtained from different algorithms or applications of a given algorithm to resampled versions of the learning set, is a general approach for improving stability. This can be viewed as the unsupervised analog of ensemble methods in supervised learning, e.g., bagging, boosting, random forests.
- Our approach is related to bagged/consensus/tight clustering (Dudoit and Fridlyand, 2003; Leisch, 1999; Tseng and Wong, 2005).
- R package clusterExperiment, released through the Bioconductor Project.

Resampling-Based Sequential Ensemble Clustering

RSEC: Resampling-based Sequential Ensemble Clustering.

- Given a base clustering algorithm (e.g., PAM, k-means) and associated tuning parameters (e.g., number of principal components, number of clusters k, distance matrix), generate a single candidate clustering using
  - resampling-based clustering to find robust and tight clusters;
  - sequential clustering to find stable clusters over a range of numbers of clusters (Tseng and Wong, 2005).
- Generate a collection of candidate clusterings by repeating the above procedure for different base clustering algorithms and tuning parameters.
- Identify a consensus over the different candidate clusterings.
- Merge non-differential clusters.
- Find cluster signatures by testing for differential expression between selected subsets of clusters.

- Visualization. Comparison of multiple clusterings of the same samples, heatmaps of co-clustering matrices, heatmaps with hierarchical clustering of genes and/or samples.

- Consider a particular base partitioning clustering algorithm (e.g., PAM, k-means) and associated tuning parameters (e.g., number of principal components, number of clusters k, distance matrix).
- Generate B subsamples of the n observations to be clustered, e.g., bootstrap or random sample without replacement (of size $0.7n$, say).
- For each subsample, apply the base clustering algorithm (with k clusters) and assign all n original observations to clusters based on their distances from cluster centroids/medoids.
- Generate an $n \times n$ co-clustering matrix C by recording for each pair of observations the proportion of times they are clustered together (out of B subsamples).

- Identify tight clusters based on the co-clustering matrix. For example, given a constant α close to 0, cut branches top-down from a hierarchical clustering based on the co-clustering matrix C, such that $C_{i,j} \geq 1 - \alpha \ \forall i, j$ in the selected branch (or some other less stringent function than min). Retain all clusters above a certain size or the largest certain number of clusters.

Based on Tseng and Wong (2005).

- Starting with a user-supplied $k = k_0$, generate a collection of clusterings, $\{\mathcal{C}_1^k, \mathcal{C}_2^k, \ldots, \mathcal{C}_{m_k}^k\}$, by applying the above resampling-based procedure to the base partitioning clustering algorithm with k clusters.
- Given a constant β close to 1, increase k by one until $s(\mathcal{C}_i^k, \mathcal{C}_j^{k+1}) \geq \beta$, for some i, j, where s is a measure of similarity between clusters.
- Identify $\mathcal{C}_j^{k+1}$ as a stable cluster.
- Remove $\mathcal{C}_j^{k+1}$ and repeat the above steps with $k \leftarrow k - 1$, until k is below a certain threshold or a certain number of clusters are identified.

- Here, we evaluate a collection of clusters for stability based on the pairwise similarity measure

$$s(\mathcal{C}_i, \mathcal{C}_j) \equiv |\mathcal{C}_i \cap \mathcal{C}_j| / |\mathcal{C}_i \cup \mathcal{C}_j|. \quad (1)$$

- Generate a collection of candidate clusterings as above, for a range of base clustering algorithms and tuning parameters.
- Find a single consensus clustering based on the co-clustering matrix (as in resampling, only now across different algorithms or tuning parameters).
- Merge non-differential clusters by creating a hierarchy of clusters, working up the tree, testing for differential expression between sister nodes, and collapsing nodes with insufficient DE.

- Find cluster gene expression signatures, i.e., marker genes, by testing for differential expression between selected subsets of clusters.
- Standard F-statistic. Tests for any difference between clusters. Sensitive to outlying samples/clusters. Non-specific, i.e., not useful for interpreting differences between clusters.
- Standard solution in (generalized) linear models/ANOVA is to consider contrasts between groups of clusters. By using the machinery of the (generalized) linear model, we use all of the samples in testing these contrasts, rather than just those samples involved in the corresponding clusters.
  - All pairwise. All pairwise comparisons between clusters.
  - One against all. Compare each cluster to union of remaining clusters.

  - Dendrogram. Create a hierarchy of clusters, work up the tree, test for DE between sister nodes (as in approach used for merging clusters).

- For each contrast, test for DE using empirical Bayes linear modeling approach of R package limma, with `voom` option to account for mean-variance relationship of log-counts (i.e., over-dispersion).

Workflow.

- `clusterMany`. Generate a collection of candidate clusterings, for different base clustering algorithms and tuning parameters, with option to use resampling and sequential approaches.
- `combineMany`. Find consensus clustering across several clusterings.
- Identify non-differential clusters that should be merged into larger clusters.
  - `makeDendrogram`. Hierarchical clustering of the clusters found by `combineMany`.
  - `mergeClusters`. Merge clusters of this hierarchy based on DE between nodes.
- `RSEC`. Wrapper function around the clusterExperiment workflow.

- `getBestFeatures`. Find cluster signatures by testing for differential expression between selected subsets of clusters.
- Visualization.
  - `plotClusters`. Comparison of multiple clusterings of the same samples. Based on ConsensusClusterPlus package.
  - `plotHeatmap`. Heatmaps of co-clustering matrices, heatmaps with hierarchical clustering of genes and/or samples (interface to `aheatmap` from NMF package).

Application to OE p63 dataset.

- `clusterMany`: Generate 22 candidate clusterings.
  - Dimensionality reduction: 25, 50 PC.
  - Euclidean distance.
  - Base clustering method: PAM, $k = 5, \cdots, 15$.
  - Resampling-based clustering: $B = 100$, proportion $= 0.7$, $\alpha = 0.3$.
  - Sequential clustering: $k_0 = 15$, $\beta = 0.9$.
  - `clusterFunction=c("hierarchical01")`.
- `combineMany(ce, clusterFunction="hierarchical01", whichClusters="workflow", proportion=0.7, propUnassigned=0.5, minSize=5)`.
- `mergeClusters(ce, mergeMethod="adjP", cutoff=0.05)`.

