Further strategies for analyzing single-cell RNA-seq data

***Aaron T. L. Lun*[1], *Davis J. McCarthy*[2,3] *and John C. Marioni*[1,2,4]**

[1]Cancer Research UK Cambridge Institute, Li Ka Shing Centre, Robinson Way, Cambridge CB2 0RE, United Kingdom
[2]EMBL European Bioinformatics Institute, Wellcome Genome Campus, Hinxton, Cambridge CB10 1SD, United Kingdom
[3]St Vincent's Institute of Medical Research, 41 Victoria Parade, Fitzroy, Victoria 3065, Australia
[4]Wellcome Trust Sanger Institute, Wellcome Genome Campus, Hinxton, Cambridge CB10 1SA, United Kingdom

2019-01-09

1 Overview

Here, we describe a few additional analyses that can be performed with single-cell RNA sequencing data. This includes detection of significant correlations between genes and regressing out the effect of cell cycle from the gene expression matrix.

2 Identifying correlated gene pairs with Spearman’s rho

scRNA-seq data is commonly used to identify correlations between the expression profiles of different genes. This is quantified by computing Spearman’s rho, which accommodates non-linear relationships in the expression values. Non-zero correlations between pairs of genes provide evidence for their co-regulation. However, the noise in the data requires some statistical analysis to determine whether a correlation is significantly non-zero.

To demonstrate, we use the `correlatePairs` function to identify significant correlations between the various histocompatability antigens in the haematopoietic stem cell (HSC) dataset (Wilson et al. 2015). The significance of each correlation is determined using a permutation test. For each pair of genes, the null hypothesis is that the expression profiles of two genes are independent. Shuffling the profiles and recalculating the correlation yields a null distribution that is used to obtain a *p*-value for each observed correlation value (Phipson and Smyth 2010).

```
library(scran)
sce.hsc <- readRDS("hsc_data.rds")

set.seed(100)
var.cor <- correlatePairs(sce.hsc, subset.row=grep("^H2-", r
ownames(sce.hsc)))
head(var.cor)
```

```
## DataFrame with 6 rows and 6 columns
##         gene1       gene2               rho
p.value
##   <character> <character>         <numeric>            <n
umeric>
## 1      H2-Ab1      H2-Eb1 0.497634203104049   1.999998000
002e-06
## 2       H2-Aa      H2-Ab1 0.488479262672811   1.999998000
002e-06
## 3       H2-D1       H2-K1 0.412280566558266  3.3999966000
034e-05
## 4       H2-Aa      H2-Eb1  0.41029237242421  3.7999962000
038e-05
## 5      H2-Ab1     H2-DMb1 0.359662777615092 0.00045599954
4000456
## 6       H2-Q6       H2-Q7 0.339981196923693  0.0009899990
1000099
##                    FDR   limited
##              <numeric> <logical>
## 1 0.000434999565000435      TRUE
## 2 0.000434999565000435      TRUE
## 3  0.00413249586750413     FALSE
## 4  0.00413249586750413     FALSE
## 5   0.0396719603280397     FALSE
## 6   0.0717749282250718     FALSE
```

Correction for multiple testing across many gene pairs is performed by controlling the FDR at 5%.

```
sig.cor <- var.cor$FDR <= 0.05
summary(sig.cor)
```

```
##    Mode   FALSE    TRUE
## logical     430       5
```

