Code

Laboratory Report - Investigating the role of Nanog during peri-implatation of mouse embryo using scRNA-seq expression analysis

Salvatore Ivan Puglisi (B125042)

21/05/2018 - 31/07/2018

Background

The pluripotent state of mouse embryonic stem cells (ESCs) is commonly associated with high levels of certain transcription factors such as Nanog, Sox2 and Oct4, which are important for establishing and maintaining the pluripotent state. The influence of specific ratios of Oct4 and Nanog in individual ESCs on the pluripotency establishment, has already been observed in-vitro (Muñoz Descalzo Silvia et al. 2012), suggesting the possibility of using the distribution of this ratio as a quantifiers to distinguish between three subpopulations in a ESCs culture: pluripotent, lineage-primed and differentiating cells. Nanog is a divergent homeodomain protein found in mammalian pluripotent cells (Chambers et al. 2003), its presence is considered and hallmark of pluripotent cells in vivo and in-vitro, and its loss an early marker of differentiation (Chambers et al. 2007). Nanog is a core element of the pluripotency gene regulatory network (GRN) (Boyer et al. 2005) and previous studies (Ptashne and Gann 2001) suggests a combinatorial control of transcription factors to taking place in proximity of their target genes. What is still unclear is the magnitude of influence upon the GRN targets delivered by the expression variation of a single factor. A culture of mouse ESCs consists in a mixture of cells with different levels of Nanog expression, because individual mouse ESCs express different levels of Nanog. Individual Nanog low cells, if grown in culture, can reproduce the whole population of cells expressing different levels of Nanog. ESCs expressing Nanog at lower levels are more likely to differentiate than cells expressing high Nanog levels and so the Nanog low state may be considered a marker of early stage in differentiation.
ESCs knockout for Nanog gene can be produced and maintained in culture. By inducing the expression of a Nanog transgene is possible to revert back the Nanog expression of the WT ESCs. The inability of Nanog -/-cells to complete transcription-factor-based reprogramming mirrors the phenotype observed in Nanog null embryos, providing a model to study the unique role of Nanog during the acquisition of pluripotency in early development.

In a previous study (Festuccia et al. 2012) the effects of altering the expression of Nanog has been analyzed, generating in-vitro ESCs knockout for Nanog. This ESCs Nanog -/- model could provide a way to study the unique role of Nanog during the acquisition of pluripotency in early development. More than 5,000 genes were confirmed to bind Nanog using ChIP-seq experiments, but only a small subset of 64 genes showed a > 1.5 fold change in increasing or decreasing expression, after reinduction of Nanog activity in the ESCs Nanog -/-. Among this 64 Nanog identified target genes, ESRRB shows the strongest transcriptional induction and has been proven to substitute for Nanog function in pluripotent cells. Furthermore, the findings that ESRRB is a direct target of Nanog, together with the notion that ESRRB can positively regulate Nanog, demonstrate the existence of a positive feedback mechanism (Oliveri, Tu, and Davidson 2008). Similar observations suggest that the presence of Nanog is necessary but not sufficient to alter the transcriptional rates of it's target genes and confirm the needing of multiple additional pluripotency transcription factors to bind the same targets. Apparently some pluripotency factors like Oct4 are essential to maintain the pluripotent state (Oct4 depletion leads to differentiation (Hall et al. 2009)) while fluctuations in Nanog (and consequently ESRRB) confer flexibility to the GRN, tuning the expression of downstream genes and leading to cell fate decisions (Chambers et al. 2007). The pre-implantation mouse embryo at day E3.5 consists of an inner cell mass (ICM) not possessing distinct lineage identities. Networks of genes including several known pluripotency markers are observed exclusively at this stage. Implantation occurs at approximately embryonic day E4.5 and marks several key changes in the embryo. The embryonic epiblast is formed, combined with two extra-embryonic layers: the trophectoderm and primitive endoderm (PrE). the pluripotent epiblast dynamically

changes post-implantation, developing into a transcriptionally distinct entity primed for differentiation (Mohammed et al. 2017). ESCs are derived from the embryos at stage E3.5 of differentiation and might be expected to show similar expression profile for Nanog. Embryonic cells at stage E4.5 still express Nanog but at a lower average level. Nanog expression is indeed able to affects the ability of cells to differentiate in-vitro but is unknown how the same genes responding to Nanog observed in-vitro, correlate with Nanog in the embryo at stages E3.5 and E4.5 of differentiation. In this study the following question will be addressed, using single cell RNA expression data.

- Is it possible to clarify how the 64 genes responding to Nanog identified using the ESCs Nanog -/- model (Festuccia et al. 2012) behave in the embryo?
- Are these genes unregulated or regulated in the same way observed in the ESCs in-vitro models?
- Is it possible to measure how the Nanog expression is distributed across individual cells? Does it fluctuates naturally as expected in the real mouse embryo?

Performing a correlation analysis between the Nanog expression profile and the correspondent fluctuation of its target genes in the mouse embryo, will be possible to compare the behaviour of these gene between the ESCs, pre-implantation (E3.5) and post-implantation cells (E4.5).

Purpose

The aim of this study is to investigate if the Nanog target genes proven to bind and respond to Nanog in the ESCs in-vitro model, are correlated with Nanog expression fluctuation in single mESC cells profiled using single cell RNA-seq (scRNA-seq). We would then like to know how these genes are correlated to Nanog expression during two stages of the early mouse embryo development: the pre-implantation inner cell mass at E3.5 and the post-implantation epiblast at E4.5.

Analysis

An outline of the analytic methods used during my intership is reported below. More detailed R scripts and higher resolution output graphics can be found into the “analysis” folder, inside the specific subdirectories “all”, “e35”, “e45”. It is also possible to load the R Workspace Files and run the script inside the R markdown notebook changing some parameters.

Dataset Description

The data analyzed in this project are originated from

> Mohammed, H., Hernando-Herraez, I., Savino, A., Scialdone, A., Macaulay, I., Mulas, C., Chandra, T., Voet, T., Dean, W., Nichols, J., et al. (2017). Single-Cell Landscape of Transcriptional Heterogeneity and Cell Fate Decisions during Mouse Early Gastrulation. Cell Rep. 20, 1215-1228.

The authors isolated Single cells from C57Bl/6Babr Mus musculus embryos at E3.5, E4.5, E5.5 and E6.5 stages, subjecting them to single-cell RNA-seq protocol, using the platform Illumina HiSeq 2500 (generating 100 bp paired-end reads).

The references for the online repositories are reported below:

BioProject ID: PRJNA392258 (https://www.ncbi.nlm.nih.gov/bioproject/PRJNA392258)

GEO ID: GSE100597 (https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE100597)

SRA ID: SRP110669 (https://trace.ncbi.nlm.nih.gov/Traces/sra/sra.cgi?study=SRP110669)

Only E3.5 & E4.5 are considered in this study, to explore lineage and regulatory processes involved in early peri-implantation mouse embryos.

Total SRA Experiments: 721 samples (SRR5763563 - SRR5764283)

E3.5 subset: 99 samples (SRR5763563 - SRR5763661)

E4.5 subset: 105 samples (SRR5763662 - SRR5763766)

Basic Statistics

Measure	Value
Filename	SRR5763563_1.fastq
File type	Conventional base calls
Encoding	Sanger / Illumina 1.9
Total Sequences	799689
Sequences flagged as poor quality	0
Sequence length	100
%GC	51

Fig. 1 - Example summary generated by FastQC for one sample.

Quality control & preprocessing of reads

The analysis has been carried out via SSH on the following server

> bioinfmsc3.mvm.ed.ac.uk"

The UNIX environment has been prepared, upgrading R to the release 3.5.0 as a requirement for installing the [“scater”] (D. J. McCarthy et al. 2017) and other [Bioconductor] (Huber et al. 2015) packages (A. Lun 2018), (Ritchie et al. 2015).

```
source("https://bioconductor.org/biocLite.R")
biocLite(c("scater","scran","limma"))
```

Other command-line tools required for the data preprocessing and the genomic allignment have been installed via Bioconda (https://bioconda.github.io/) (Grüning et al. 2018).

```
wget https://repo.anaconda.com/archive/Anaconda3-5.2.0-Linux-x86_64.sh
sh Anaconda3-5.2.0-Linux-x86_64.sh # added Anaconda/bin to PATH
conda install -c bioconda sra-tools fastqc multiqc trim-galore kallisto
```

SRA Toolkit (https://www.ncbi.nlm.nih.gov/books/NBK158900/) is a collection of tools and libraries for using data in the INSDC Sequence Read Archives. FastQC (https://www.bioinformatics.babraham.ac.uk/projects/fastqc/) (Andrews 2018) and MultiQC (http://multiqc.info/) (Ewels et al. 2016) are the quality control tools which have been used to assess the overall quality of the reads. Trim Galore! (https://www.bioinformatics.babraham.ac.uk/projects/trim_galore/) (Krueger 2018) is a wrapper of cutadapt and Fastqc, used to filter-out the low quality reads and trimming the adapter contamination detected during the quality control. kallisto (https://pachterlab.github.io/kallisto/) (Bray et al. 2016) is a pseudoalignment program for rapidly determining the compatibility of high-throughput sequencing reads with targets sequences, without the need for alignment. It has been used for quantifying abundances of transcripts in the Mus musculus transcriptome from the scRNA-Seq data.

The main “project” folder has been created on my home directory, with the following subfolders:

- data -> containing the fastq files downloaded and processed from the SRA repositories
- references -> containing the M. musculus transcriptome and annotation files dowloaded from ENSEMBL repositories
- indices -> containing the index file generated by kallisto, to be used for the preudoalignment
- results -> containing the abundances files generated by kallisto quant
- analysis -> containing the RData files and the output of the Bioconductor analysis

The experiment “SRP110669” has been downloaded from SRA repository inside the “data” folder and the sample files converted to compressed FastQ files by the fastq-dump utility (part of SRA Toolkit).

```
wget -r -N -nd ftp://ftp-trace.ncbi.nih.gov/sra/sra-instant/reads/ByStudy/sra/SRP/SRP110/SRP11066
9/

# using the option --split-files forward and reverse reads in every SRA file are separated in 2 F
astq files

fastq-dump --split-files --gzip --sra-id *.sra

# Run FastQC for all samples

$ fastqc *.fastq.gz
```

Looking at the HTML reports generated by FastQC the quality appears to be on average over 25 phred score along the lenght of each read.

Fig. 2 - MultiQC Sequence Quality Histograms. The mean quality value across each base position in the read for all samples.

Anyway it is possible to observe that 5’ end of the reads in any sample is affected by base content similarity, due to use of random k-mers (which are not so random) or transposases (nextera transposases, causing tagmentation) in library preparation.

Per base sequence content

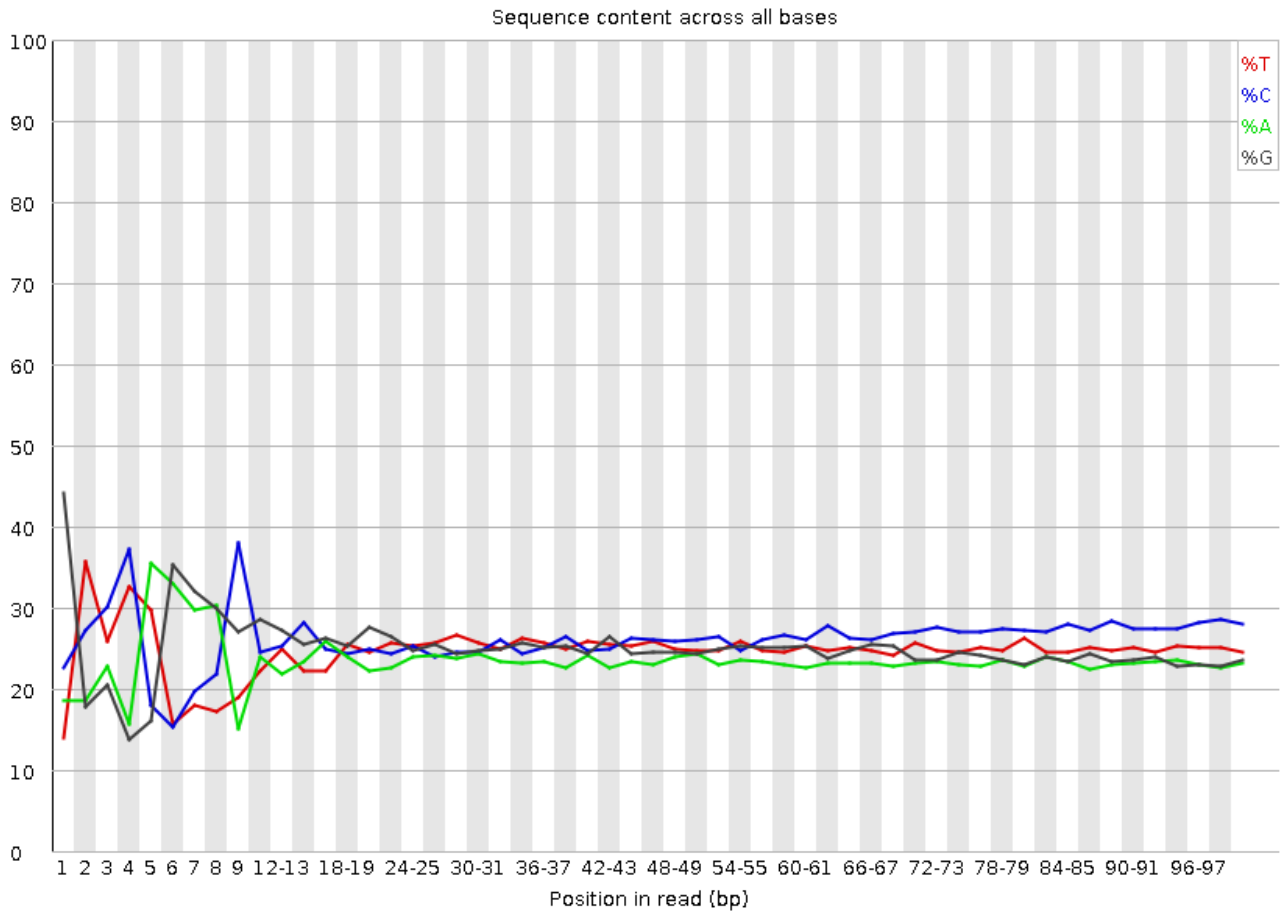


Fig. 3 - Per Base Sequence Content Plot from FastQC for one sample, showing the relative amount of each base at each position.

The first run of FASTQC detected adapter contamination by nextera transposase.