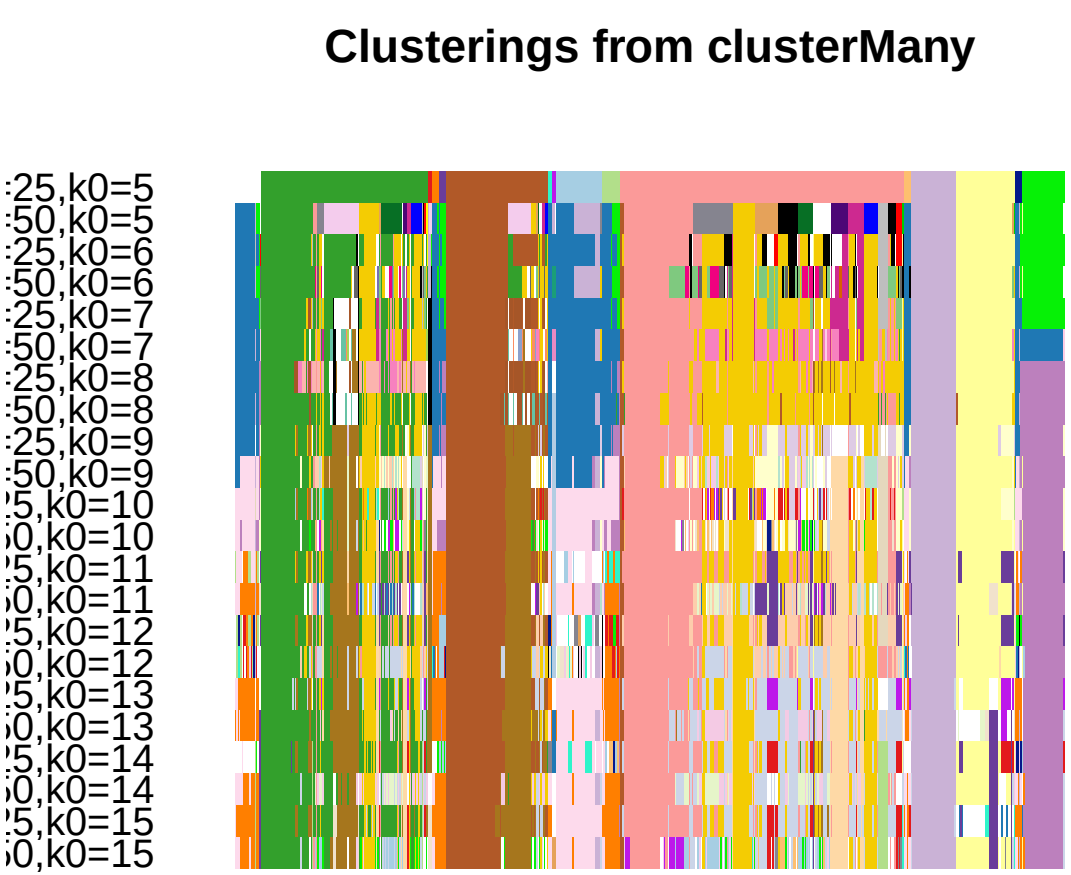


Figure 21: *clusterExperiment*. Comparison of 22 `clusterMany` clusterings using `plotClusters`.

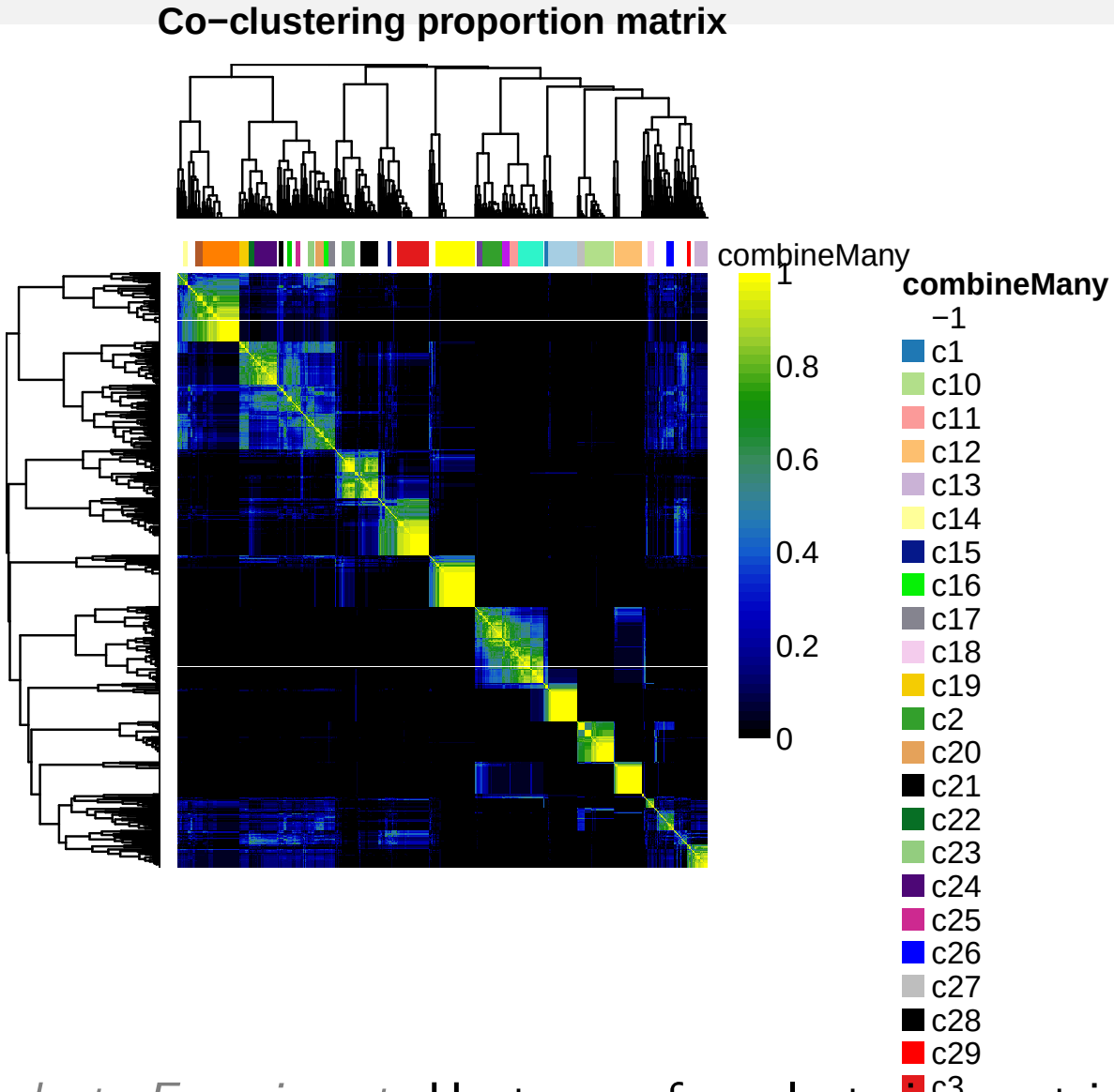


Figure 22: *clusterExperiment*. Heatmap of co-clustering matrix for `clusterMany` clusterings, used to create `combineMany` clustering (`plotHeatmap`).