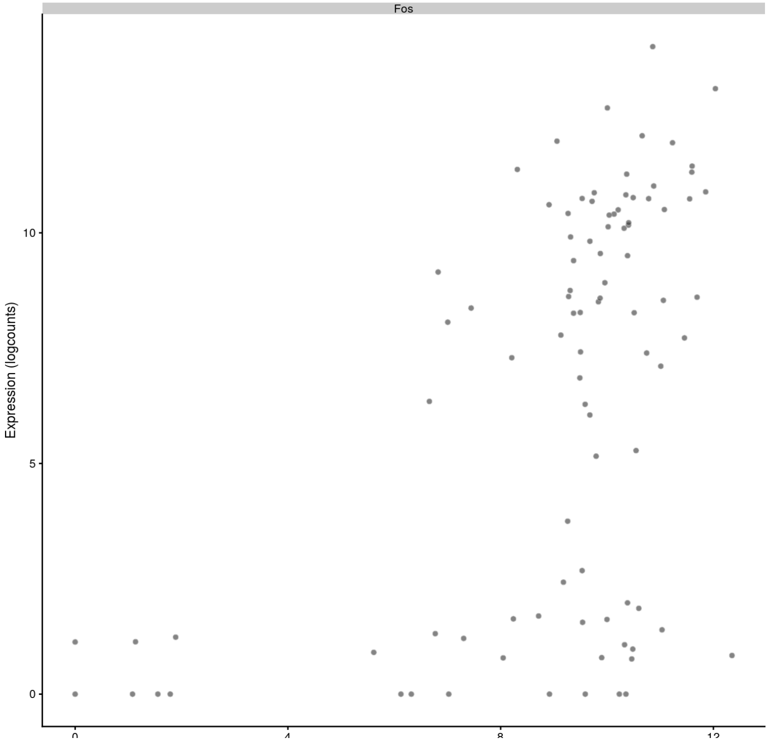
We can also compute correlations between specific pairs of genes, or between all pairs between two distinct sets of genes. The example below computes the correlation between *Fos* and *Jun*, which dimerize to form the AP-1 transcription factor (Angel and Karin 1991).

```
correlatePairs(sce.hsc, subset.row=cbind("Fos", "Jun"))
```

```
## DataFrame with 1 row and 6 columns
##         gene1       gene2               rho            p.
value
##   <character> <character>         <numeric>          <num
eric>
## 1         Fos         Jun 0.466855724920241 1.99999800000
2e-06
##                  FDR   limited
##            <numeric> <logical>
## 1 1.999998000002e-06      TRUE
```

Examination of the expression profiles in Figure 1 confirms the presence of a modest correlation between these two genes.

```
library(scater)
plotExpression(sce.hsc, features="Fos", x="Jun")
```



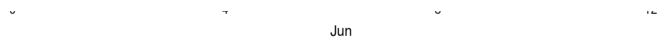


Figure 1: Expression of *Fos* plotted against the expression of *Jun* for all cells in the HSC dataset.

The use of `correlatePairs` is primarily intended to identify correlated gene pairs for validation studies. Obviously, non-zero correlations do not provide evidence for a direct regulatory interaction, let alone specify causality. To construct regulatory networks involving many genes, we suggest using dedicated packages such as *WCGNA (https://CRAN.R-project.org/package=WCGNA)*.

Comments from Aaron:

- We suggest only computing correlations between a subset of genes of interest, known either *a priori* or empirically defined, e.g., as HVGs. Computing correlations across all genes will take too long; unnecessarily increase the severity of the multiple testing correction; and may prioritize strong but uninteresting correlations, e.g., between tightly co-regulated house-keeping genes.
- The `correlatePairs` function can also return gene-centric output by setting `per.gene=TRUE`. This calculates a combined *p*-value (Simes 1986) for each gene that indicates whether it is significantly correlated to any other gene. From a statistical perspective, this is a more natural approach to correcting for multiple testing when genes, rather than pairs of genes, are of interest.
- The `Limited` field indicates whether the *p*-value was lower-bounded by the number of permutations. If this is `TRUE` for any non-significant gene at the chosen FDR threshold, consider increasing the number of permutations to improve power.