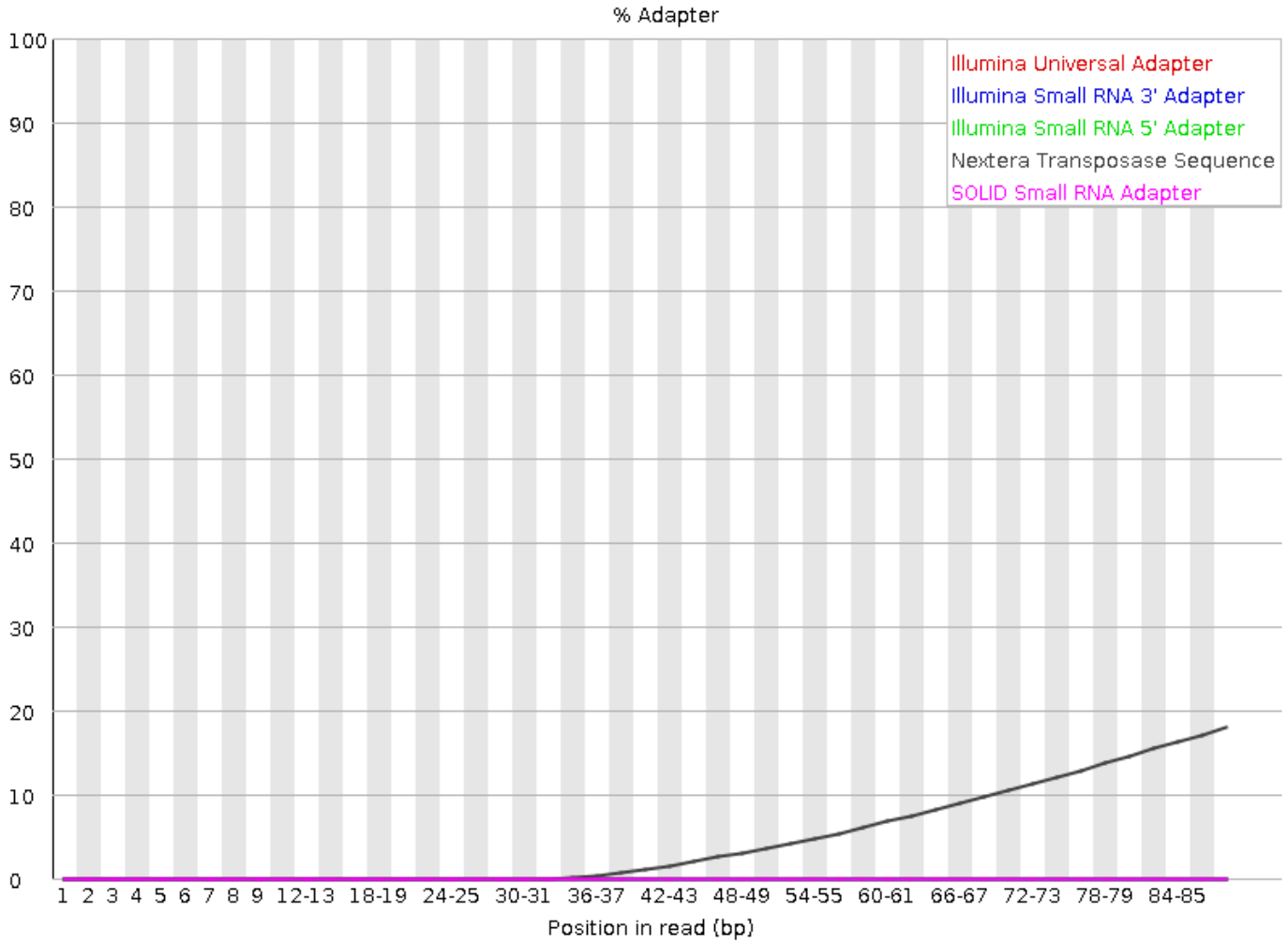


Fig. 4 - FastQC Adapter Content plot, showing a cumulative percentage count of the proportion of library which has seen each of the adapter sequences at each position

The non-random per base sequence content caused by random k-mers cannot be corrected but should not affect the analysis. The removal of the adapter contamination is possible.

Using Trim Galore! has been performed a quality trimming/filtering (phred score >20) and sequence trimming (for nextera trasposase).

First I prepared a tab-separated list of the fastq paired files

fastq_gz_trim_list.txt

```
SRR5763563_1.fastq.gz   SRR5763563_2.fastq.gz
SRR5763564_1.fastq.gz   SRR5763564_2.fastq.gz
SRR5763565_1.fastq.gz   SRR5763565_2.fastq.gz
SRR5763566_1.fastq.gz   SRR5763566_2.fastq.gz
..............      ..............
```

and script to create a parallel job list that runs trim-galore!

Make_job_list_trim.sh

```
#!/bin/bash

[ $# -ne 2 ] && { echo -en "\n*** This script generates jobs for GNU parallel. *** \n\n Error Not
hing to do, usage: < input tab delimited list > < output run list file >\n\n" ; exit 1; }
set -o pipefail

# Get command-line args
INPUT_LIST=$1
OUTPUT=$2

# Set counter
COUNT=1
END=$(wc -l $INPUT_LIST | awk '{print $1}')

echo " "
echo " * Input file is: $INPUT_LIST"
echo " * Number of runs: $END"
echo " * Output job list for GNU parallel saved to: $OUTPUT"
echo " "

# Main bit of command-line for job

CMD="trim_galore --nextera --paired"

# In alternative, to generate also BAM files
#CMD="kallisto quant -t 30 -b 30 --bias --pseudobam --genomebam --gtf ~/project/annotation/Mus_mu
sculus.GRCm38.92.gtf --chromosomes ~/project/reference/chr_len.txt -i ~/project/indices/kallist$

# Main Loop
[ -e $OUTPUT ] && rm $OUTPUT
while [ $COUNT -le $END ];
do
    LINE=( $(awk "NR==$COUNT" $INPUT_LIST) )
    # Make file list
    echo "Working on $COUNT of $END, Files ${LINE[0]} ${LINE[1]}"
    echo "$CMD ${LINE[0]} ${LINE[1]} " >> $OUTPUT
    ((COUNT++))
done
```

```
# run script to create job list
sh Make_job_list_trim.sh fastq_gz_trim_list.txt trim_jobs_temp.txt

# converting carriage return into new line
tr '\r' '\n' < trim_jobs_temp.txt > trim_jobs.txt | rm trim_jobs_temp.txt

# run parallel on 8 threads (saving log)
parallel --progress --jobs 8 --joblog trim_joblog.txt < trim_jobs.txt

# rename generated trimmed files
rename "_val_1" "" *.fq.gz
rename "_val_2" "" *.fq.gz
PRE-TRIM vs POST-TRIM MultiQC Comparison
```

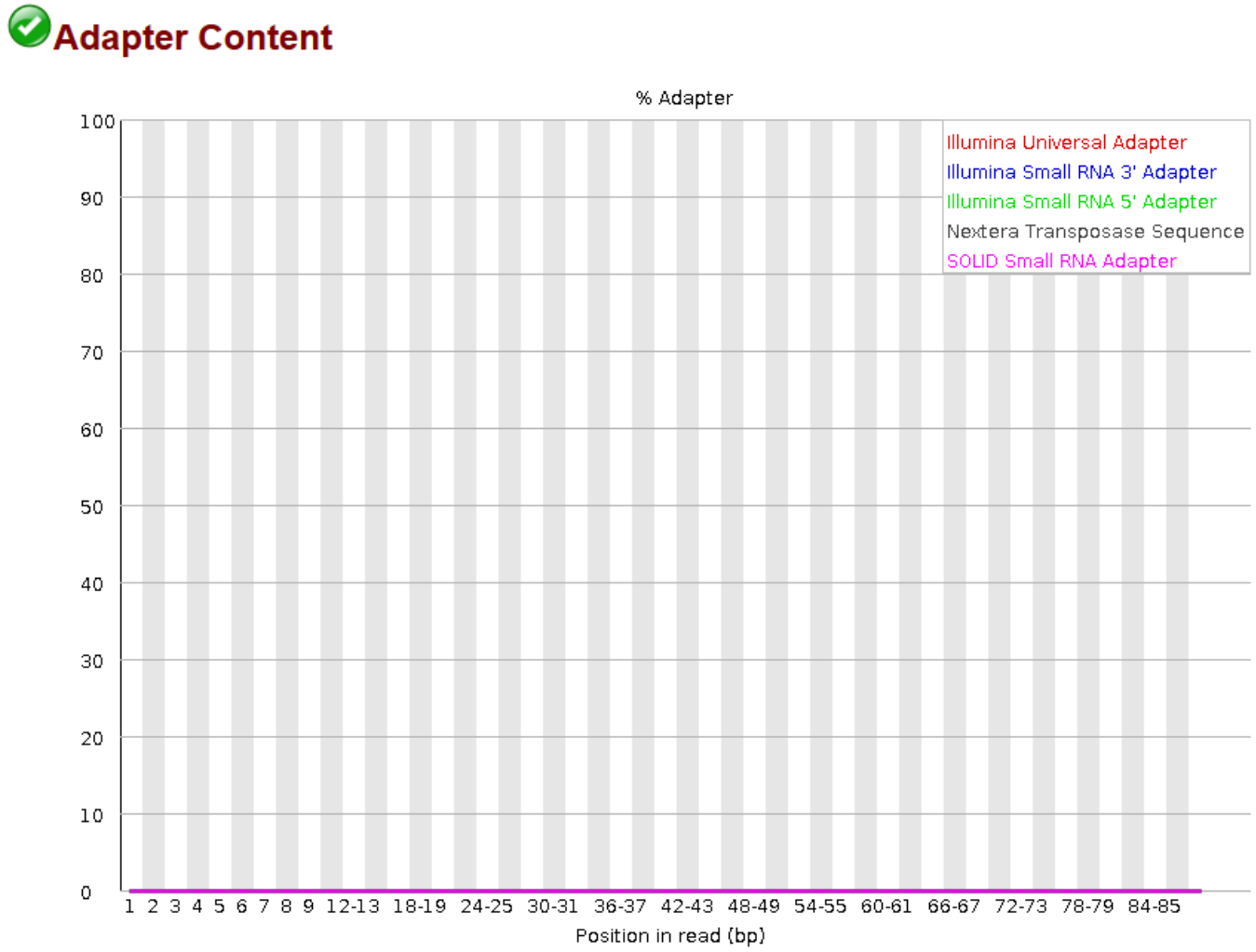


Fig.5 - FastQC Adapter Content plot after adapter trimming and quality filtering.

Nextera transposase residual sequences have been removed removed but a slight heterogeneity in sequence length distribution has been generated across the samples.

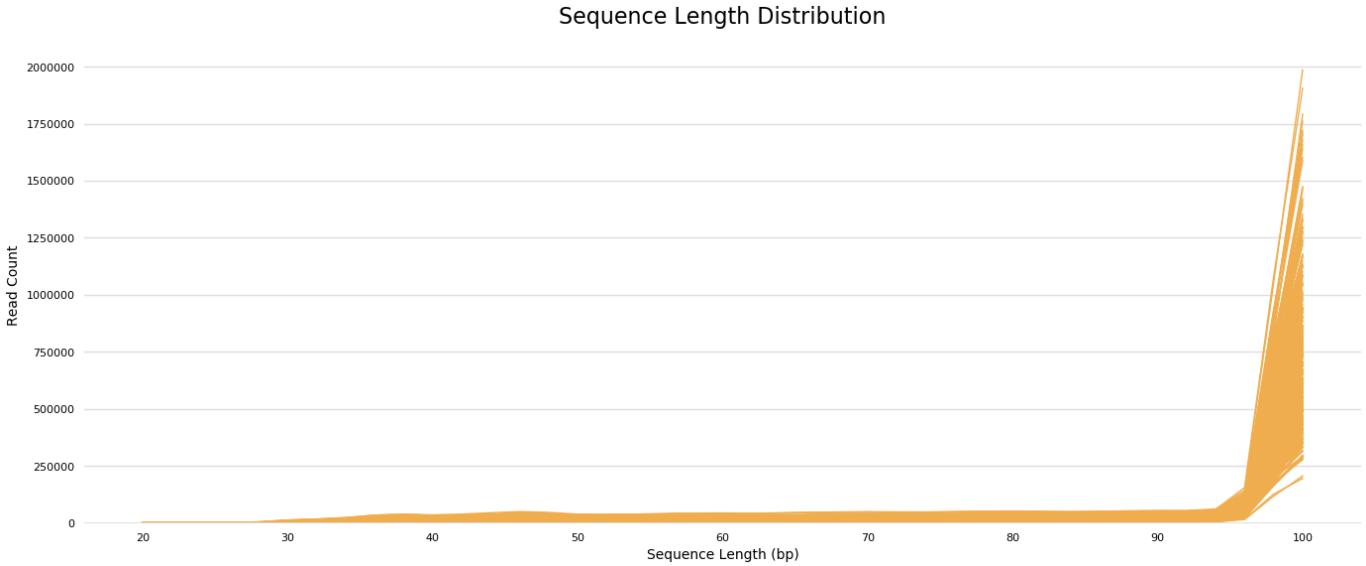


Fig. 6 - Sequence Length Distribution plot, showing the distribution of fragment sizes (read lengths) found.

Reads alignment to transcriptome (Kallisto)

Befor to perform the pseudo-alignment against the mouse transcriptome, kallisto requires to create an index from a FASTA formatted file of target sequences

M. musculus transcriptome (cDNA) has been dowloaded from the Ensembl FTP repository

```
wget ftp://ftp.ensembl.org/pub/release-92/fasta/mus_musculus/cdna/Mus_musculus.GRCm38.cdna.all.fa.gz /reference
```

Using this file a kallisto index has been generated with a default k-mer size of 31.

```
kallisto index  -i indices/kallisto/cdna.kidx reference/Mus_musculus.GRCm38.cdna.all.fa.gz
```

```
[build] loading fasta file reference/transcripts/Mus_musculus.GRCm38.cdna.all.fa
[build] k-mer length: 31
[build] warning: clipped off poly-A tail (longer than 10) from 589 target sequences
[build] warning: replaced 3 non-ACGUT characters in the input sequence with pseudorandom nucleotides
[build] counting k-mers ... done.
[build] building target de Bruijn graph ...  done
[build] creating equivalence classes ...  done
[build] target de Bruijn graph has 691348 contigs and contains 97656136 k-mers
```

In order to produce the quantification, Kallisto has to be run for every paired-end sample, generating the “abundance” files to be imported in R and evaluate the differential expression.

kallisto can run on multiple instances using GNU parallel, in order to speed-up the process. To do so I prepared a tab-separated list of the samples

```
SRR5763563    SRR5763563_1.fq    SRR5763563_2.fq
SRR5763564    SRR5763564_1.fq    SRR5763564_2.fq
SRR5763565    SRR5763565_1.fq    SRR5763565_2.fq
SRR5763566    SRR5763566_1.fq    SRR5763566_2.fq
SRR5763567    SRR5763567_1.fq    SRR5763567_2.fq
.........       ..........          ..........
```

And a bash script to ingest this list and generate the jobs for parallel…

> Make_job_list_kallisto.sh

```
#!/bin/bash

[ $# -ne 2 ] && { echo -en "\n*** This script generates jobs for GNU parallel. *** \n\n Error Not
hing to do, usage: < input tab delimited list > < output run list file >\n\n" ; exit 1; }
set -o pipefail

# Get command-line args
INPUT_LIST=$1
OUTPUT=$2

# Set counter
COUNT=1
END=$(wc -l $INPUT_LIST | awk '{print $1}')

echo " "
echo " * Input file is: $INPUT_LIST"
echo " * Number of runs: $END"
echo " * Output job list for GNU parallel saved to: $OUTPUT"
echo " "

# Main bit of command-line for job

CMD="kallisto quant -t 30 -b 30 --bias -i ~/project/indices/kallisto/cdna.kidx"

# In alternative, to generate also BAM files
#CMD="kallisto quant -t 30 -b 30 --bias --pseudobam --genomebam --gtf ~/project/annotation/Mus_mu
sculus.GRCm38.92.gtf --chromosomes ~/project/reference/chr_len.txt -i ~/project/indices/kallisto/
cdna.kidx"

# Main Loop
[ -e $OUTPUT ] && rm $OUTPUT
while [ $COUNT -le $END ];
do
    LINE=( $(awk "NR==$COUNT" $INPUT_LIST) )
    # Make file list
    echo "Working on $COUNT of $END Sample ID: ${LINE[0]}, Files ${LINE[1]} ${LINE[2]}"
    echo "$CMD -o ~/project/results/kallisto/${LINE[0]} ${LINE[1]} ${LINE[2]} " >> $OUTPUT
    ((COUNT++))
done
```

Other than index file and the FastQ files, additional options can be specified for Kallisto: - –bias - learns parameters for a model of sequences specific bias and corrects the abundances accordlingly. - –b - number of bootstrap for estimating the technical variance in samples (all bootstrap are compressed inside abundance.h5 output) - –pseudobam - save pseudoalignments to transcriptome to BAM file - –genomebam - Project pseudoalignments to genome sorted BAM file (used in combination with -g to specify the annotation GTF and -c to give a list of chromosome lenghts)

```
#Generating the job list
sh Make_job_list_kallisto.sh fastq_gz_kal_list.txt kallisto_jobs_temp.txt

# converting carriage line to newline
tr '\r' '\n' < kallisto_jobs_temp.txt > kallisto_jobs.txt | rm kallisto_jobs_temp.txt

# running parallel for 8 threads (saving log)
nohup parallel --progress --jobs 8 --joblog kallisto_joblog.txt < kallisto_jobs.txt &
```

Outputs of kallisto are in folder “project/results/kallisto/”.