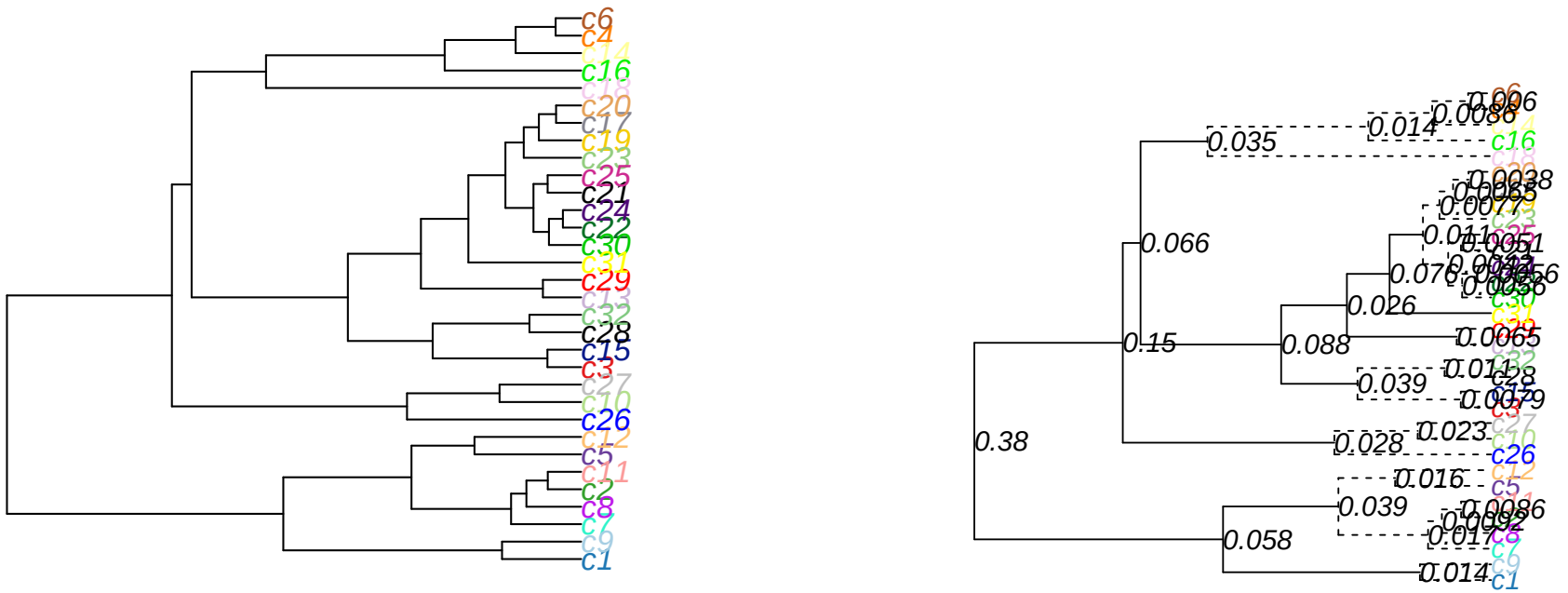


Figure 23: *clusterExperiment*. Dendrograms from `combineMany` and `mergeClusters`.

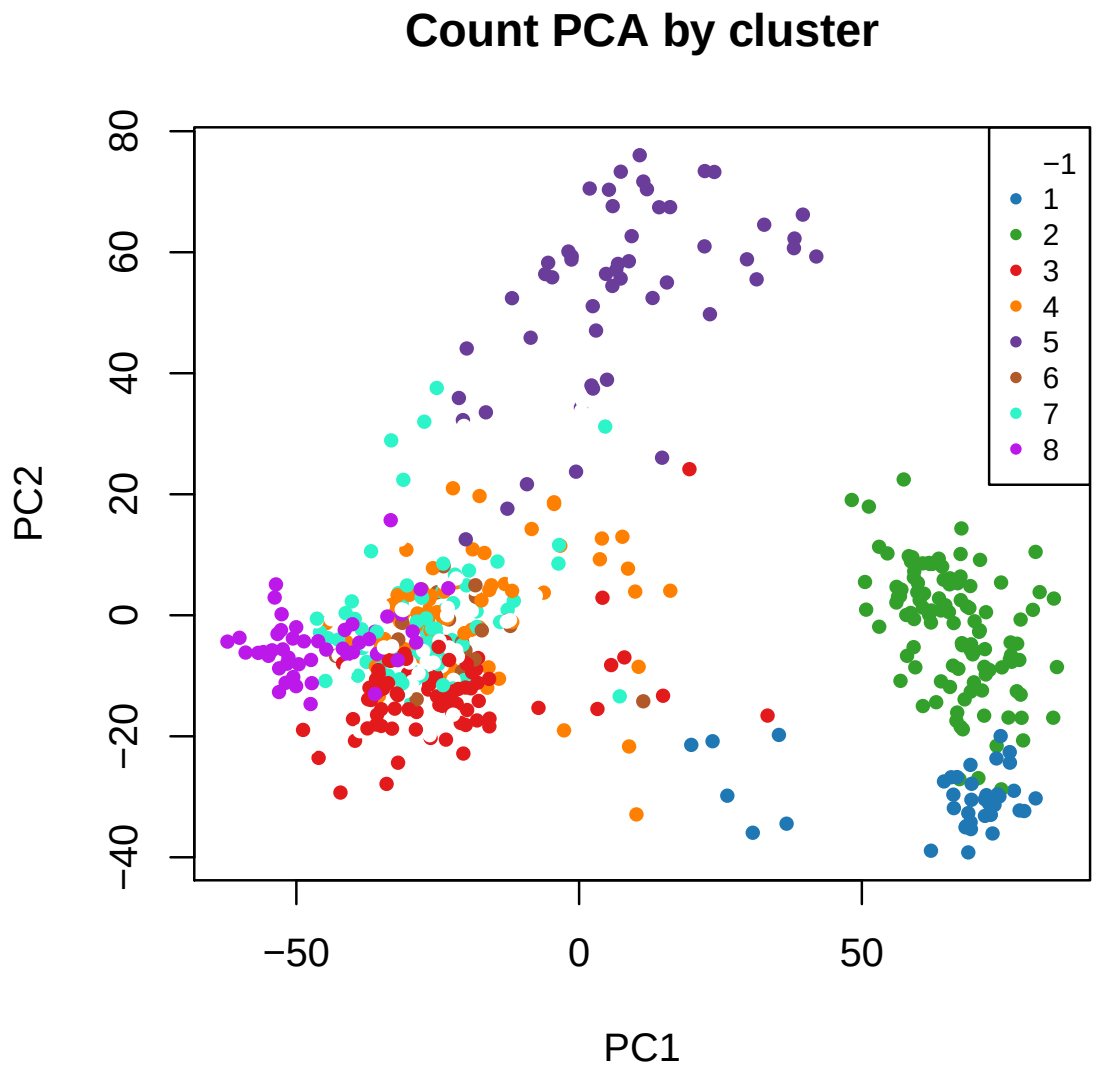


Figure 24: *clusterExperiment*. PCA of gene-level log-counts, colored by `mergeClusters`.