3 Blocking on the cell cycle phase

Cell cycle phase is usually uninteresting in studies focusing on other aspects of biology. However, the effects of cell cycle on the expression profile can mask other effects and interfere with the interpretation of the results. This cannot be avoided by simply removing cell cycle marker genes, as the cell cycle can affect a substantial number of other transcripts (Buettner et al. 2015). Rather, more sophisticated strategies are required, one of which is demonstrated below using data from a study of T Helper 2 (T_H2) cells (Mahata et al. 2014).

```
library(BiocFileCache)
bfc <- BiocFileCache("raw_data", ask = FALSE)
mahata.fname <- bfcrpath(bfc,
    "http://www.nature.com/nbt/journal/v33/n2/extref/nbt.310
2-S7.xlsx")
```

Buettner et al. (2015) have already applied quality control and normalized the data, so we can use them directly as log-expression values (accessible as Supplementary Data 1 of https://dx.doi.org/10.1038/nbt.3102 (https://dx.doi.org/10.1038/nbt.3102)).

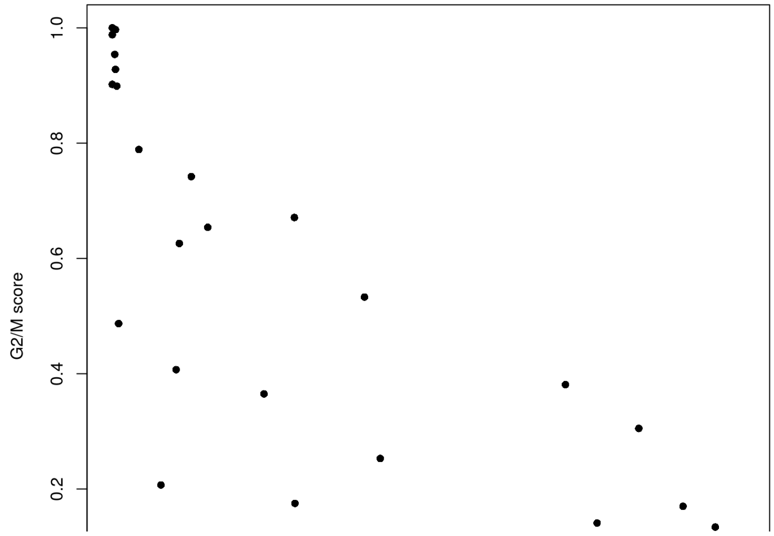
```
library(readxl)
incoming <- as.data.frame(read_excel(mahata.fname, sheet=1))
rownames(incoming) <- incoming[,1]
incoming <- incoming[,-1]
incoming <- incoming[,!duplicated(colnames(incoming))] # Rem
ove duplicated genes.
sce.th2 <- SingleCellExperiment(list(logcounts=t(incoming)))
```

We empirically identify the cell cycle phase using the pair-based classifier in `cyclone`. The majority of cells in Figure 2 seem to lie in G1 phase, with small numbers of cells in the other phases.

```
library(org.Mm.eg.db)
ensembl <- mapIds(org.Mm.eg.db, keys=rownames(sce.th2), keyt
ype="SYMBOL", column="ENSEMBL")

set.seed(100)
mm.pairs <- readRDS(system.file("exdata", "mouse_cycle_marke
rs.rds",
    package="scran"))
assignments <- cyclone(sce.th2, mm.pairs, gene.names=ensemb
l, assay.type="logcounts")

plot(assignments$score$G1, assignments$score$G2M,
    xlab="G1 score", ylab="G2/M score", pch=16)
```



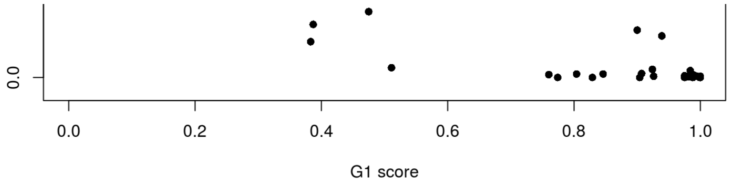


Figure 2: Cell cycle phase scores from applying the pair-based classifier on the T_H2 dataset, where each point represents a cell.