Bioconductor analysis

The same analysis pipeline has been performed for the abundance files referring to the stages E3.5 and E4.5 following the guidlines published here:

> Lun, A.T.L., McCarthy, D.J., and Marioni, J.C. (2016). A step-by-step workflow for low-level analysis of single-cell RNA-seq data. F1000Research 5, 2122.

All the R files and outputs are insite the “analysis” organized in subfolders depending on which samples the analysis has been performed: - all -> contains the R output of the whole dataset analysis - e35 -> contains the R output of the stage E3.5 analysis - e45 -> contains the R output of the stage E4.5 analysis - report -> contains the R output and the image files of this report

The exemple code used for the stage E4.5 is reported below.

Set working directory

Hide

```
setwd("~/project/analysis/e45")
```

```
The working directory was changed to C:/Users/Ivan/Documents/project/analysis/e45 inside a notebo
ok chunk. The working directory will be reset when the chunk is finished running. Use the knitr r
oot.dir option in the setup chunk to change the working directory for notebook chunks.
```

Data praparation

Loading required packages

Preparing E4.5 samples files

Hide

```
metadata <- read.delim("samples_info.txt", check.names=FALSE, header=TRUE)
samples <- as.character(metadata$sample[metadata$stage == "E4.5"])
kal_dir <- "../../results/kallisto"
files <- file.path(kal_dir, samples, "abundance.tsv")
```

Creating tx2gene to convert transcript_ids into gene_names in kallisto results

Creating singleCellExperiment from Kallisto results using Tximport, collapsing the rows by genes (txOut = FALSE)

Hide

```
sce <- readTxResults(samples = samples, files = files, type = "kallisto", tx2gene = tx2gene, igno
reTxVersion = TRUE, txOut = FALSE)
```

```
Kallisto log not provided - assuming all runs successful
```

```
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 3
6 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68
69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100
101 102 103 104 105
transcripts missing from tx2gene: 26755
summarizing abundance
summarizing counts
summarizing length
```

```
Using log2(TPM + logExprsOffset) as 'exprs' values in output.
```

Query ensembldb using the Ensembl transcript ID (rownames) to retriver the annotation info to store into sce By default BiomaRt uses the “mmusculus_gene_ensembl” dataset, retrieving “ensembl_transcript_id”, “ensembl_gene_id”, “mgi_symbol”, “chromosome_name”, “transcript_biotype”, “transcript_start”, “transcript_end”

Rename rownames with gene symbols

Hide

```
rowData(sce)$ensembl_gene_id <- rownames(sce)
new.names <- rowData(sce)$mgi_symbol
missing.name <- is.na(new.names)
new.names[missing.name] <- rowData(sce)$ensembl_gene_id[missing.name]
dup.name <- new.names %in% new.names[duplicated(new.names)]
new.names[dup.name] <- paste0(new.names, "_", rowData(sce)$ensembl_gene_id)[dup.name]
rownames(sce) <- new.names
head(rownames(sce))
```

```
[1] "Gnai3" "Pbsn"  "Cdc45" "Scml2" "Apoh"  "Narf"
```

Define references for control_features to be used in calculateQCMetrics()

Hide

```
mito <- which(rowData(sce)$chromosome_name=="MT")
```

Calculate cell metrics

Hide

```
sce <- calculateQCMetrics(sce, feature_controls=list(Mt=mito))
```

```
Note that the names of some metrics have changed, see 'Renamed metrics' in ?calculateQCMetrics.
Old names are currently maintained for back-compatibility, but may be removed in future releases.
```

Hide

```
names(colData(sce))
```

```
 [1] "n_features"                                    "n_bootstraps"

 [3] "is_cell_control"                               "total_features_by_counts"

 [5] "log10_total_features_by_counts"                "total_counts"

 [7] "log10_total_counts"                            "pct_counts_in_top_50_features"

 [9] "pct_counts_in_top_100_features"                "pct_counts_in_top_200_features"

[11] "pct_counts_in_top_500_features"                "total_features"

[13] "log10_total_features"                          "pct_counts_top_50_features"

[15] "pct_counts_top_100_features"                   "pct_counts_top_200_features"

[17] "pct_counts_top_500_features"                   "total_features_by_counts_endogenous"

[19] "log10_total_features_by_counts_endogenous"     "total_counts_endogenous"

[21] "log10_total_counts_endogenous"                 "pct_counts_endogenous"

[23] "pct_counts_in_top_50_features_endogenous"      "pct_counts_in_top_100_features_endogenous"

[25] "pct_counts_in_top_200_features_endogenous"     "pct_counts_in_top_500_features_endogenous"

[27] "total_features_endogenous"                     "log10_total_features_endogenous"

[29] "pct_counts_top_50_features_endogenous"         "pct_counts_top_100_features_endogenous"

[31] "pct_counts_top_200_features_endogenous"        "pct_counts_top_500_features_endogenous"

[33] "total_features_by_counts_feature_control"      "log10_total_features_by_counts_feature_con
trol"
[35] "total_counts_feature_control"                  "log10_total_counts_feature_control"

[37] "pct_counts_feature_control"                    "total_features_feature_control"

[39] "log10_total_features_feature_control"          "total_features_by_counts_Mt"

[41] "log10_total_features_by_counts_Mt"             "total_counts_Mt"

[43] "log10_total_counts_Mt"                         "pct_counts_Mt"

[45] "total_features_Mt"                             "log10_total_features_Mt"
```

Cell-based QC

First let's have an overview of the features counts across the cells

Hide

```
par(mfrow=c(2,2))
hist(sce$total_counts/1e6, xlab="Library sizes (millions)", main="",
     breaks=20, col="grey80", ylab="Number of cells")
hist(sce$total_features, xlab="Number of expressed genes", main="",
     breaks=20, col="grey80", ylab="Number of cells")
```

Hide

```
plot(sce$total_features, sce$total_counts/1e6, xlab="Number of expressed genes", ylab="Library si
ze (millions)")
plot(sce$total_features, sce$pct_counts_Mt, xlab="Number of expressed genes", ylab="Mitochondrial
 proportion (%)")
```

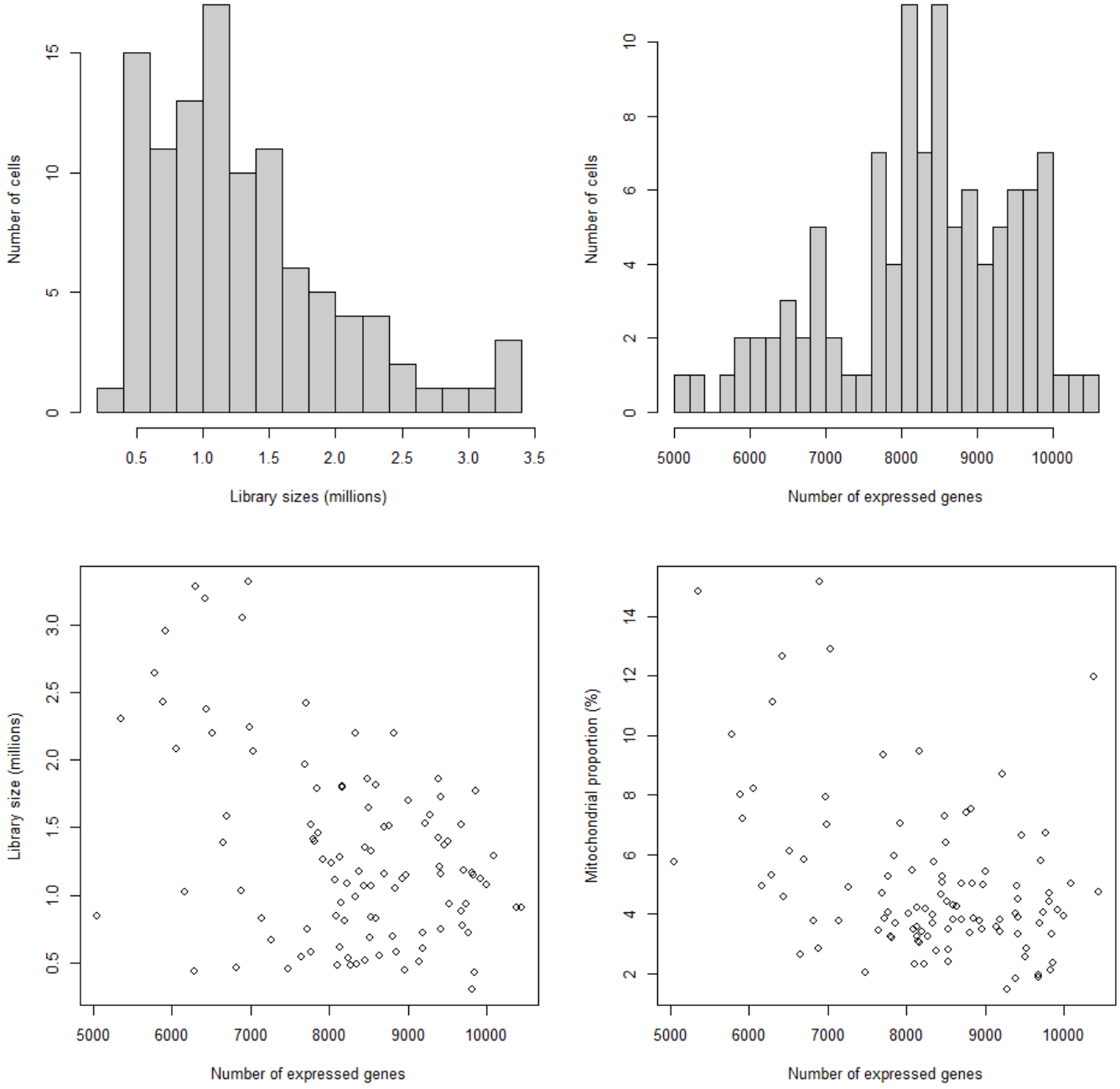


Fig. 7 - Behaviour of each QC metric compared to the total number of expressed features. Each point represents a cell in the E4.5 dataset.

Low-quality cells need to be removed to ensure that technical effects do not distort downstream analysis results. Two common measures of cell quality are the library size and the number of expressed features in each library. The library size is defined as the total sum of counts across all features. Cells with relatively small library sizes are considered to be of low quality as the RNA has not been efficiently captured (i.e., converted into cDNA and amplified) during library preparation. The number of expressed features in each cell is defined as the number of features with non-zero counts for that cell. Any cell with very few expressed genes is likely to be of poor quality as the diverse transcript population has not been successfully captured.

Identifying outliers for each metrics

Outliers are defined based on the median absolute deviation (MADs) from the median value of each metric across all cells. We remove cells with log-library sizes that are more than 3 MADs below the median log-library size & cells where the log-transformed number of expressed genes is 3 MADs below the median value. Another measure of quality is the proportion of reads mapped to genes in the mitochondrial genome. High proportions are indicative of poor-quality cells ((Ilicic et al. 2016), (Islam et al. 2014)), possibly because of increased apoptosis and/or loss of cytoplasmic RNA from lysed cells.

Hide

```
libsize.drop <- isOutlier(sce$total_counts, nmads=3, type="lower", log=TRUE)
feature.drop <- isOutlier(sce$total_features, nmads=3, type="lower", log=TRUE)
mito.drop <- isOutlier(sce$pct_counts_Mt, nmads=3, type="higher")
```

Subsetting by column will retain only the high-quality cells that pass each filter described. Let's see the number of cells removed by each filter as well as the total number of retained cells.

Hide

```
keep <- !(libsize.drop | feature.drop | mito.drop)
data.frame(ByLibSize=sum(libsize.drop), ByFeature=sum(feature.drop), ByMito=sum(mito.drop), Remai
ning=sum(keep))
```

ByLibSize <int>	ByFeature <int>	ByMito <int>	Remaining <int>
0	2	10	94

1 row

Now subset the SingleCellExperiment object to retain only the putative high-quality cells.

Hide

```
sce <- sce[,!(libsize.drop | feature.drop | mito.drop)]
```

Classification of cell-cycle phase

Using a method reported in literature (Scialdone et al. 2015) is possible to classify cells into cell cycle phases based on the gene expression data. Using a training dataset, the sign of the difference in expression between two genes was computed for each pair of genes. Pairs with changes in the sign across cell cycle phases were chosen as markers. Cells in a test dataset can then be classified into the appropriate phase, based on whether the observed sign for each marker pair is consistent with one phase or another. The cyclone package contains a pre-trained set of marker pairs for mouse data, which can be loaded in the the readRDS function. We use the Ensembl identifiers for each gene in our dataset to match up with the names in the pre-trained set of gene pairs.