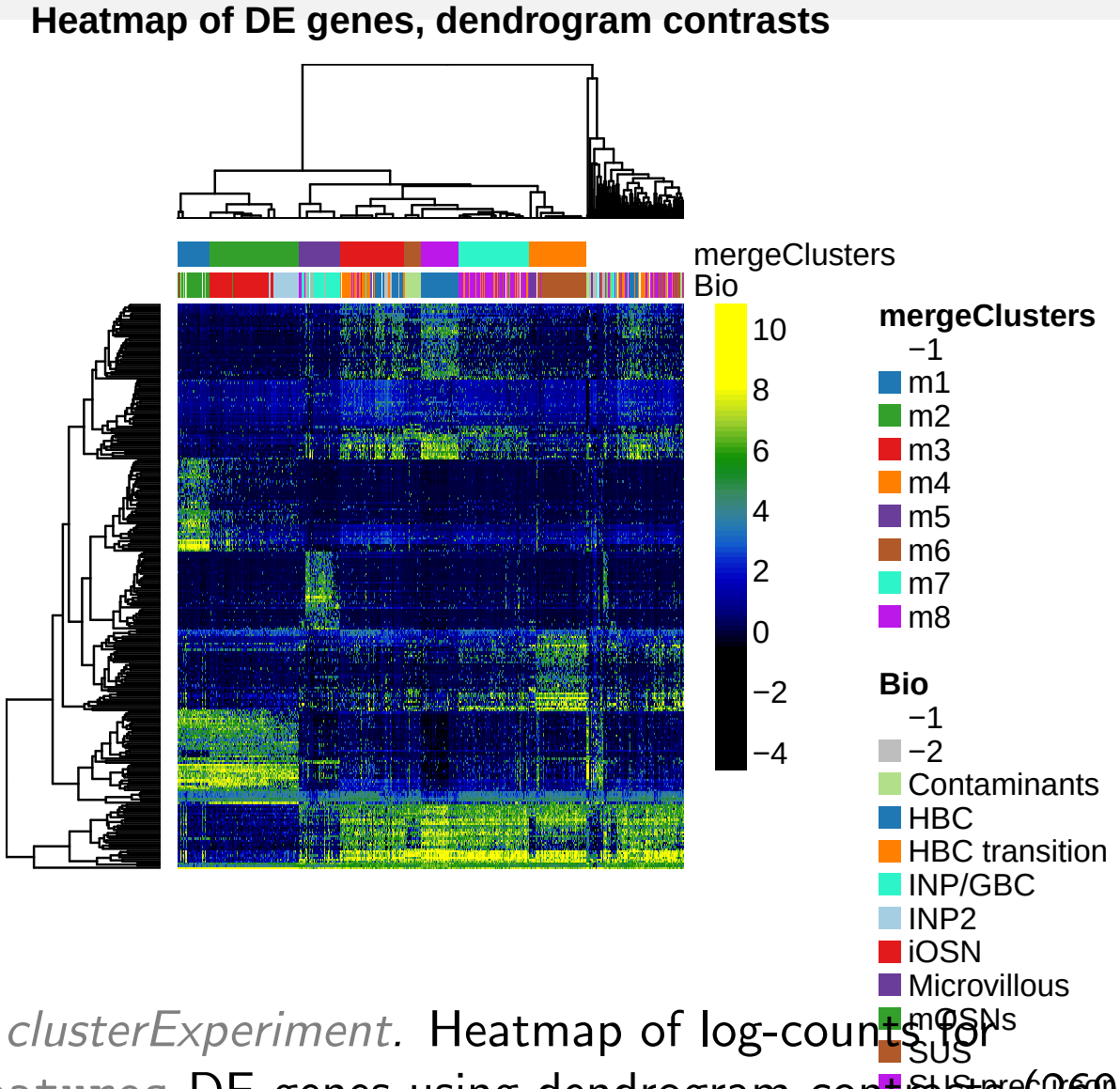


Figure 25: *clusterExperiment*. Heatmap of log-counts for `getBestFeatures` DE genes using dendrogram contrasts (269, top 50 in each of the 6 nodes).

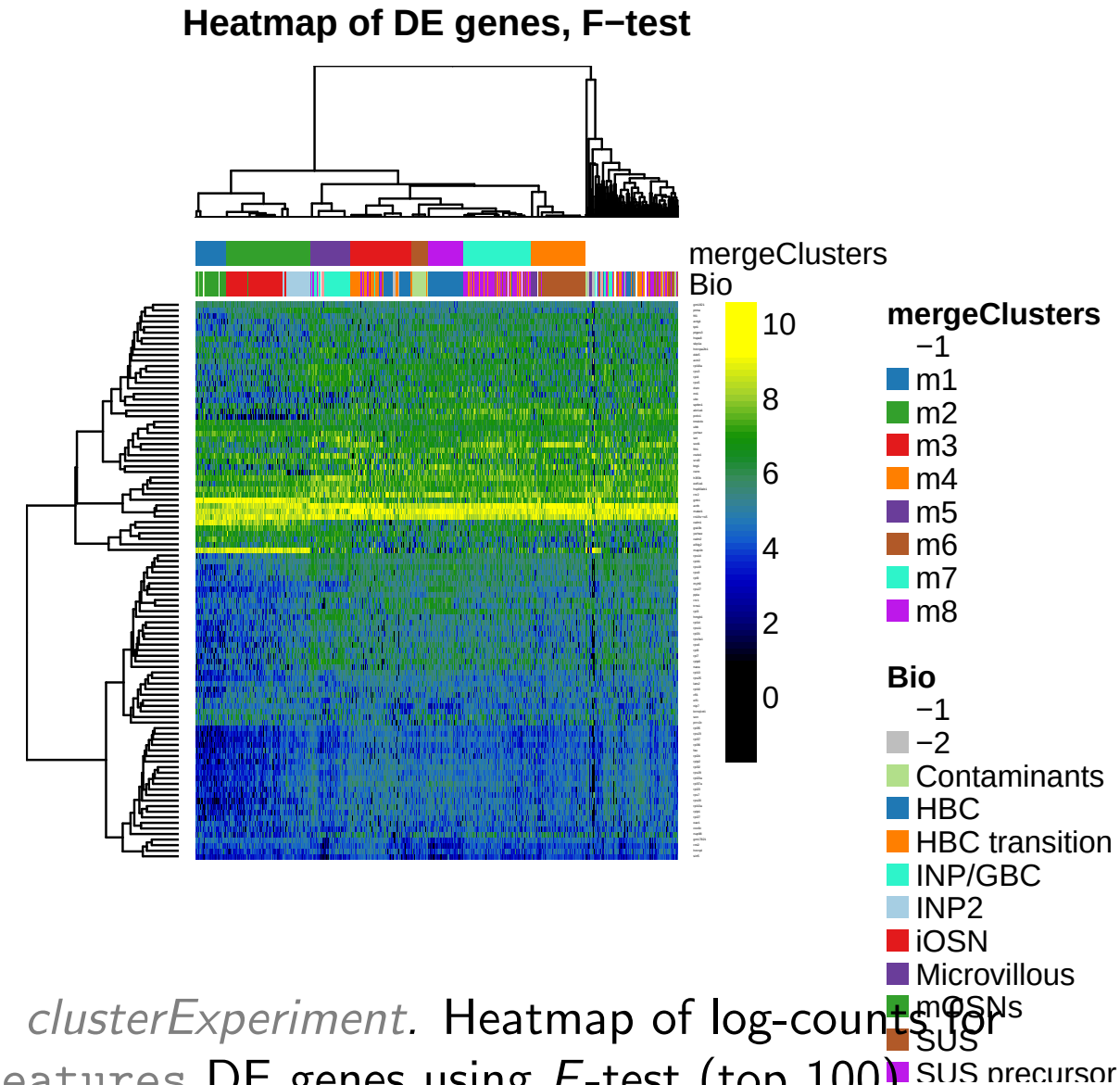


Figure 26: *clusterExperiment*. Heatmap of log-counts for `getBestFeatures` DE genes using F-test (top 100).