We can block directly on the phase scores in downstream analyses. This is more graduated than using a strict assignment of each cell to a specific phase, as the magnitude of the score considers the uncertainty of the assignment. The phase covariates in the design matrix will absorb any phase-related effects on expression such that they will not affect estimation of the effects of other experimental factors. Users should also ensure that the phase score is not confounded with other factors of interest. For example, model fitting is not possible if all cells in one experimental condition are in one phase, and all cells in another condition are in a different phase.

```
design <- model.matrix(~ G1 + G2M, assignments$score)
fit.block <- trendVar(sce.th2, design=design, parametric=TRU
E, use.spikes=NA)
dec.block <- decomposeVar(sce.th2, fit.block)

library(limma)
sce.th2.block <- sce.th2
assay(sce.th2.block, "corrected") <- removeBatchEffect(
    logcounts(sce.th2), covariates=design[,-1])

sce.th2.block <- denoisePCA(sce.th2.block, technical=dec.blo
ck,
    assay.type="corrected")
dim(reducedDim(sce.th2.block, "PCA"))
```

```
## [1] 81  5
```

The result of blocking on `design` is visualized with some PCA plots in Figure 3. Before removal, the distribution of cells along the first two principal components is strongly associated with their G1 and G2/M scores. This is no longer the case after removal, which suggests that the cell cycle effect has been mitigated.

```
sce.th2$G1score <- sce.th2.block$G1score <- assignments$scor
e$G1
sce.th2$G2Mscore <- sce.th2.block$G2Mscore <- assignments$sc
ore$G2M

# Without blocking on phase score.
fit <- trendVar(sce.th2, parametric=TRUE, use.spikes=NA)
sce.th2 <- denoisePCA(sce.th2, technical=fit$trend)
fontsize <- theme(axis.text=element_text(size=12), axis.titl
e=element_text(size=16))
out <- plotReducedDim(sce.th2, use_dimred="PCA", ncomponents
=2, colour_by="G1score",
    size_by="G2Mscore") + fontsize + ggtitle("Before remova
l")

# After blocking on the phase score.
out2 <- plotReducedDim(sce.th2.block, use_dimred="PCA", ncom
ponents=2,
    colour_by="G1score", size_by="G2Mscore") + fontsize +
    ggtitle("After removal")
multiplot(out, out2, cols=2)
```

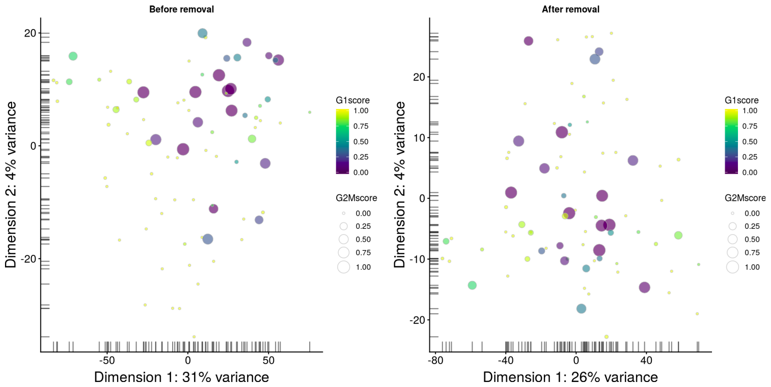


Figure 3: PCA plots before (left) and after (right) removal of the cell cycle effect in the T_H2 dataset. Each cell is represented by a point with colour and size determined by the G1 and G2/M scores, respectively.