Hide

```
library(scran)
set.seed(100)
mm.pairs <- readRDS(system.file("exdata", "mouse_cycle_markers.rds", package="scran"))
assignments <- cyclone(sce, mm.pairs, gene.names=rowData(sce)$ensembl_gene_id)
plot(assignments$score$G1, assignments$score$G2M, xlab="G1 score", ylab="G2/M score", pch=16)
```

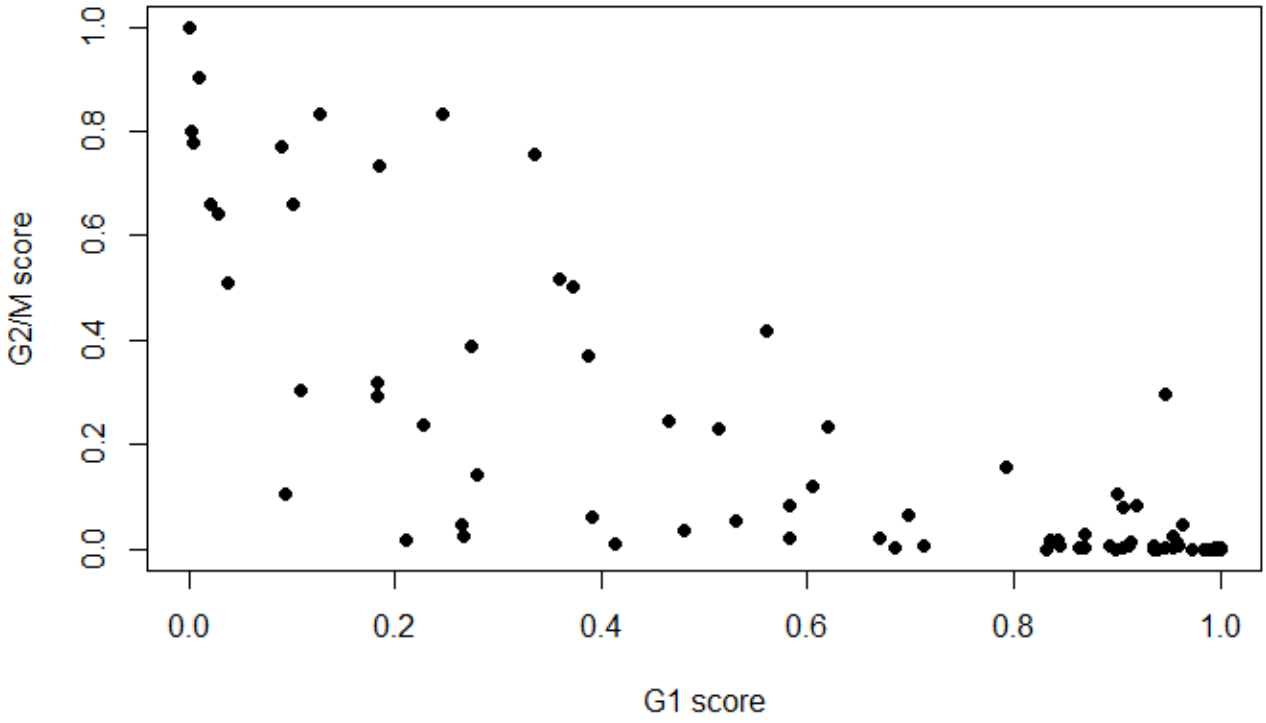


Fig. 8 - Cell cycle phase scores from applying the pair-based classifier on the dataset. Each point represents a cell, plotted according to its scores for G1 and G2/M phases.

How many cells have been predicted as dividing?

Hide

```
sce$phases <- assignments$phases
table(sce$phases)
```

```
 G1 G2M   S
 64  15  15
```

Each cell is assigned a score for each phase, with a higher score corresponding to a higher probability that the cell is in that phase. We focus on the G1 and G2/M scores as these are the most informative for classification. Cells are classified as being in G1 phase if the G1 score is above 0.5 and greater than the G2/M score; in G2/M phase if the G2/M score is above 0.5 and greater than the G1 score; and in S phase if neither score is above 0.5. We can use the assigned phase as a blocking factor in downstream analyses. This protects against cell cycle effects without discarding information.

Examining gene-level expression metrics

Inspecting the most highly expressed genes. This should generally be dominated by constitutively expressed transcripts, such as those for ribosomal or mitochondrial proteins. The presence of other classes of features may be cause for concern if they are not consistent with expected biology. For example, the absence of ribosomal proteins and/or the presence of their pseudogenes are indicative of suboptimal alignment.

Hide

```
fontsize <- theme(axis.text=element_text(size=12), axis.title=element_text(size=16))
plotQC(sce, type = "highest-expression", n=50) + fontsize
```

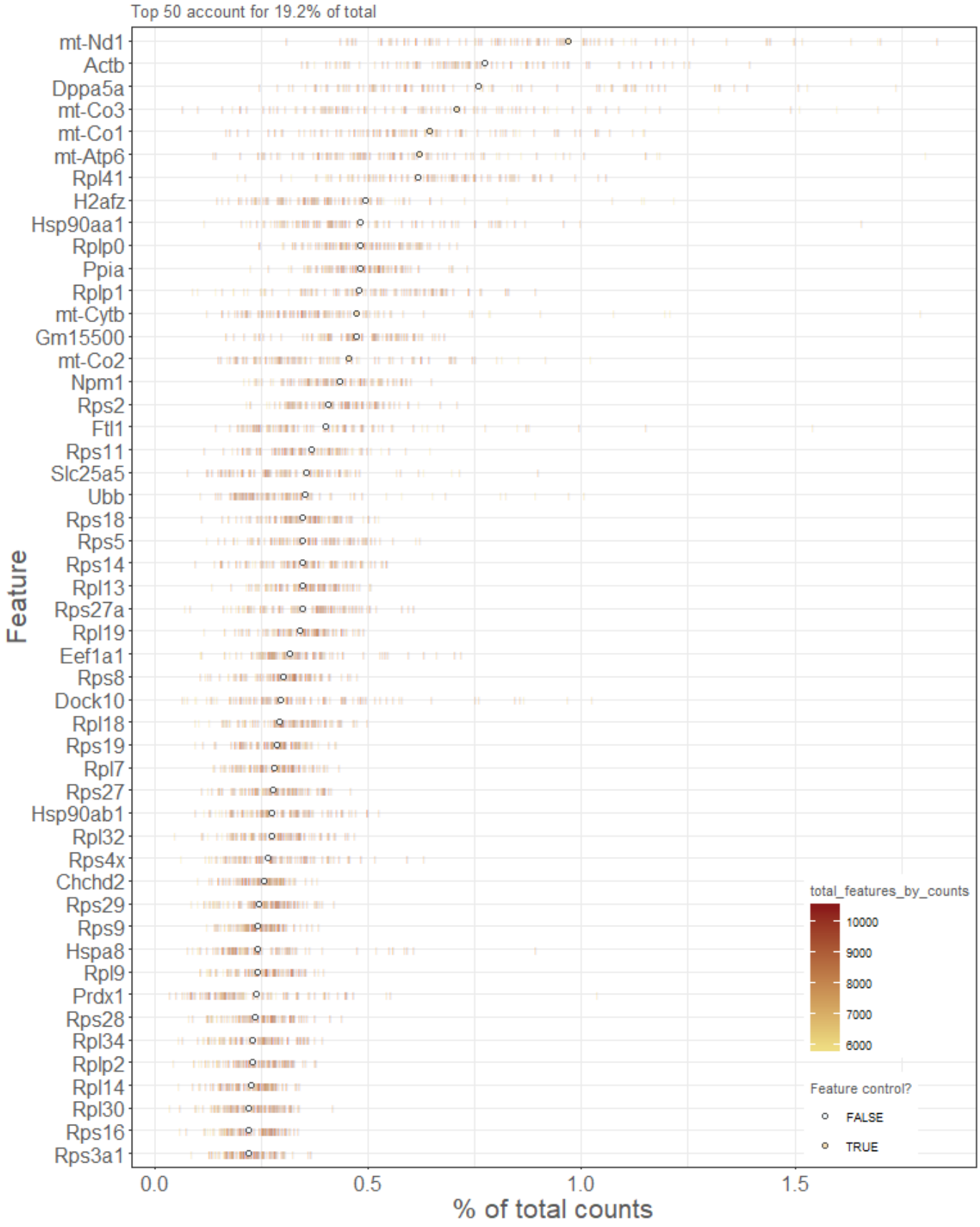


Fig. 9 - Percentage of total counts assigned to the top 50 most highly-abundant features in the dataset. For each feature, each bar represents the percentage assigned to that feature for a single cell, while the circle represents the average across all cells. Bars are coloured by the total number of expressed features in each cell, while circles are coloured according to whether the feature is labelled as a control feature.

Filtering out low-abundance genes

The average count for each gene, is computed across all cells in the dataset using the calcAverage() function, which also performs some adjustment for library size differences between cells. Typically can be observed a peak of moderately expressed genes following a plateau of lowly expressed genes.

Hide

```
ave.counts <- calcAverage(sce, use_size_factors=FALSE)
hist(log10(ave.counts), breaks=100, main="", col="grey80", xlab=expression(Log[10]~"average coun
t"))
abline(v=log10(1), col="blue", lwd=2, lty=2)
```

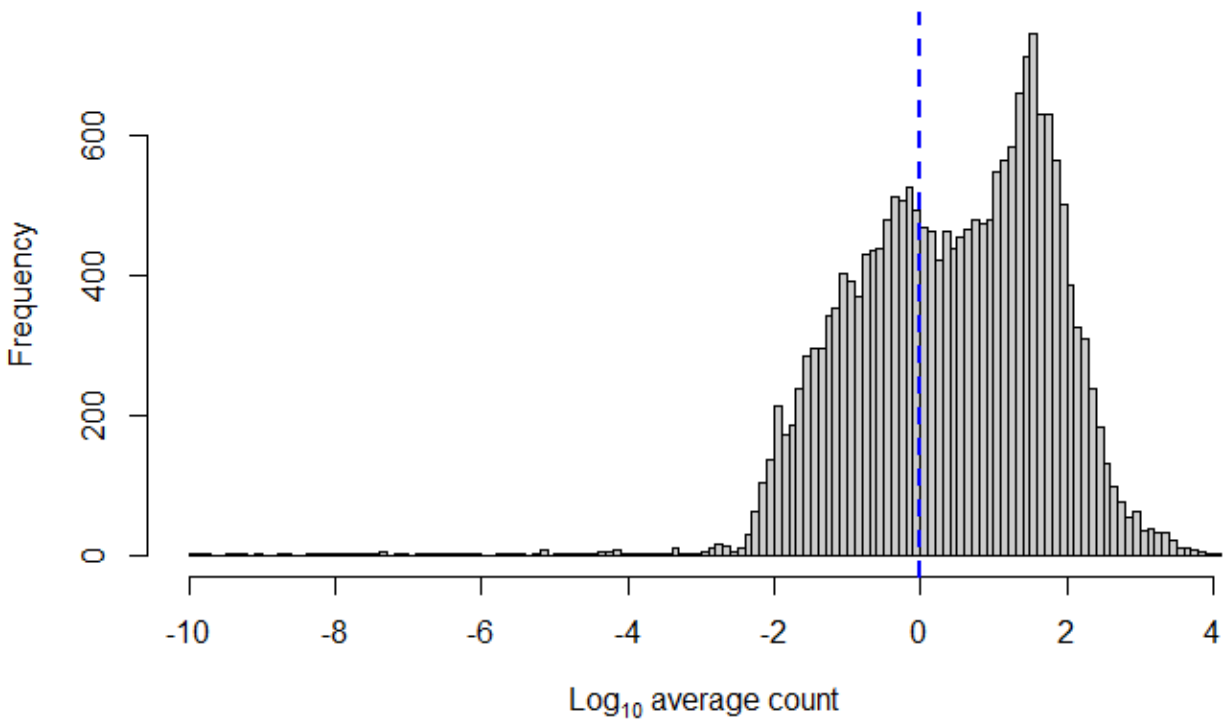


Fig. 10 - Histogram of log-average counts for all genes in the dataset. The filter threshold is represented by the blue line.

All the genes with average counts less than 1 should be removed

Hide

```
demo.keep <- ave.counts >= 1
summary(demo.keep)
```

```
   Mode   FALSE    TRUE
logical   17470   12799
```

The number of TRUE values corresponds to the number of retained rows/genes after filtering. Apply the threshold and create a filtered sce

Hide

```
filtered.sce <- sce[demo.keep,]
```

Examine number of cells that express each gene

Genes expressed in very few cells are often uninteresting as they are driven by amplification artifacts.

Hide

```
num.cells <- nexprs(sce, byrow=TRUE)
smoothScatter(log10(ave.counts), num.cells, ylab="Number of cells", xlab=expression(Log[10]~"aver
age count"))
```

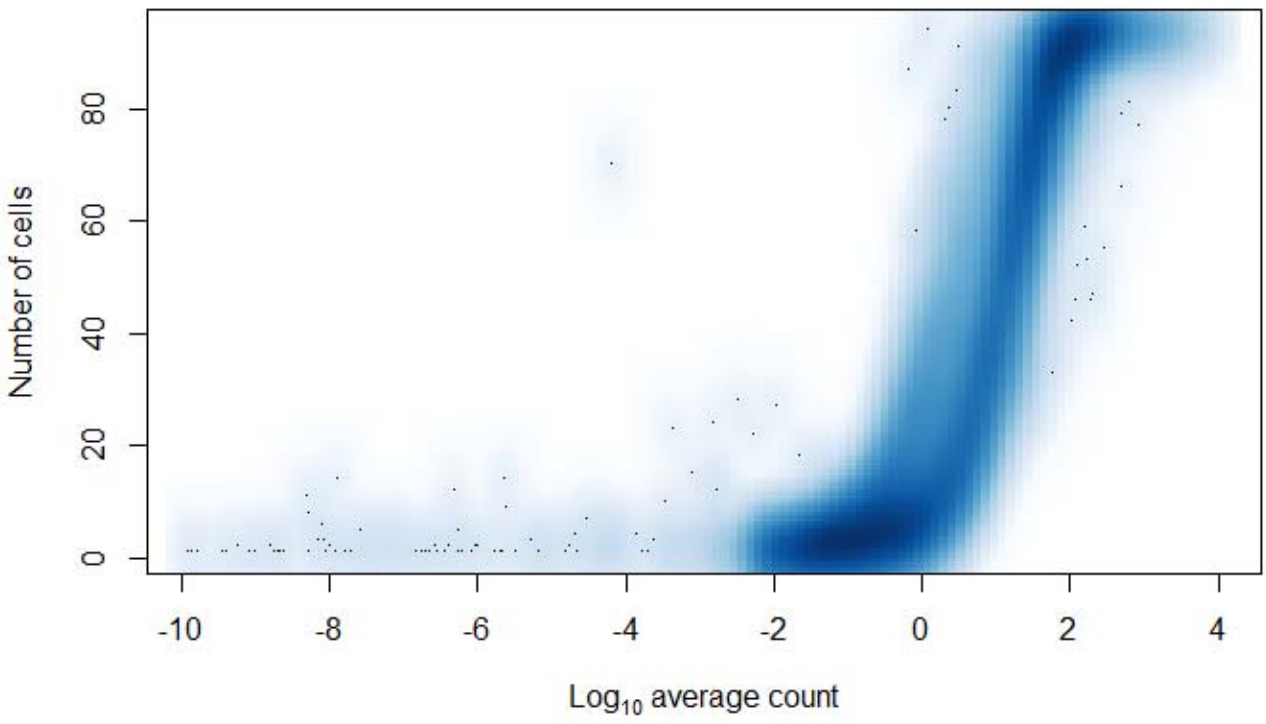


Fig. 11 - The number of cells expressing each gene in the dataset, plotted against the log-average count. Intensity of colour corresponds to the number of genes at any given location.

Genes that are not expressed in any cell are removed to reduce teh computational work in downstream steps

Hide

```
to.keep <- num.cells > 0
sce <- sce[to.keep,]
summary(to.keep)
```

```
   Mode   FALSE    TRUE
logical    9593   20676
```

Normalization of cell-specific biases

Any systematic difference in count size across the non-DE majority of genes between two cells is assumed to represent bias and is removed by scaling. More specifically, “size factors” are calculated that represent the extent to which counts should be scaled in each library. Single-cell data can be problematic for these bulk data-based methods due to the dominance of low and zero counts. To overcome this, counts are pooled from many cells to increase the count size for accurate size factor estimation. Pool-based size factors are then “deconvolved” into cell-based factors for cell-specific normalization.