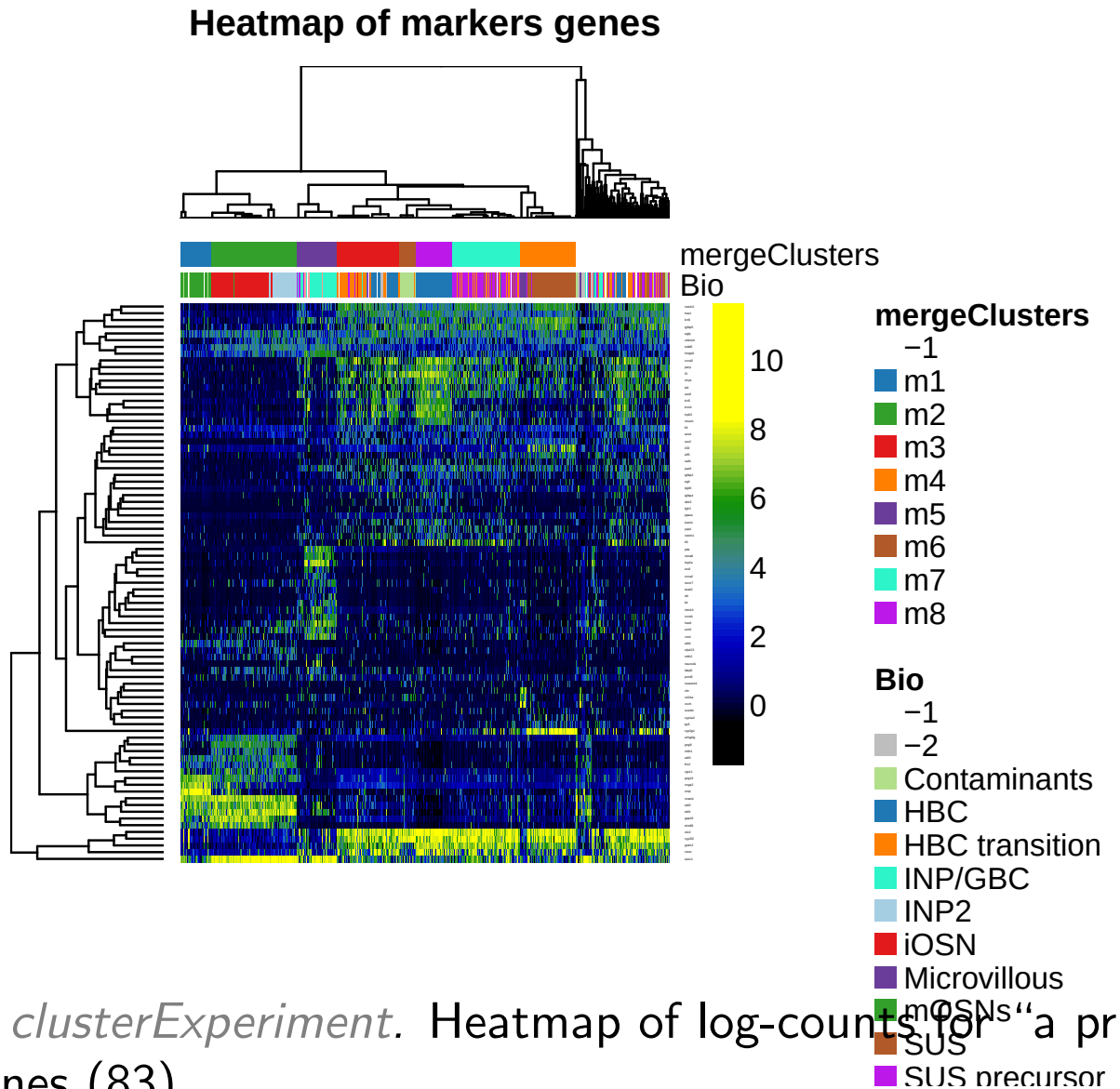


Figure 27: *clusterExperiment*. Heatmap of log-counts for "a priori" markers genes (83).

- Tuning parameters.
  - α controls tightness.
  - β controls stability.
  - k_0, the initial number of clusters used in sequential clustering, is the parameter with the greatest impact on the results. Larger k_0 tend to lead to smaller and tighter clusters.
- Caveat. The DE analysis is exploratory and nominal p-values only a rough summary of significance (reliance on models, tiny p-values even after adjustment for multiple testing, same data used to define clusters and to perform DE analysis).
- Ongoing efforts.
  - Cluster confidence measures.
  - Potentially assign unclustered observations to clusters.
  - DE using ZINB model.
  - Greater modularity (e.g., distance functions, DE test).
  - `Shiny` app for interactive web interface.

K. Street, D. Risso, E. Purdom

- Mapping transcriptional progression from stem cells to specialized cell types is essential for properly understanding the mechanisms regulating cell and tissue differentiation.
- There may not always be a clear distinction between states, but rather a smooth transition, with individual cells existing on a continuum between states.
- In such a case, cells may undergo gradual transcriptional changes, where the relationship between states can be represented as a continuous lineage dependent upon an underlying spatial or temporal variable. This representation, referred to as pseudotemporal ordering, can help us understand how cells differentiate and how cell fate decisions are made (Bendall et al., 2014; Campbell et al., 2015; Ji and Ji, 2016; Petropoulos et al., 2016; Shin et al., 2015; Trapnell et al., 2014).

- We have developed Slingshot as a flexible and robust framework for inferring cell lineages and pseudotimes in the study of continuous differentiation processes.

- Input/Output.
  - Input. Normalized gene expression measures and cell clustering.
  - Output. Cell lineages, i.e., subsets of ordered cell clusters. Cell pseudotimes, i.e., for each lineage, ordered sequence of cells and associated pseudotimes.
- Dimensionality reduction.
  - Principal component analysis (PCA) seems effective and simple, in conjunction with steps detailed next.
  - Other approaches include related linear methods, e.g., independent component analysis (ICA) (Trapnell et al., 2014, Monocle), and non-linear methods, e.g., Laplacian eigenmaps/spectral embedding (Campbell et al., 2015, Embeddr), t-distributed stochastic neighbor embedding (t-SNE) (Bendall et al., 2014; Petropoulos et al., 2016, Wanderlust).