As an aside, this dataset contains cells at various stages of differentiation (Mahata et al. 2014). This is an ideal use case for diffusion maps which perform dimensionality reduction along a continuous process. In Figure 4, cells are arranged along a trajectory in the low-dimensional space. The first diffusion component is likely to correspond to T_H2 differentiation, given that a key regulator *Gata3* (Zhu et al. 2006) changes in expression from left to right.

```
plotDiffusionMap(sce.th2.block, colour_by="Gata3",
    run_args=list(use_dimred="PCA", sigma=25)) + fontsize
```

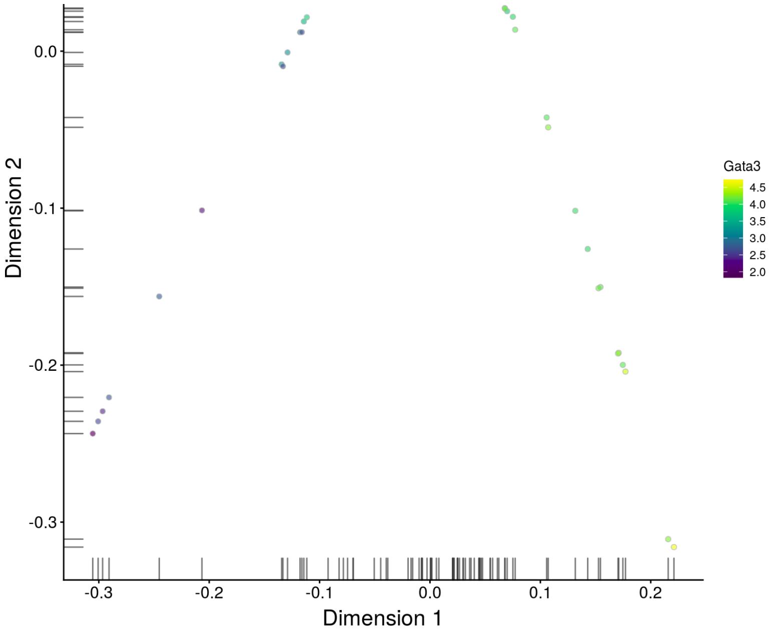


Figure 4: A diffusion map for the T_H2 dataset, where each cell is coloured by its expression of *Gata3*. A larger `sigma` is used compared to the default value to obtain a smoother plot.