Hide

```
# check how many samples remained
dim(sce)
```

```
[1] 20676    94
```

Hide

```
# pooling groups of cells to calculate size factor
sce <- computeSumFactors(sce, sizes=c(10, 20, 30, 40, 50, 60, 70, 80, 94))
summary(sizeFactors(sce))
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 0.2601  0.6346  0.9523  1.0000  1.2943  2.4277
```

Hide

```
plot(sizeFactors(sce), sce$total_counts/1e6, log="xy",
     ylab="Library size (millions)", xlab="Size factor")
```

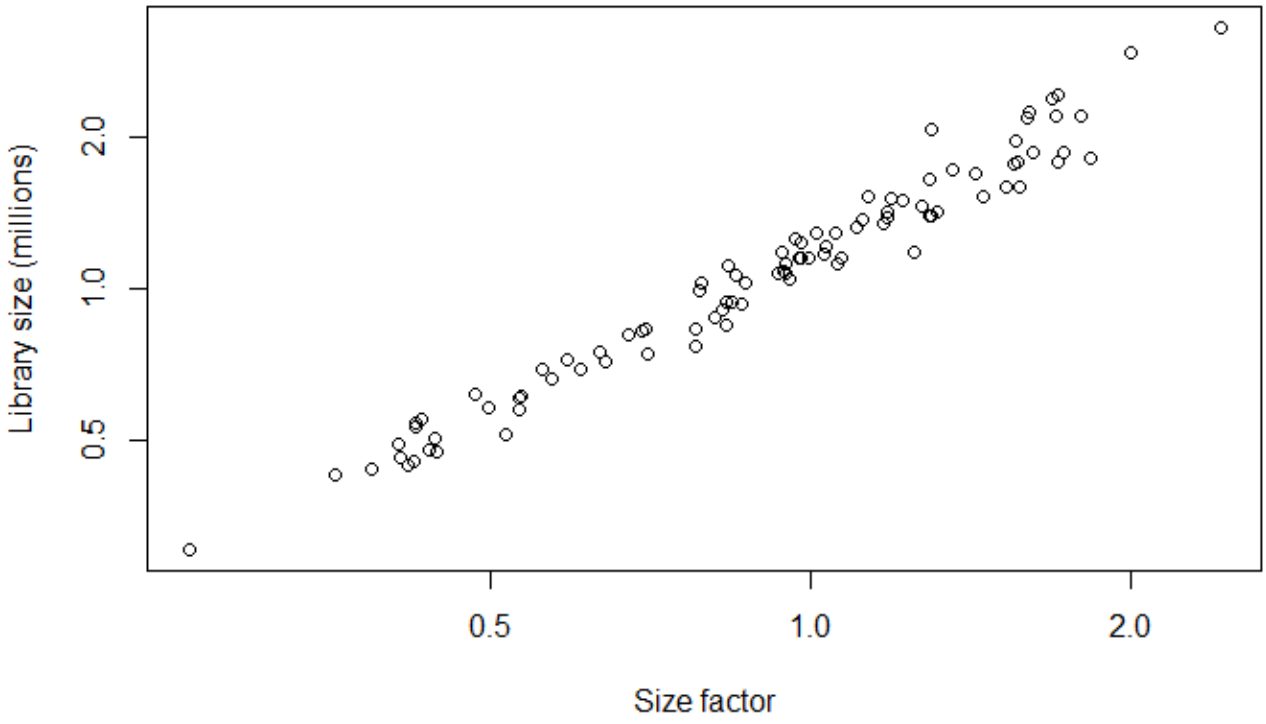


Fig. 12 - Size factors from deconvolution, plotted against library sizes for all cells in the dataset. Axes are shown on a log-scale.

Applying the size factors to normalize gene expression

Hide

```
sce <- normalize(sce)
```

Modelling the technical noise in gene expression

Testing for significantly positive biological components (The biological component for each gene is defined as the difference between its total variance and the fitted value of the trend) First, the biological and technical components of the gene-specific variance are computed…

Hide

```
var.fit <- trendVar(sce, parametric=TRUE, loess.args=list(span=0.3), use.spikes=FALSE)
```

…then biological and technical variance are decomposed

Hide

```
var.out <- decomposeVar(sce, var.fit)
head(var.out)
```

```
DataFrame with 6 rows and 6 columns
                    mean            total                  bio              tech               p.val
ue                   FDR
               <numeric>        <numeric>            <numeric>         <numeric>             <numeri
c>             <numeric>
Gnai3  6.43996844488087 2.12755865901438 -0.0843446387881199   2.2119032978025    0.5848602190016
21                     1
Cdc45  3.55108717032302 7.06734145541664   0.938095533748257  6.12924592166838    0.1484635897737
48    0.590425530887672
Scml2 0.180777536838387 1.01831320963581    0.48428613984977 0.534027069786036 3.23700043355205e-
07 1.71610822984929e-05
Narf   2.19078721201621 4.95617208316592  -0.669945190403642  5.62611727356956    0.7873513318557
59                     1
Klf6   6.73669098818416 3.64187713178669    1.76877770470604  1.87309942708065 1.34943627895102e-
07 7.68621060704992e-06
Scmh1  2.19380979999772 4.08765520704053   -1.54255903638057   5.6302142434211     0.978444514132
97                     1
```

We can visually inspect the trend

Hide

```
plot(var.out$mean, var.out$total, pch=16, cex=0.6, xlab="Mean log-expression",
    ylab="Variance of log-expression", main="E4.5")
curve(var.fit$trend(x), col="dodgerblue", lwd=2, add=TRUE)
```

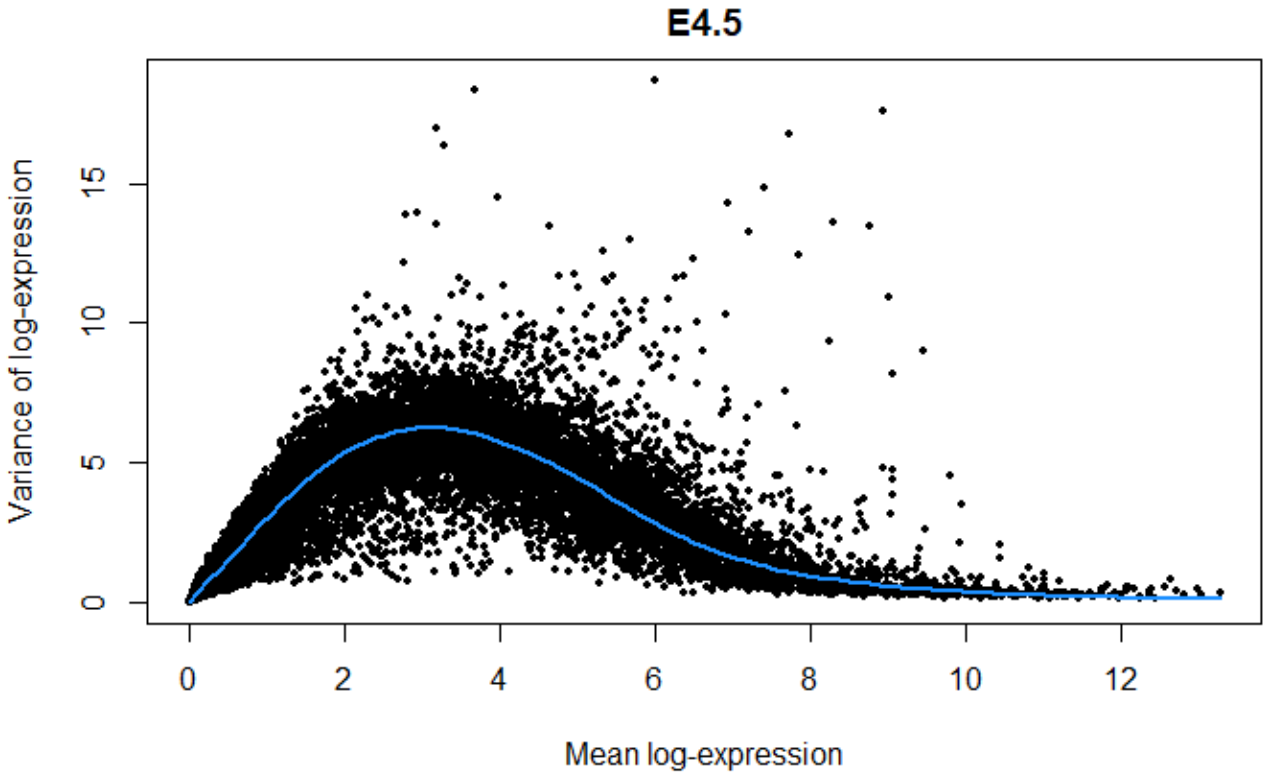


Fig. 13 - Variance of normalized log-expression values for each gene in the dataset, plotted against the mean log-expression. The blue line represents the mean-dependent trend fitted to the variances.

The wave-like shape is typical of the mean-variance trend for log-expression values. A linear increase in the variance is observed as the mean increases from zero, as larger variances are possible when the counts increase. At very high abundances, the effect of sampling noise decreases due to the law of large numbers, resulting in a decrease in the variance.

Visualizing data in low-dimensional space

Once the technical noise is modelled, principal components analysis can be used to remove random technical noise. Consider that each cell represents a point in the high-dimensional expression space, where the spread of points represents the total variance. PCA identifies axes in this space that capture as much of this variance as possible. Each axis is a principal component (PC), where any early PC will explain more of the variance than a later PC.

It is assumed that biological processes involving co-regulated groups of genes will account for the most variance in the data. If this is the case, this process should be represented by one or more of the earlier PCs. In contrast, random technical noise affects each gene independently and will be represented by later PCs. The denoisePCA() function removes later PCs until the total discarded variance is equal to the sum of technical components for all genes used in the PCA.

Hide

```
sce <- denoisePCA(sce, technical=var.out, assay.type="logcounts")
dim(reducedDim(sce, "PCA"))
```

```
[1] 94  5
```

The function returns a SingleCellExperiment object containing the PC scores for each cell in the reducedDims slot. The aim is to eliminate technical noise and enrich for biological signal in the retained PCs. This improves resolution of the underlying biology during downstream procedures such as clustering.

Now relationships between cells can be visualized by constructing pairwise PCA plots for the first three components

Hide

```
plotReducedDim(sce, use_dimred="PCA", ncomponents=3) + fontsize
```

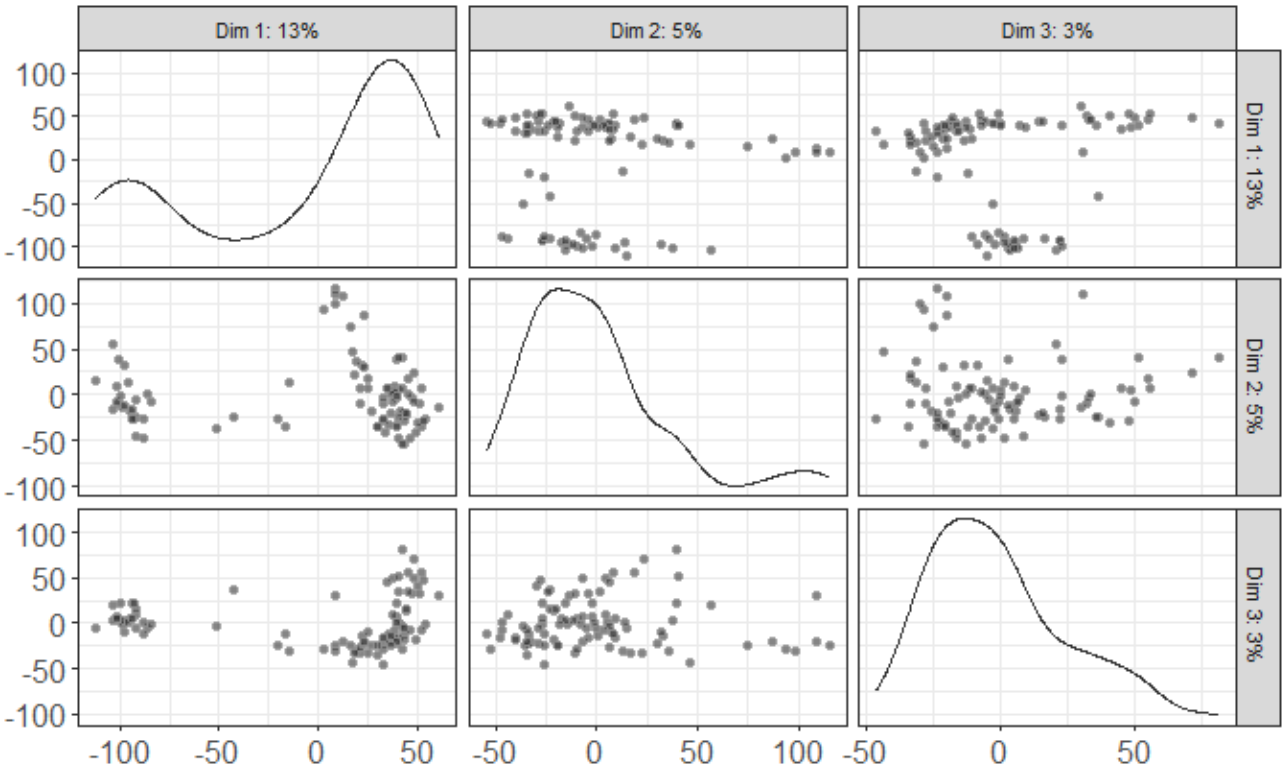


Fig. 14 - Pairwise PCA plots of the first three PCs in the E4.5 dataset, constructed from normalized log-expression values of genes with positive biological components

t-SNE (t-stochastic neighbour embedding) tends to work better than PCA for separating cells in more diverse populations. This is because the former can directly capture non-linear relationships in high-dimensional space, whereas the latter must represent them on linear axes.

use_dimred="PCA" can be set to perform the t-SNE on the low-rank approximation of the data, allowing the algorithm to take advantage of the previous denoising step.

Hide

```
set.seed(100)
out5 <- plotTSNE(sce, run_args=list(use_dimred="PCA", perplexity=5)) + fontsize + ggtitle("Perple
xity = 5")
set.seed(100)
out10 <- plotTSNE(sce, run_args=list(use_dimred="PCA", perplexity=10)) + fontsize + ggtitle("Perp
lexity = 10")
set.seed(100)
out20 <- plotTSNE(sce, run_args=list(use_dimred="PCA", perplexity=20)) + fontsize + ggtitle("Perp
lexity = 20")
multiplot(out5, out10, out20, cols=3)
```

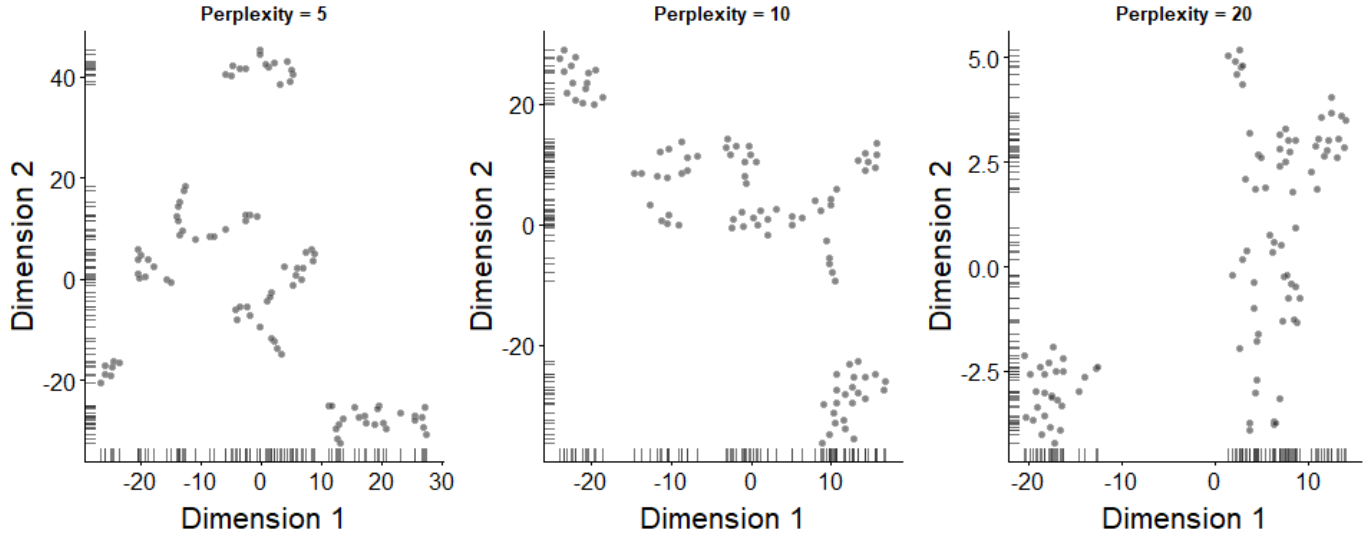


Fig. 15 - t-SNE plots constructed from the denoised PCs in the E4.5 dataset, using a range of perplexity values.