- Inferring cell lineages.
  - Minimum spanning tree (MST; ape package) over cell clusters, with between-cluster distance based on Euclidean distance between cluster means scaled by within-cluster covariance.
  - Outlying clusters. Identified using granularity parameter ω that limits maximum edge weight in the tree. Specifically, build MST using an artificial cluster Ω, with distance ω from other clusters (a fraction of maximum pairwise distance between clusters), and then remove Ω.
  - Root and leaf nodes. May either be pre-specified or automatically selected.
    Root node. If not pre-specified, selected based on parsimony (i.e., set of lineages with maximal number of clusters shared between them).
    Leaf nodes. If pre-specified, constrained MST.
  - A lineage is then defined as any unique path coming out of the root node and ending in a leaf node.

  - Constructing the MST on clusters (Ji and Ji, 2016; Shin et al., 2015, TSCAN,Waterfall) vs. cells (Trapnell et al., 2014, Monocle) offers greater stability and computational efficiency, less complex lineages, and easier determination of directionality and branching.

- Inferring cell pseudotimes.
  - Iterative procedure inspired from the principal curve algorithm of Hastie and Stuetzle (1989); `principal.curve` function in princurve package.
  - In the case of branching lineages, a shrinkage step is included at each iteration, that forces a degree of similarity between the curves in the neighborhood of shared clusters.
  - Pseudotime values are derived by orthogonal projection onto the curves.
  - Cells belonging to clusters that are included in multiple lineages have multiple, similar pseudotime values.

  - Previous approaches also use smooth curves to represent lineages (Campbell et al., 2015; Petropoulos et al., 2016, Embeddr), while others use piecewise linear paths through the MST and extract orderings either by orthogonal projection (Ji and Ji, 2016; Shin et al., 2015, TSCAN,Waterfall) or PQ tree (Trapnell et al., 2014, Monocle).
  - We find that smooth curves provide discerning power not found in piecewise linear trajectories, while also adding stability over a range of dimensionality reduction and clustering methods.
- Differential expression. Regression of gene expression measures on pseudotime, e.g., generalized additive models (GAM) (Ji and Ji, 2016, TSCAN).
- Visualization. Two- and three-dimensional plots of cell lineages and pseudotimes, gene-level trajectories, heatmaps for DE genes.

- R package slingshot, to be released through the Bioconductor Project: `github.com/kstreet13/slingshot`.

- Modularity.
  - Integrates easily with a range of normalization, clustering, and dimensionality reduction methods.
  - `get_lineages`: Given expression measures and cluster labels, use MST to infer lineages.
    `get_lineages(X, clus.labels, start.clus = NULL, end.clus = NULL, dist.fun = NULL, omega = Inf, distout = FALSE)`.
  - `get_curves`: Given lineages, infer pseudotimes.
    `get_curves(X, clus.labels, lineages, thresh = 1e-04, maxit = 100, stretch = 2, shrink = TRUE)`.
- Flexibility. Can be used with varying levels of supervision.
  - Cluster-based approach allows for easy supervision when researchers have prior knowledge of cell classes, while still being able to detect novel branching events.
  - User-supplied or data-driven selection of root and leaf nodes.

- Visualization.
  - `plot_tree`: MST in 2 and 3D.
  - `plot_curves`: Lineage curves in 2 and 3D.

Application to OE p63 dataset.

- Applied to the first three principal components, Slingshot identifies two lineages: The first corresponds to the HBC-to-neurons transition, the second to the HBC-to-sustentacular cells transition.
- A first-pass DE analysis, based on a regression of log-count on pseudotime using GAM, suggests that many genes are involved in the differentiation process.
- Among the top 100 DE genes for each lineage, only 13 are DE in both, suggesting distinct processes in the neuronal vs. non-neuronal lineages.

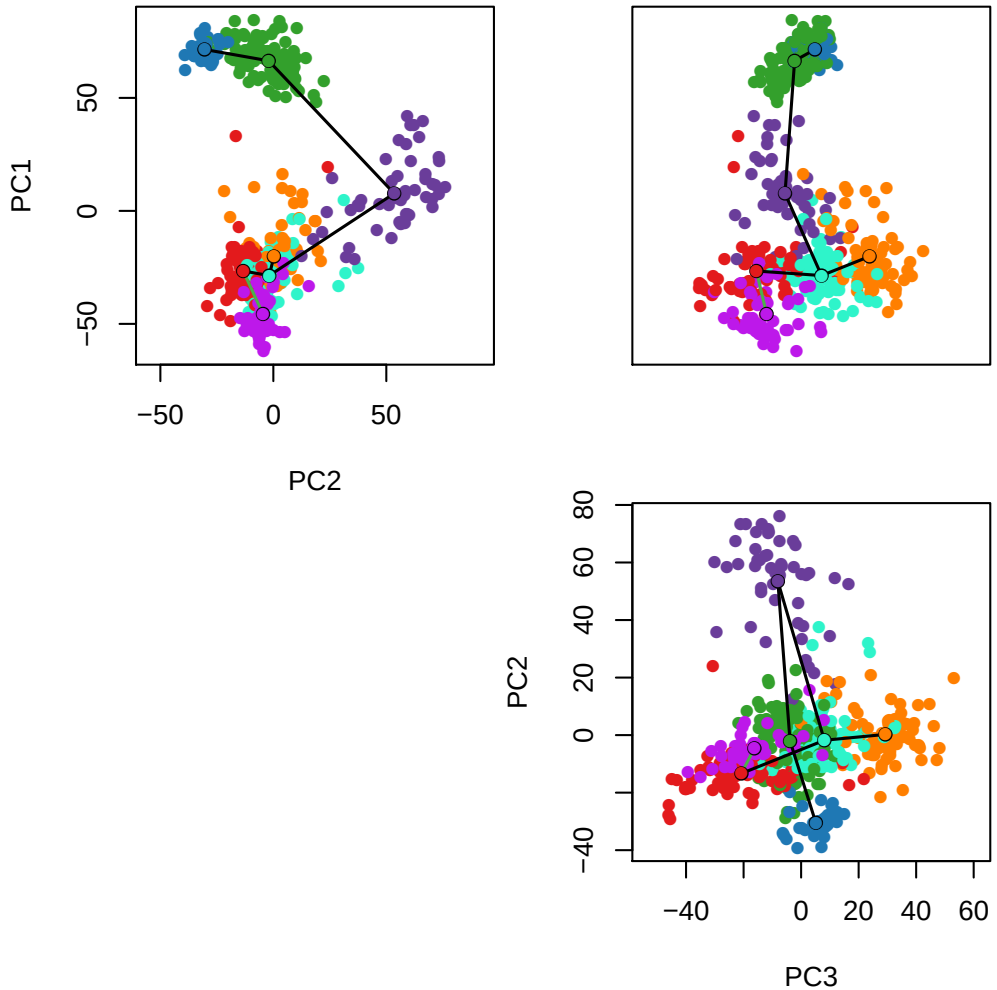


Figure 28: *slingshot*. PCA of gene-level log-counts, colored by clusters, with MST edges used to infer lineages (`get_lineages`, `plot_tree`).