4 Concluding remarks

All software packages used in this workflow are publicly available from the Comprehensive R Archive Network (https://cran.r-project.org (https://cran.r-project.org)) or the Bioconductor project (http://bioconductor.org (http://bioconductor.org)). The specific version numbers of the packages used are shown below, along with the version of the R installation.

```
sessionInfo()
```

```
## R version 3.5.2 (2018-12-20)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: Ubuntu 16.04.5 LTS
##
## Matrix products: default
## BLAS: /home/biocbuild/bbs-3.8-bioc/R/lib/libRblas.so
## LAPACK: /home/biocbuild/bbs-3.8-bioc/R/lib/libRlapack.so
##
## locale:
##  [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C
##  [3] LC_TIME=en_US.UTF-8        LC_COLLATE=C
##  [5] LC_MONETARY=en_US.UTF-8    LC_MESSAGES=en_US.UTF-8
##  [7] LC_PAPER=en_US.UTF-8       LC_NAME=C
##  [9] LC_ADDRESS=C               LC_TELEPHONE=C
## [11] LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C
##
## attached base packages:
## [1] parallel  stats4    stats     graphics  grDevices uti
ls     datasets
## [8] methods   base
##
## other attached packages:
##  [1] readxl_1.2.0
##  [2] gdata_2.18.0
##  [3] R.utils_2.7.0
##  [4] R.oo_1.22.0
##  [5] R.methodsS3_1.7.1
##  [6] scRNAseq_1.8.0
##  [7] edgeR_3.24.3
##  [8] Matrix_1.2-15
##  [9] org.Hs.eg.db_3.7.0
## [10] EnsDb.Hsapiens.v86_2.99.0
## [11] ensembldb_2.6.3
## [12] AnnotationFilter_1.6.0
## [13] DropletUtils_1.2.2
## [14] pheatmap_1.0.12
## [15] cluster_2.0.7-1
## [16] dynamicTreeCut_1.63-1
## [17] limma_3.38.3
## [18] scran_1.10.2
## [19] scater_1.10.1
## [20] ggplot2_3.1.0
## [21] TxDb.Mmusculus.UCSC.mm10.ensGene_3.4.0
## [22] GenomicFeatures_1.34.1
## [23] org.Mm.eg.db_3.7.0
## [24] AnnotationDbi_1.44.0
## [25] SingleCellExperiment_1.4.1
## [26] SummarizedExperiment_1.12.0
## [27] DelayedArray_0.8.0
## [28] BiocParallel_1.16.5
## [29] matrixStats_0.54.0
## [30] Biobase_2.42.0
## [31] GenomicRanges_1.34.0
## [32] GenomeInfoDb_1.18.1
## [33] IRanges_2.16.0
## [34] S4Vectors_0.20.1
## [35] BiocGenerics_0.28.0
## [36] bindrcpp_0.2.2
## [37] BiocFileCache_1.6.0
## [38] dbplyr_1.2.2
## [39] knitr_1.21
## [40] BiocStyle_2.10.0
##
## loaded via a namespace (and not attached):
##   [1] tidyselect_0.2.5         RSQLite_2.1.1
##   [3] grid_3.5.2               trimcluster_0.1-2.1
##   [5] Rtsne_0.15               munsell_0.5.0
##   [7] destiny_2.12.0           statmod_1.4.30
##   [9] sROC_0.1-2               withr_2.1.2
##  [11] colorspace_1.3-2         highr_0.7
##  [13] robustbase_0.93-3        vcd_1.4-4
##  [15] VIM_4.7.0                TTR_0.23-4
##  [17] labeling_0.3             GenomeInfoDbData_1.2.0
##  [19] cvTools_0.3.2            bit64_0.9-7
##  [21] rhdf5_2.26.2             xfun_0.4
##  [23] ggthemes_4.0.1           diptest_0.75-7
##  [25] R6_2.3.0                 ggbeeswarm_0.6.0
##  [27] robCompositions_2.0.9    RcppEigen_0.3.3.5.0
##  [29] locfit_1.5-9.1           mvoutlier_2.0.9
##  [31] flexmix_2.3-14           bitops_1.0-6
##  [33] reshape_0.8.8            assertthat_0.2.0
##  [35] scales_1.0.0             nnet_7.3-12
##  [37] beeswarm_0.2.3           gtable_0.2.0
##  [39] rlang_0.3.1              scatterplot3d_0.3-41
##  [41] splines_3.5.2            rtracklayer_1.42.1
##  [43] lazyeval_0.2.1           BiocManager_1.30.4
##  [45] yaml_2.2.0               reshape2_1.4.3
##  [47] abind_1.4-5              tools_3.5.2
##  [49] bookdown_0.9             zCompositions_1.1.2
##  [51] RColorBrewer_1.1-2       proxy_0.4-22
##  [53] Rcpp_1.0.0               plyr_1.8.4
##  [55] progress_1.2.0           zlibbioc_1.28.0
##  [57] purrr_0.2.5              RCurl_1.95-4.11
##  [59] prettyunits_1.0.2        viridis_0.5.1
##  [61] cowplot_0.9.4            zoo_1.8-4
##  [63] haven_2.0.0              magrittr_1.5
##  [65] data.table_1.11.8        openxlsx_4.1.0
##  [67] lmtest_0.9-36            truncnorm_1.0-8
##  [69] mvtnorm_1.0-8            ProtGenerics_1.14.0
##  [71] hms_0.4.2                evaluate_0.12
##  [73] smoother_1.1             XML_3.98-1.16
##  [75] rio_0.5.16               mclust_5.4.2
##  [77] gridExtra_2.3            compiler_3.5.2
##  [79] biomaRt_2.38.0           tibble_2.0.0
##  [81] KernSmooth_2.23-15       crayon_1.3.4
##  [83] htmltools_0.3.6          pcaPP_1.9-73
##  [85] rrcov_1.4-7              DBI_1.0.0
##  [87] MASS_7.3-51.1            fpc_2.1-11.1
##  [89] rappdirs_0.3.1           boot_1.3-20
##  [91] car_3.0-2                 sgeostat_1.0-27
##  [93] bindr_0.1.1               igraph_1.2.2
##  [95] forcats_0.3.0             pkgconfig_2.0.2
##  [97] GenomicAlignments_1.18.1 foreign_0.8-71
##  [99] laeken_0.4.6              sp_1.3-1
## [101] vipor_0.4.5               XVector_0.22.0
## [103] NADA_1.6-1                stringr_1.3.1
## [105] digest_0.6.18             pls_2.7-0
## [107] Biostrings_2.50.2         rmarkdown_1.11
## [109] cellranger_1.1.0          DelayedMatrixStats_1.4.0
## [111] curl_3.2                  kernlab_0.9-27
## [113] gtools_3.8.1              Rsamtools_1.34.0
## [115] modeltools_0.2-22         Rhdf5lib_1.4.2
## [117] carData_3.0-2             BiocNeighbors_1.0.0
## [119] viridisLite_0.3.0         pillar_1.3.1
## [121] lattice_0.20-38           GGally_1.4.0
## [123] httr_1.4.0                DEoptimR_1.0-8
## [125] survival_2.43-3           xts_0.11-2
## [127] glue_1.3.0                zip_1.0.0
## [129] prabclus_2.2-6            bit_1.1-14
## [131] class_7.3-15              stringi_1.2.4
## [133] HDF5Array_1.10.1          blob_1.1.1
## [135] memoise_1.1.0             dplyr_0.7.8
## [137] irlba_2.3.2               e1071_1.7-0
```