Scripts should set a seed to ensure that the chosen results are reproducible. It is also advisable to test different settings of the “perplexity” parameter as this will affect the distribution of points in the low-dimensional space.

Now run runTSNE() with a perplexity of 20 to store the t-SNE results inside our SingleCellExperiment object. This avoids repeating the calculations whenever we want to create a new plot with plotTSNE(), as the stored results will be used instead.

Hide

```
set.seed(100)
sce <- runTSNE(sce, use_dimred="PCA", perplexity=20)
reducedDimNames(sce)
```

```
[1] "PCA"  "TSNE"
```

Clustering cells into putative subpopulations

The denoised log-expression values are used to cluster cells into putative subpopulations. Specifically, hierarchical clustering is performed on the Euclidean distances between cells, using Ward’s criterion to minimize the total variance within each cluster. This yields a dendrogram that groups together cells with similar expression patterns across the chosen genes.

Hide

```
pcs <- reducedDim(sce, "PCA")
my.dist <- dist(pcs)
my.tree <- hclust(my.dist, method="ward.D2")
```

Clusters are explicitly defined by applying a dynamic tree cut (Langfelder, Zhang, and Horvath 2008) to the dendrogram. This exploits the shape of the branches in the dendrogram to refine the cluster definitions.

Hide

```
library(dynamicTreeCut)
my.clusters <- unname(cutreeDynamic(my.tree, distM=as.matrix(my.dist),
    minClusterSize=10, verbose=0))
```

Let’s see the distribution of cells in each cluster with respect to known factors.

Hide

```
table(my.clusters)
```

```
my.clusters
 1  2  3  4
38 27 15 14
```

Visualize the cluster assignments for all cells on the t-SNE plot

Hide

```
sce$cluster <- factor(my.clusters)
plotTSNE(sce, colour_by="cluster") + fontsize
```

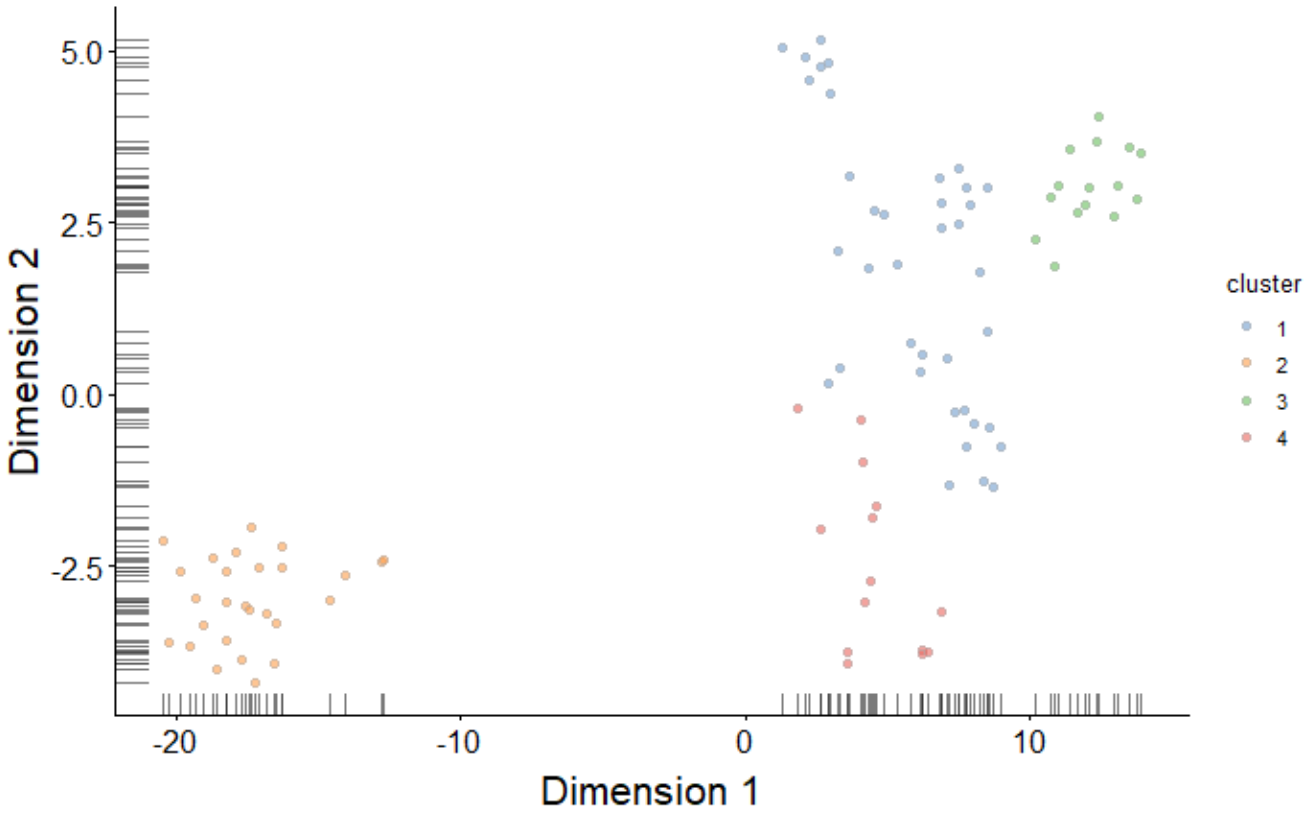


Fig. 16 - t-SNE plot of the denoised PCs of the E4.5 dataset

The separatedness of the clusters is checked using the silhouette width. Cells with large positive silhouette widths are closer to other cells in the same cluster than to cells in different clusters. Conversely, cells with negative widths are closer to other clusters than to other cells in the cluster to which it was assigned.

Hide

```
library(cluster)
clust.col <- scater:::.get_palette("tableau10medium")
sil <- silhouette(my.clusters, dist = my.dist)
sil.cols <- clust.col[ifelse(sil[,3] > 0, sil[,1], sil[,2])]
sil.cols <- sil.cols[order(-sil[,1], sil[,3])]
plot(sil, main = paste(length(unique(my.clusters)), "clusters"), border=sil.cols, col=sil.cols, d
o.col.sort=FALSE)
```

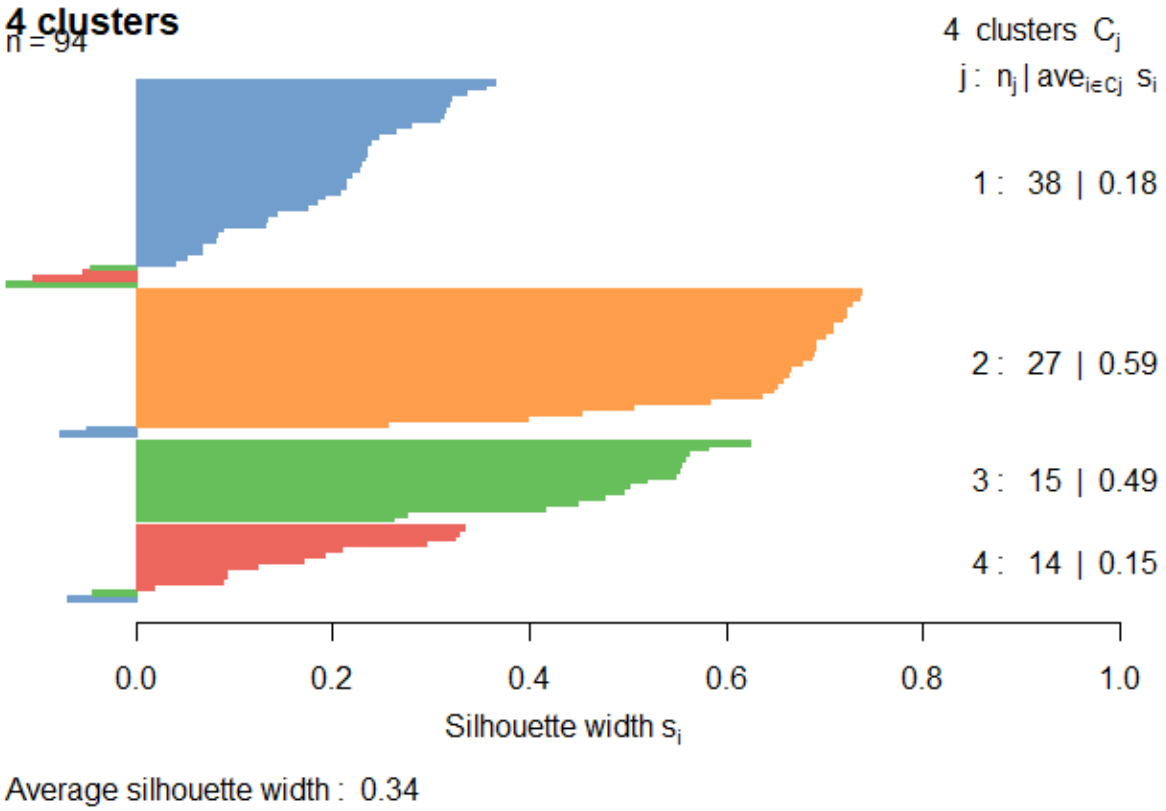


Fig. 17 - Barplot of silhouette widths for cells in each cluster.

Clusters 2 and 3 have very positive widths.

Detecting marker genes between clusters

Once putative subpopulations are identified by clustering, marker genes for each cluster can be identified using the findMarkers function. This fits a linear model to the log-expression values for each gene using limma. The aim is to test for DE in each cluster compared to the others. The top DE genes are likely to be good candidate markers as they can effectively distinguish between cells in different clusters.

For each cluster, the DE results of the relevant comparisons are consolidated into a single output table. This allows a set of marker genes to be easily defined by taking the top DE genes from each pairwise comparison between clusters.

For example, to construct a marker set for cluster 2 from the top 10 genes of each comparison, marker.set is filtered to retain rows with Top less than or equal to 10. Other statistics are also reported for each gene, including the adjusted p-values (see below) and the log-fold changes relative to every other cluster.

Hide

```
markers <- findMarkers(sce, my.clusters)
marker.set <- markers[["2"]]
head(marker.set, 10)
```

```
DataFrame with 10 rows and 5 columns
             Top                  FDR          logFC.1          logFC.3          logFC.4
       <integer>            <numeric>        <numeric>        <numeric>        <numeric>
Slc7a3         1 9.34697574443068e-39 8.83811777941236 7.60948855238368 8.44623677123695
Morc1          1 7.56741266434735e-32 7.58670859197319 7.51067713491794  8.1801100665976
Fgf4           1 3.41096017727184e-30 6.54564441592509 6.52193116147973 6.75382601973837
Sept1          2 9.67829361003804e-32 6.61414090993151 6.31188344791184 6.37536633010199
Tcea3          2 4.14151109901872e-28  6.2273484261884 6.51001387386247 6.51001387386247
Nmnat2         3 4.22278166487226e-22 4.97694490008745 5.25727871733298  4.8505102888349
Igfbp2         4 2.61344641362596e-29 8.22371771149004 7.51508948119044 8.67277447253381
Ubd            4 2.53207652758511e-24 5.89399203616243 6.24006172193389 6.24006172193389
Tdgf1          4 4.64508602169587e-21 8.55626892278324 7.64709415281283 9.49144570438014
Esrrb          5 1.19852696485774e-16 5.33960348343922 5.97772838435186 5.25395552303197
```

Save the list of candidate marker genes for further examination.

Hide

```
write.table(marker.set, file="e45_markers_cl2.tsv", sep="\t",  quote=FALSE, row.names=TRUE)
```

The expression profiles of the top candidates can be visualized to verify that the DE signature is robust. The clusters assignment for every sample and some markers of pluripotency are reported on the top rows.

Identifying correlated gene pairs within the HVGs using Spearman's rho

HVGs are defined as genes with biological components that are significantly greater than zero at a false discovery rate (FDR) of 5%. These genes are interesting as they drive differences in the expression profiles between cells, and should be prioritized for further investigation. In addition, we only consider a gene to be a HVG if it has a biological component greater than or equal to 0.5. For transformed expression values on the log2 scale, this means that the average difference in true expression between any two cells will be at least 2-fold. (This reasoning assumes that the true log-expression values are Normally distributed with variance of 0.5. The root-mean-square of the difference between two values is treated as the average log2-fold change between cells and is equal to unity.) We rank the results by the biological component to focus on genes with larger biological variability.