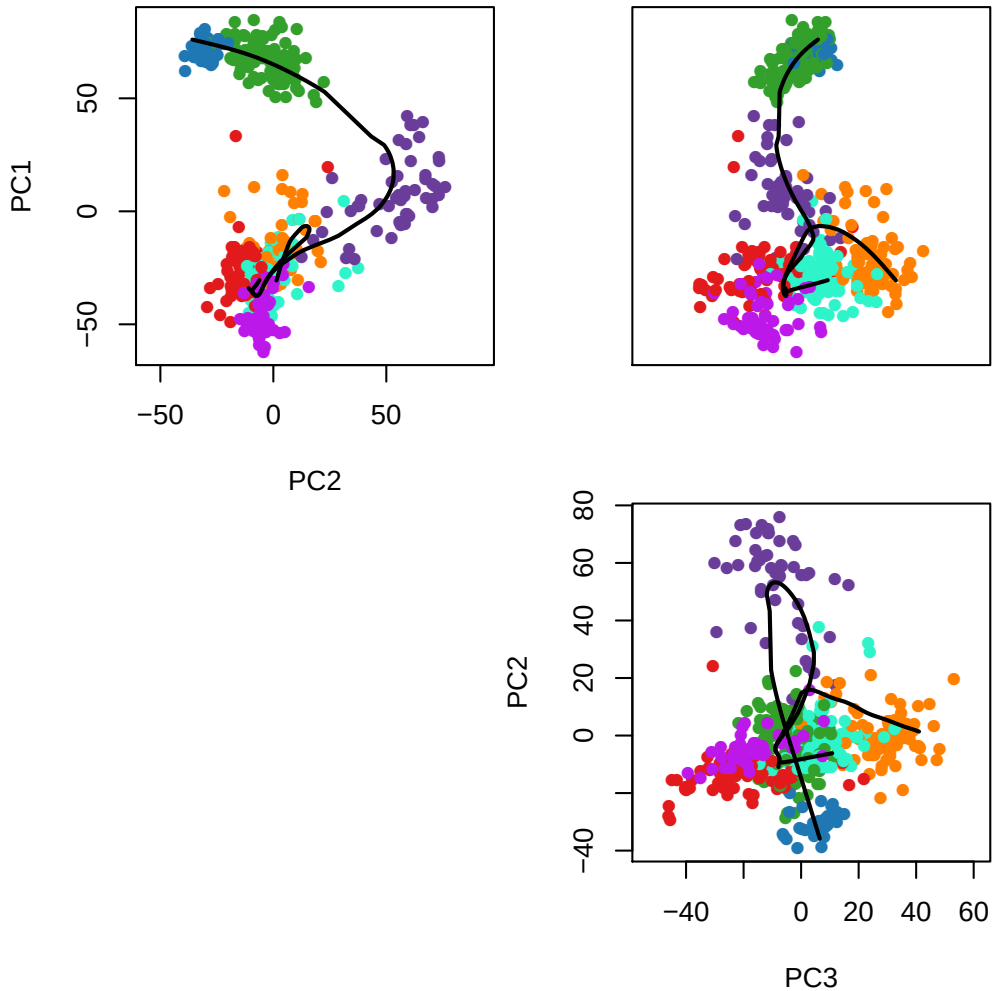


Figure 29: *slingshot*. PCA of gene-level log-counts, colored by clusters, with smooth curves representing lineages and used to infer pseudotimes (`get_curves`, `plot_curves`).

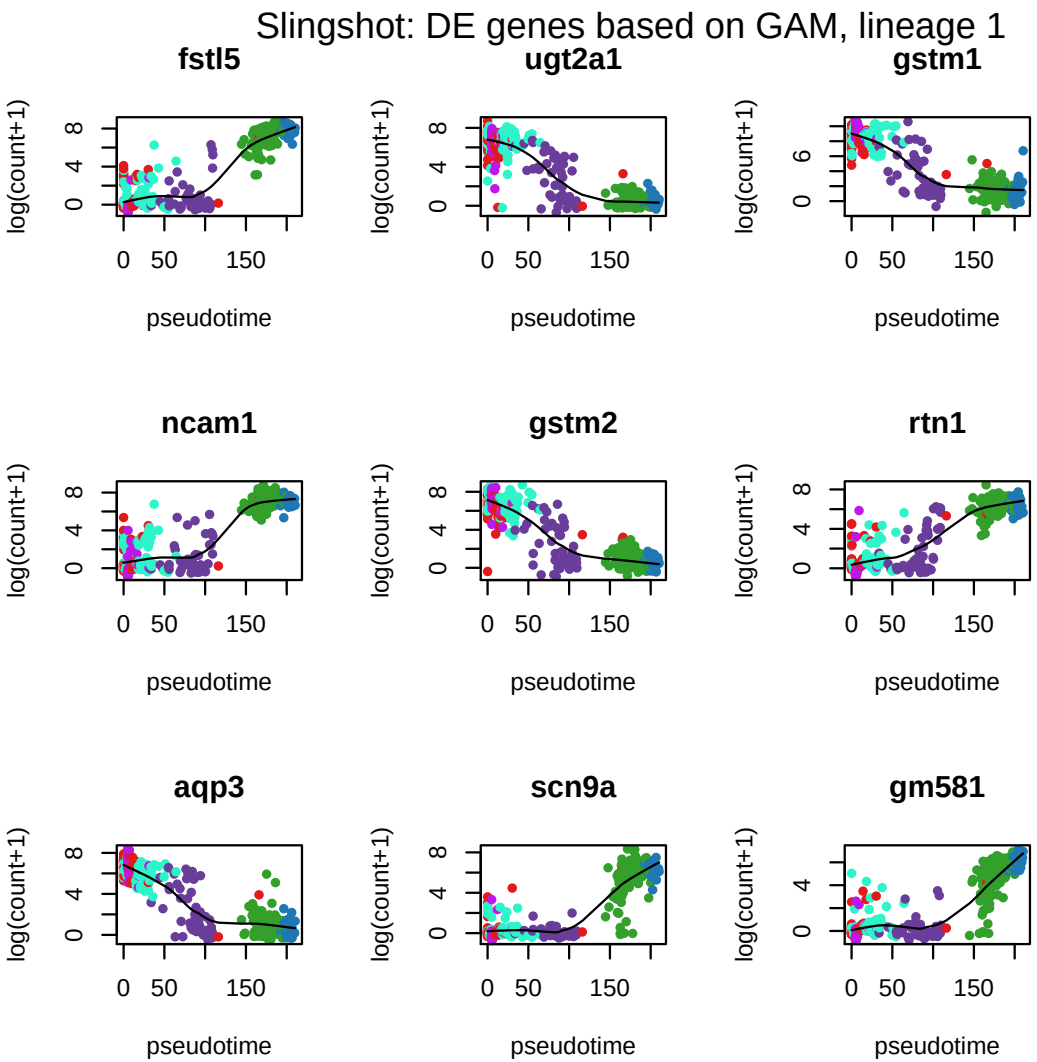


Figure 30: *slingshot*. Scatterplots of gene-level log-count vs. pseudotime for GAM DE genes in lineage 1 (HBC–Neurons).