References

Angel, P., and M. Karin. 1991. “The role of Jun, Fos and the AP-1 complex in cell-proliferation and transformation.” *Biochim. Biophys. Acta* 1072 (2-3):129–57.

Buettner, F., K. N. Natarajan, F. P. Casale, V. Proserpio, A. Scialdone, F. J. Theis, S. A. Teichmann, J. C. Marioni, and O. Stegle. 2015. “Computational analysis of cell-to-cell heterogeneity in single-cell RNA-sequencing data reveals hidden subpopulations of cells.” *Nat. Biotechnol.* 33 (2):155–60.

Mahata, B., X. Zhang, A. A. Kołodziejczyk, V. Proserpio, L. Haim-Vilmovsky, A. E. Taylor, D. Hebenstreit, et al. 2014. “Single-cell RNA sequencing reveals T helper cells synthesizing steroids de novo to contribute to immune homeostasis.” *Cell Rep.* 7 (4):1130–42.

Phipson, B., and G. K. Smyth. 2010. “Permutation P-values should never be zero: calculating exact P-values when permutations are randomly drawn.” *Stat. Appl. Genet. Mol. Biol.* 9:Article 39.

Simes, R. J. 1986. “An Improved Bonferroni Procedure for Multiple Tests of Significance.” *Biometrika* 73 (3):751–54.

Wilson, N. K., D. G. Kent, F. Buettner, M. Shehata, I. C. Macaulay, F. J. Calero-Nieto, M. Sanchez Castillo, et al. 2015. “Combined single-cell functional and gene expression analysis resolves heterogeneity within stem cell populations." *Cell Stem Cell* 16 (6):712–24.

Zhu, J., H. Yamane, J. Cote-Sierra, L. Guo, and W. E. Paul. 2006. "GATA-3 promotes Th2 responses through three different mechanisms: induction of Th2 cytokine production, selective growth of Th2 cells and inhibition of Th1 cell-specific factors." *Cell Res.* 16 (1):3–10.