Hide

```
hvg.out <- var.out[which(var.out$FDR <= 0.05 & var.out$bio >= 0.5),]
hvg.out <- hvg.out[order(hvg.out$bio, decreasing=TRUE),]
nrow(hvg.out)
```

```
[1] 980
```

Hide

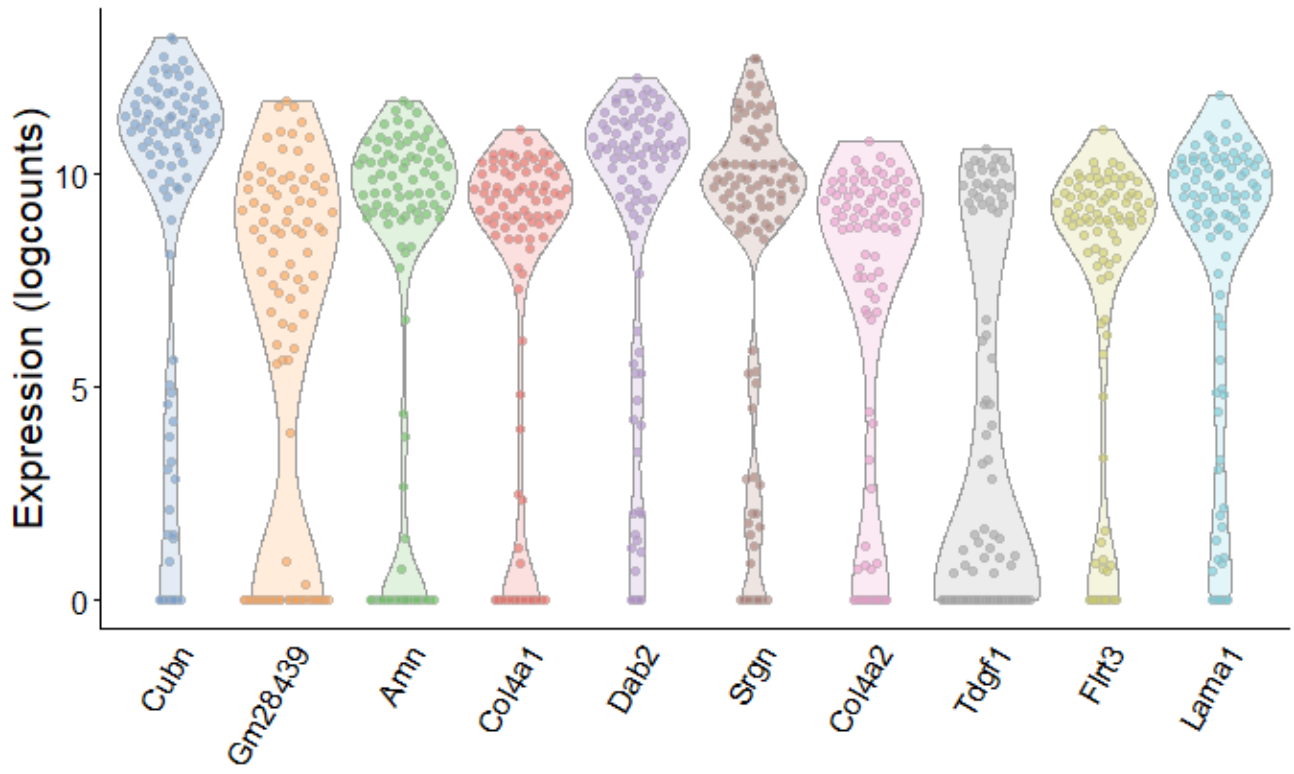
```
write.table(file="e45_hvg.tsv", hvg.out, sep="\t", quote=FALSE, col.names=NA)
head(hvg.out)
```

```
DataFrame with 6 rows and 6 columns
                    mean            total              bio              tech                p.valu
e                    FDR
               <numeric>        <numeric>        <numeric>         <numeric>              <numeric
>             <numeric>
Cubn     8.9464231216087 17.6117331001593 17.0327723852671 0.578960714892159
0                     0
Gm28439 5.99430633063245 18.7253867665672 15.8853638054789  2.84002296108832  1.40763714577247e-7
7  1.32292298299962e-74
Amn     7.72427667622211 16.7970195211331 15.7169862825191  1.08003323861399   1.385724696815e-24
1 4.77520730522448e-238
Col4a1  7.41606583730855 14.8627648752744 13.5919083801956  1.27085649507882 2.59781215100737e-16
9 5.37123640342285e-166
Dab2    8.77095494370033 13.4932465874802 12.8617497535907 0.631496833889549
0                     0
Srgn    8.29228792137331 13.6348570287952 12.8279113084952 0.806945720300036  4.2173898282994e-26
7 1.74397504179837e-263
```

The distribution of expression values for the top HVGs can be seen to ensure that the variance estimate is not being dominated by one or two outlier cells

Hide

```
#png("e45_violin_top_HVGs.png")
plotExpression(sce, rownames(hvg.out)[1:10]) + fontsize
```



Hide

```
#dev.off()
```

Fig. 18 - Violin plots of normalized log-expression values for the top 10 genes with the largest biological components in the E4.5 dataset

Now HVGs that are highly correlated with one another can be identified. This distinguishes between HVGs caused by random noise and those involved in driving systematic differences between subpopulations. Correlations between genes are quantified by computing Spearman's rho, which accommodates non-linear relationships in the expression values. Gene pairs with significantly large positive or negative values of rho are identified using the correlatePairs function.

Calculating correlations for all possible gene pairs would require too much computational time and increase the severity of the multiple testing correction. It may also prioritize uninteresting genes that have strong correlations but low variance, e.g., tightly co-regulated house-keeping genes.

In this study the main interested is finding the genes cotrelated to Nanog. In the E4.5 dataset after gene level summarization (tximport), the technical component of the variance for Nanog expression (6.29) results very high compared to the biological component (-2.76) causing a very low significance (p-value = 0.99), so Nanog is not included among the HVGs (FRD = 1). Nanog has been explicitly included in a coercive way into the CorrelatePairs input list together with the HVGs, only for evaluation purpose on the output list of Nanog correlated genes. (The CorrelatePairs is repeated below at transcript level where the variant Nanog-203 has a good biological variance and is included among the HVGs).

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

Hide

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

```
   Mode   FALSE    TRUE
logical  360936  119754
```

Using Nanog correlated HVGs for further data exploration

Have a look to the top Nanog vorrelated genes, by restricting the list of correlated pairs by significance (FDR <= 5%) and high strength of association (rho > [0.4]).

Hide

```
sig.cor.nanog <- var.cor.nanog$FDR <= 0.05 & abs(var.cor.nanog$rho) >= 0.4
summary(sig.cor.nanog)
```

```
   Mode   FALSE    TRUE
logical     639     341
```

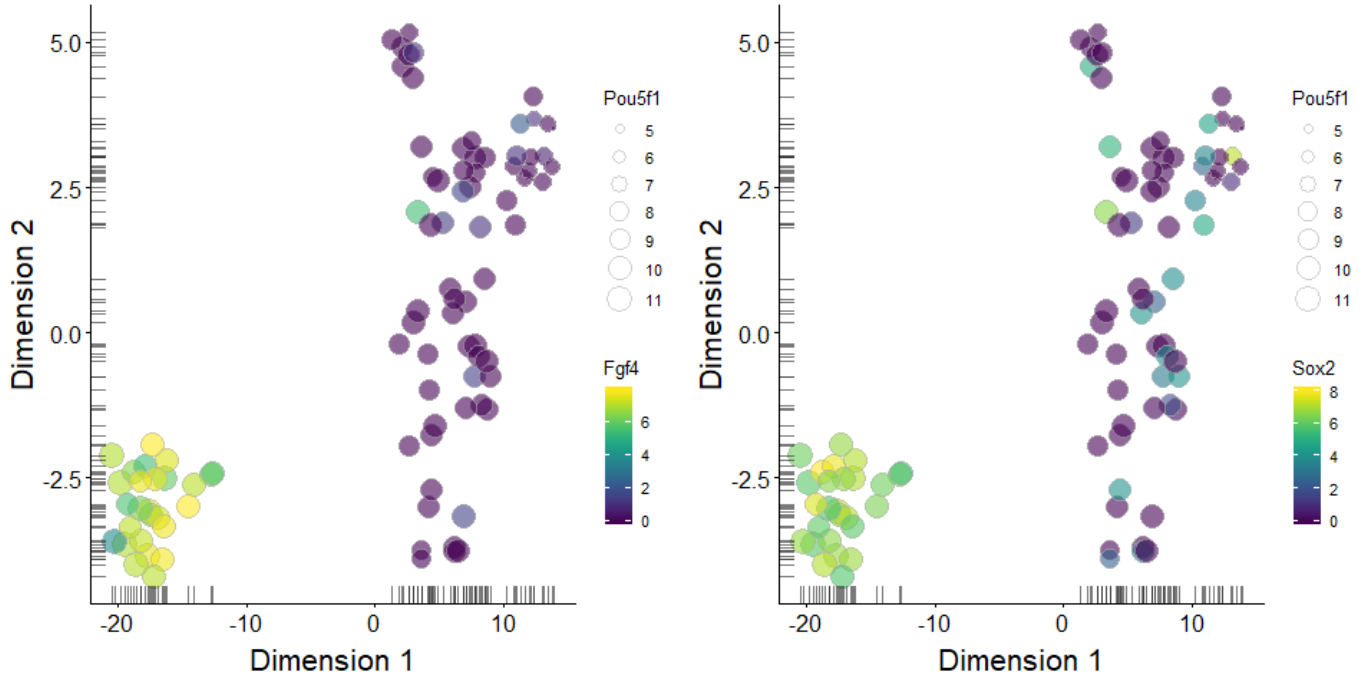
The expression profiles of these top 227 Nanog correlated HVGs can be visualized with a heatmap. All expression values are mean-centred for each gene to highlight the relative differences in expression between cells.

Finding Nanog correlated genes in the epiblast subset of cells

From the heatmak of the top marker genes identified by tSNE, We can observe that the clusters 2 represent a population of pluripotent cells (Oct4+) with epiblast features (Sox2+, Fgf4+) while the rest of cells could be attributed the the primitive endoderm (Sox17+, Dab2+). It is also known by literature (Aksoy et al. 2013) that at this stage Oct4 switches partner from Sox2 to Sox17 contributing to the commitment of the PrE differentiation. This hipotesis can be verified by comparing visually the expression levels of the markers for these two complementary populations.

Hide

```
tsne.fgf4 <- plotTSNE(sce, colour_by="Fgf4", size_by="Pou5f1") + fontsize
tsne.sox2 <- plotTSNE(sce, colour_by="Sox2", size_by="Pou5f1") + fontsize
#png("e45_tSNE_fgf4.png", width = 900)
multiplot(tsne.fgf4, tsne.sox2, cols=2)
```



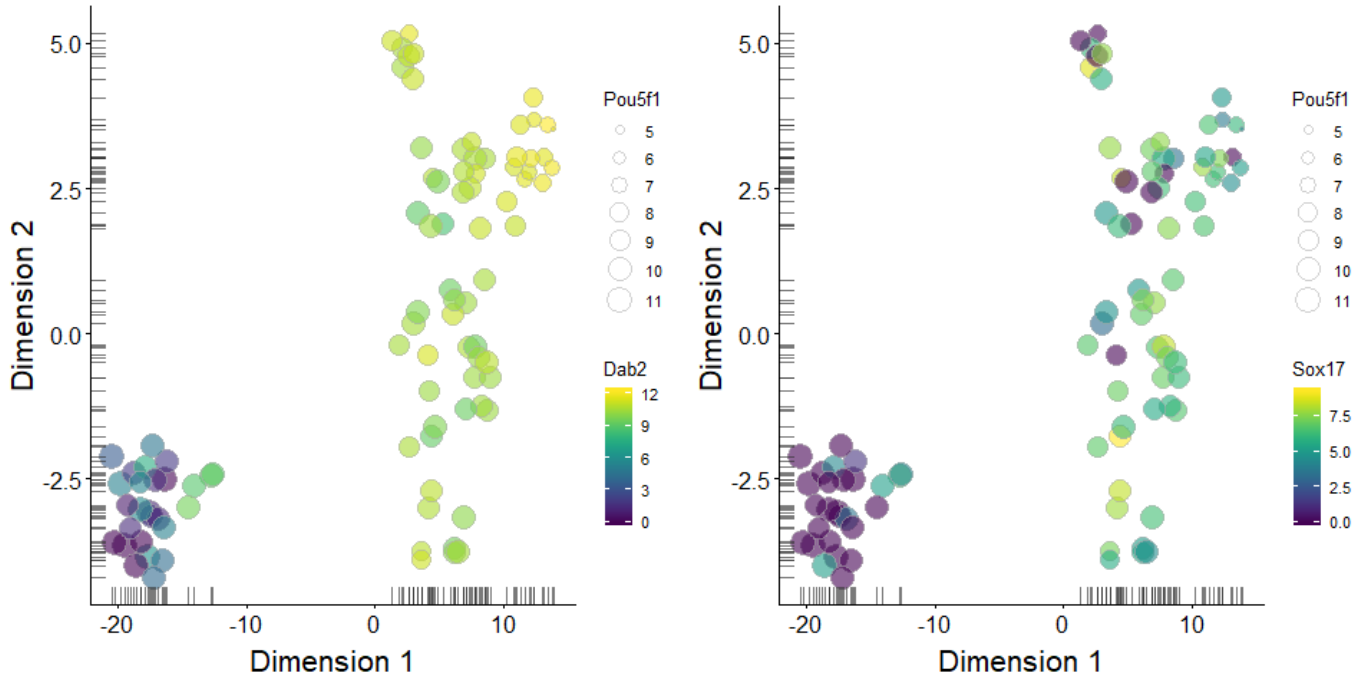
Hide

```
#dev.off()
```

Fig. 19 - t-SNE plot of the denoised PCs of the E4.5 dataset showing the expression distribution of epiblast markers across samples

Hide

```
tsne.dab2 <- plotTSNE(sce, colour_by="Dab2", size_by="Pou5f1") + fontsize
tsne.sox17 <- plotTSNE(sce, colour_by="Sox17", size_by="Pou5f1") + fontsize
#png("e45_tSNE_dab2.png", width = 900)
multiplot(tsne.dab2, tsne.sox17, cols=2)
```



Hide

```
#dev.off()
```

Fig. 20 - t-SNE plot of the denoised PCs of the E4.5 dataset showing the expression distribution of primitive endoderm markers across samples

So the analysis can be concentrated on the subset of cells of the cluster 2, showing features of the epiblast.

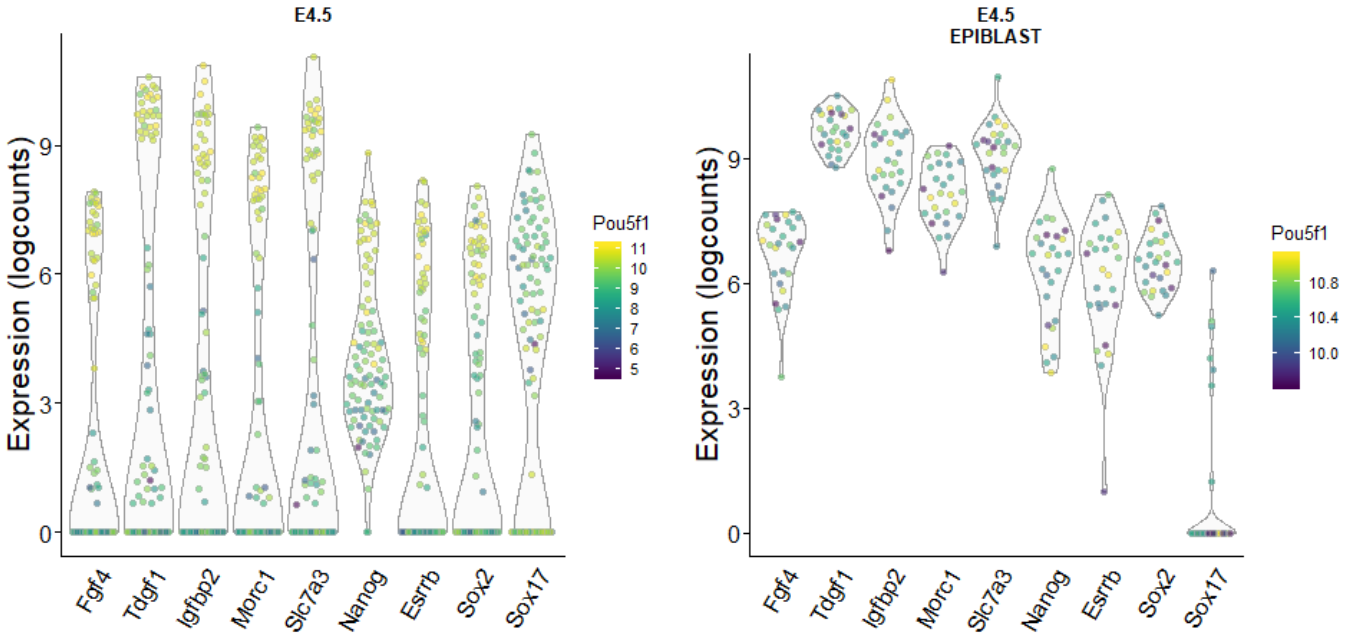
Hide

```
epi.sce <- sce[,sce$cluster == 2]
dim(epi.sce)
```

It is possible to define a set of epiblast (Epi) and primitive endoderm (PrE) markers chosen among the top markers assigned to cluster 2 by tSNE and comparing the distribution of these markers between the full dataset and the putative epiblast subset.