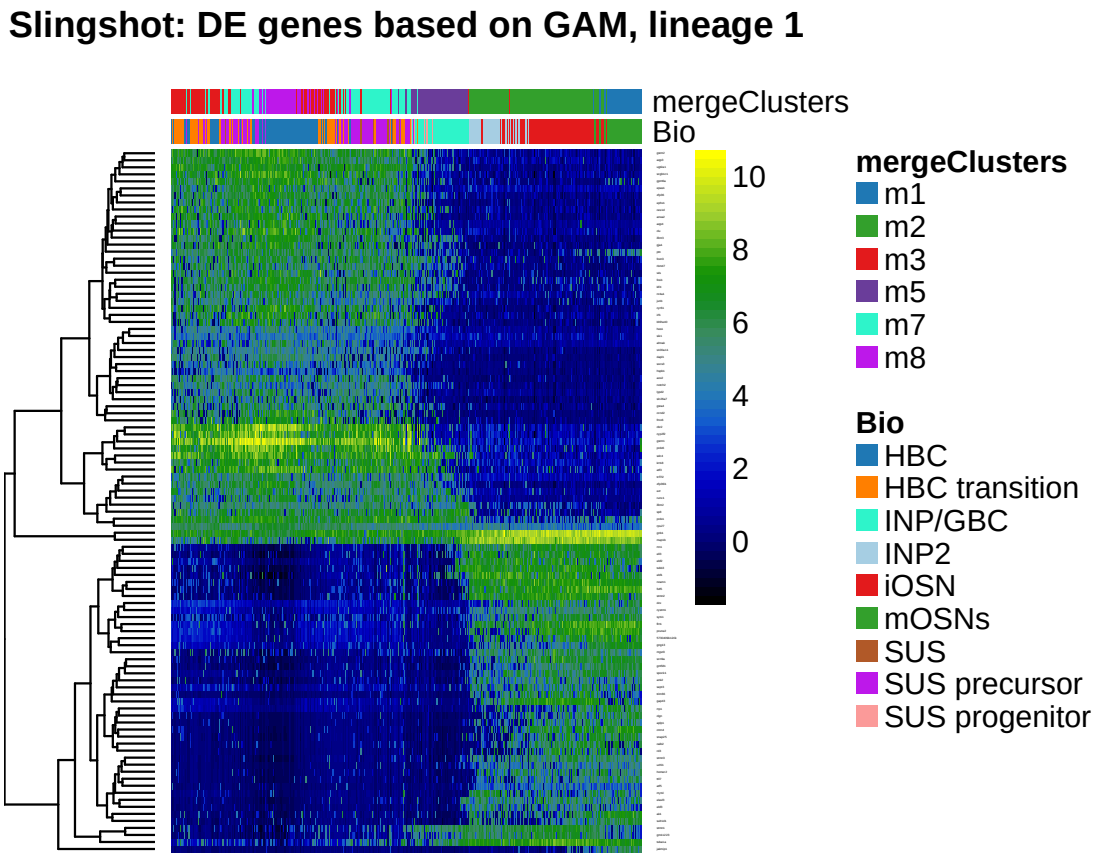


Figure 31: *slingshot*. Heatmap of log-counts for GAM DE genes in lineage 1 (HBC–Neurons), cells sorted by pseudotime.

Lineages.

```
$lineage1
[1] "8" "3" "7" "5" "2" "1"
$lineage2
[1] "8" "3" "7" "4"
```

Biological annotation of clusters.

	cl						
b	1	2	3	4	5	7	8
HBC	0	0	26	0	0	1	47
HBC transition	0	0	35	0	0	19	0
INP/GBC	0	0	1	0	41	0	0
INP2	0	34	0	0	0	0	0
iOSN	1	77	1	0	0	0	0
Microvillous	0	0	0	8	0	0	0
mOSNs	33	1	0	0	0	0	0
SUS	0	0	1	61	0	5	0
SUS precursor	0	0	7	2	4	62	0
SUS progenitor	0	0	0	0	6	0	0

Ongoing efforts.

- Number of lineages: User-supplied, testing for distinct lineages, merging non-differential lineages.
- DE within and between (i.e., bifurcation) lineages.
- Visualization.
- Performance measures.
- OOP with S4 classes and methods.
- `Shiny` app for interactive web interface.

References

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

S. C. Bendall, K. L. Davis, E. D. Amir, M. D. Tadmor, E. F. Simonds, T. J. Chen, D. K. Shenfeld, G. P. Nolan, and D. Pe'er. Single-cell trajectory detection uncovers progression and regulatory coordination in human B cell development. Cell, 157(3):714–725, 2014.

K. Campbell, C. P. Ponting, and C. Webber. Laplacian eigenmaps and principal curves for high resolution pseudotemporal ordering of single-cell RNA-seq profiles. Technical report, MRC Functional Genomics Unit, University of Oxford, UK, 2015. URL `biorxiv.org/content/early/2015/09/18/027219`.

S. Dudoit and J. Fridlyand. Bagging to improve the accuracy of a clustering procedure. Bioinformatics, 19(9):1090–1099, 2003. URL `http://bioinformatics.oxfordjournals.org/content/19/9/1090.abstract`.

T. Hastie and W. Stuetzle. Principal curves. Journal of the American Statistical Association, 84(406):502–516, 1989.

Z. Ji and H. Ji. TSCAN: Pseudo-time reconstruction and evaluation in single-cell RNA-seq analysis. Nucleic Acids Research, 2016.

References

F. Leisch. Bagged clustering. Technical Report 51, SFB Adaptive Information Systems and Modelling in Economics and Management Science, Vienna University of Economics and Business Administration, Vienna, Austria, August 1999. URL `www.ci.tuwien.ac.at/~leisch/papers/fl-techrep.html`.

S. Petropoulos, D. Edsgärd, B. Reinius, Q. Deng, S. P. Panula, S. Codeluppi, A. Plaza Reyes, S. Linnarsson, R. Sandberg, and F. Lanner. Single-cell RNA-Seq reveals lineage and X chromosome dynamics in human preimplantation embryos. Cell, 165(In press):1–15, 2016.

J. Shin, D. A. Berg, Y. Zhu, J. Y. Shin, J. Song, M. A. Bonaguidi, G. Enikolopov, D. W. Nauen, K. M. Christian, G. Ming, and H. Song. Single-cell RNA-Seq with Waterfall reveals molecular cascades underlying adult neurogenesis. Cell Stem Cell, 17(3):360–372, 2015.

C. Trapnell, D. Cacchiarelli, J. Grimsby, P. Pokharel, S. Li, M. Morse, N. J. Lennon, K. J. Livak, T. S. Mikkelsen, and J. L. Rinn. The dynamics and regulators of cell fate decisions are revealed by pseudotemporal ordering of single cells. Nature Biotechnology, 4(32):381–391, 2014.

References

G. C. Tseng and W. H. Wong. Tight clustering: a resampling-based approach for identifying stable and tight patterns in data. Biometrics, 61(1):10–16, 2005.