Hide

```
episet <- c("Fgf4", "Tdgf1", "Igfbp2", "Morc1", "Slc7a3", "Nanog", "Esrrb", "Sox2", "Sox17")
#png("e45_violin_episet.png", width = 1200, height = 640)
episet.pre.sub <- plotExpression(sce, episet, colour_by="Pou5f1") + fontsize + ggtitle("E4.5")
episet.post.sub <- plotExpression(epi.sce, episet, colour_by="Pou5f1") + fontsize + ggtitle("E4.5
\nEPIBLAST")
multiplot(episet.pre.sub, episet.post.sub, cols = 2)
```



Hide

```
#dev.off()
```

Fig. 21 - Violin plots of normalized log-expression values for a set of epiblast markers across cells before and after subsetting of tSNE cluster 2

The strongest epiblast marker identified, showing the best separation between the high Oct4 and low Oct4 populations is Tdgf1, which is coding for a protein involved in Nodal signaling and plays a role in the determination of the epiblastic cells that subsequently give rise to the mesoderm.

It is required to normalize again for cell-specific biases in the subset

Hide

```
epi.sce <- computeSumFactors(epi.sce, sizes=c(5, 10, 15, 20, 25))
summary(sizeFactors(epi.sce))
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
 0.4058  0.7282  1.0068  1.0000  1.1870  1.7142 
```

Apply the size factors to normalize gene expression

Hide

```
epi.sce <- normalize(epi.sce)
```

Identify correlated gene pairs within the epiblast subpopulation

Hide

```
set.seed(100)
epi.var.cor <- correlatePairs(epi.sce, iters=1e8, subset.row=c(rownames(hvg.out), "Nanog"))
epi.nanog.rows <- epi.var.cor$gene1 == "Nanog" | epi.var.cor$gene2 == "Nanog"
epi.var.cor.nanog <- epi.var.cor[epi.nanog.rows,]
write.table(file="epi.e45_cor.tsv", epi.var.cor, sep="\t", quote=FALSE, row.names=FALSE)
write.table(file="epi.e45_cor_nanog.tsv", epi.var.cor.nanog, sep="\t", quote=FALSE, row.names=FAL
SE)
head(epi.var.cor.nanog)
```

```
DataFrame with 6 rows and 6 columns
        gene1       gene2                rho            p.value                FDR   limited
  <character> <character>          <numeric>          <numeric>          <numeric> <logical>
1        Stx3       Nanog -0.672771672771673 0.0001784799982152 0.355408987034145     FALSE
2        Wdr5       Nanog  0.656898656898657 0.0002848199971518 0.379252422246257     FALSE
3      Eif2b5       Nanog  0.630647130647131 0.0005631199943688  0.41458809322254     FALSE
4       Smpd5       Nanog  -0.61965811965812 0.0007417399925826 0.421450351104645     FALSE
5        Dok2       Nanog -0.617826617826618 0.0007761799922382 0.422049977597682     FALSE
6     Gm28437       Nanog  0.608058608058608 0.0009720799902792 0.430411830028307     FALSE
```

Hide

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

```
   Mode   FALSE    TRUE 
logical  480686       4 
```

Hide

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

```
   Mode   FALSE 
logical     980 
```

Analysis at transcript level

Create singleCellExperiment from Kallisto abundance files using Tximport without collapsing the rows by gene (txOut = TRUE)

Hide

```
sce2 <- readTxResults(samples = samples, files = files, type = "kallisto", txOut = TRUE)
```

```
Kallisto log not provided - assuming all runs successful
```

```
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 3
6 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68
69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100
101 102 103 104 105
```

```
Using log2(TPM + logExprsOffset) as 'exprs' values in output.
```

Retrieve annotation info from ensembldb and store it into sce2

Rename rownames to include gene symbols

Hide

```
rownames(sce2) <- paste0(rowData(sce2)$mgi_symbol, "_",rownames(sce2))
head(rownames(sce2))
```

```
[1] "Trdd1_ENSMUST00000196221.1"   "Trdd1_ENSMUST00000179664.1"   "Trdd2_ENSMUST00000177564.1"
"Trbd1_ENSMUST00000178537.1"
[5] "Trbd2_ENSMUST00000178862.1"   "Ighd4-1_ENSMUST00000179520.1"
```

Define references for control_features to be used in calculateQCMetrics()

Hide

```
mito2 <- which(rowData(sce2)$chromosome_name=="MT")
```

Calculate cell metrics

Hide

```
sce2 <- calculateQCMetrics(sce2, feature_controls=list(Mt=mito2))
```

```
Note that the names of some metrics have changed, see 'Renamed metrics' in ?calculateQCMetrics.
Old names are currently maintained for back-compatibility, but may be removed in future releases.
```

Hide

```
names(colData(sce2))
```

```
 [1] "n_features"                                    "n_bootstraps"

 [3] "is_cell_control"                               "total_features_by_counts"

 [5] "log10_total_features_by_counts"                "total_counts"

 [7] "log10_total_counts"                            "pct_counts_in_top_50_features"

 [9] "pct_counts_in_top_100_features"                "pct_counts_in_top_200_features"

[11] "pct_counts_in_top_500_features"                "total_features"

[13] "log10_total_features"                          "pct_counts_top_50_features"

[15] "pct_counts_top_100_features"                   "pct_counts_top_200_features"

[17] "pct_counts_top_500_features"                   "total_features_by_counts_endogenous"

[19] "log10_total_features_by_counts_endogenous"     "total_counts_endogenous"

[21] "log10_total_counts_endogenous"                 "pct_counts_endogenous"

[23] "pct_counts_in_top_50_features_endogenous"      "pct_counts_in_top_100_features_endogenous"

[25] "pct_counts_in_top_200_features_endogenous"     "pct_counts_in_top_500_features_endogenous"

[27] "total_features_endogenous"                     "log10_total_features_endogenous"

[29] "pct_counts_top_50_features_endogenous"         "pct_counts_top_100_features_endogenous"

[31] "pct_counts_top_200_features_endogenous"        "pct_counts_top_500_features_endogenous"

[33] "total_features_by_counts_feature_control"      "log10_total_features_by_counts_feature_con
trol"
[35] "total_counts_feature_control"                  "log10_total_counts_feature_control"

[37] "pct_counts_feature_control"                    "total_features_feature_control"

[39] "log10_total_features_feature_control"          "total_features_by_counts_Mt"

[41] "log10_total_features_by_counts_Mt"             "total_counts_Mt"

[43] "log10_total_counts_Mt"                         "pct_counts_Mt"

[45] "total_features_Mt"                             "log10_total_features_Mt"
```

Define the quality control metrics and identifying outliers for each metric

Hide

```
libsize.drop2 <- isOutlier(sce2$total_counts, nmads=3, type="lower", log=TRUE)
feature.drop2 <- isOutlier(sce2$total_features, nmads=3, type="lower", log=TRUE)
mito.drop2 <- isOutlier(sce2$pct_counts_Mt, nmads=3, type="higher")
keep2 <- !(libsize.drop2 | feature.drop2 | mito.drop2)
# Subsetting by column will retain only the high-quality cells that pass each filter
sce2 <- sce2[,!(libsize.drop2 | feature.drop2 | mito.drop2)]
```

Remove genes that are not expressed in any cell to reduce computational work in downstream steps

Hide

```
num.cells2 <- nexprs(sce2, byrow=TRUE)
to.keep2 <- num.cells2 > 0
sce2 <- sce2[to.keep2,]
```

Normalization of cell-specific biases

Hide

```
dim(sce2)
```

```
[1] 81655    94
```

Hide

```
sce2 <- computeSumFactors(sce2, sizes=c(5, 10, 20, 40, 60, 80, 94))
sce2 <- normalize(sce2)
```

Model the technical noise in gene expression

Hide

```
var.fit2 <- trendVar(sce2, parametric=TRUE, loess.args=list(span=0.3), use.spikes=FALSE)
var.out2 <- decomposeVar(sce2, var.fit2)
```

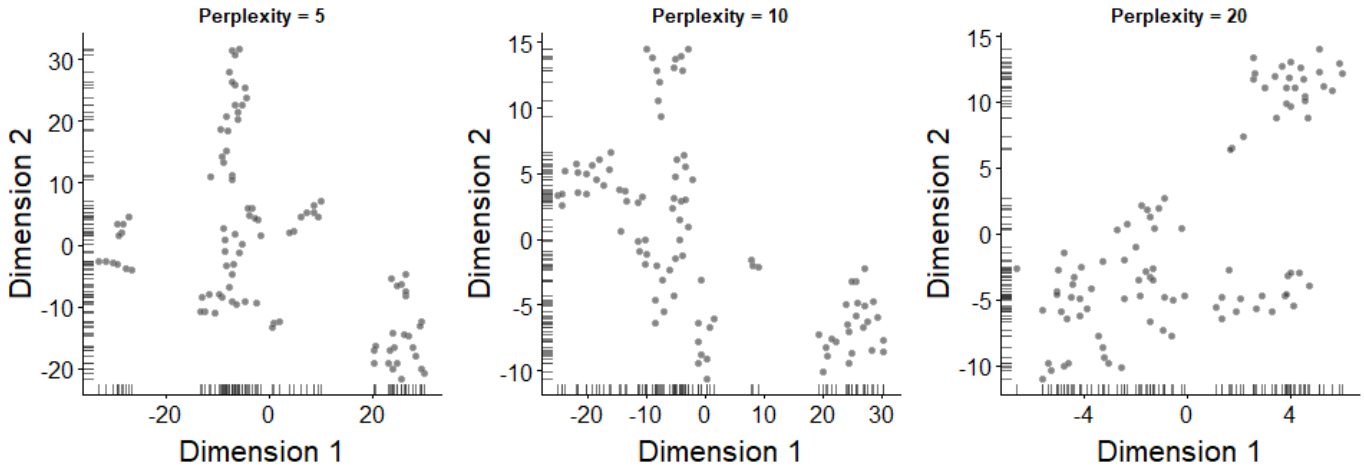
Visualize data in low-dimensional space

Hide

```
sce2 <- denoisePCA(sce2, technical=var.out2, assay.type="logcounts")
```

Hide

```
set.seed(100)
out5.2 <- plotTSNE(sce2, run_args=list(use_dimred="PCA", perplexity=5)) + fontsize + ggtitle("Per
plexity = 5")
set.seed(100)
out10.2 <- plotTSNE(sce2, run_args=list(use_dimred="PCA", perplexity=10)) + fontsize + ggtitle("P
erplexity = 10")
set.seed(100)
out20.2 <- plotTSNE(sce2, run_args=list(use_dimred="PCA", perplexity=20)) + fontsize + ggtitle("P
erplexity = 20")
#png("e45.mRNA_tSNE_test.png", width=1800, height=600)
multiplot(out5.2, out10.2, out20.2, cols=3)
```



Hide

```
#dev.off()
```

Fig. 22 - t-SNE plots constructed from the denoised PCs in the E4.5 transcripts level dataset, using a range of perplexity values

Hide

```
set.seed(100)
sce2 <- runTSNE(sce2, use_dimred="PCA", perplexity=10)
```

Clustering cells into putative subpopulations

Hide

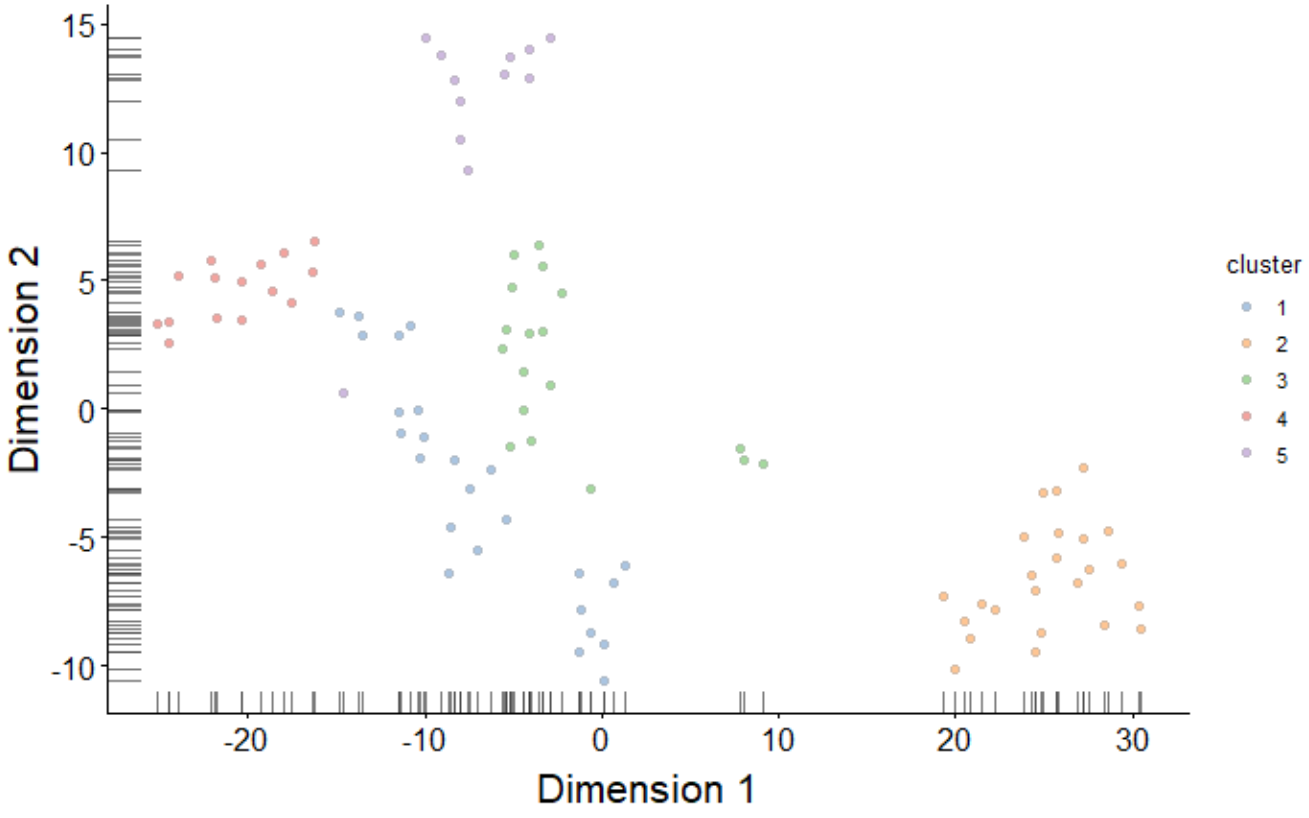
```
pcs2 <- reducedDim(sce2, "PCA")
my.dist2 <- dist(pcs2)
my.tree2 <- hclust(my.dist2, method="ward.D2")
```

Hide

```
library(dynamicTreeCut)
my.clusters2 <- unname(cutreeDynamic(my.tree2, distM=as.matrix(my.dist2),
    minClusterSize=10, verbose=0))
```

Hide

```
sce2$cluster <- factor(my.clusters2)
#png("e45.mRNA_tSNE.png")
plotTSNE(sce2, colour_by="cluster") + fontsize
```



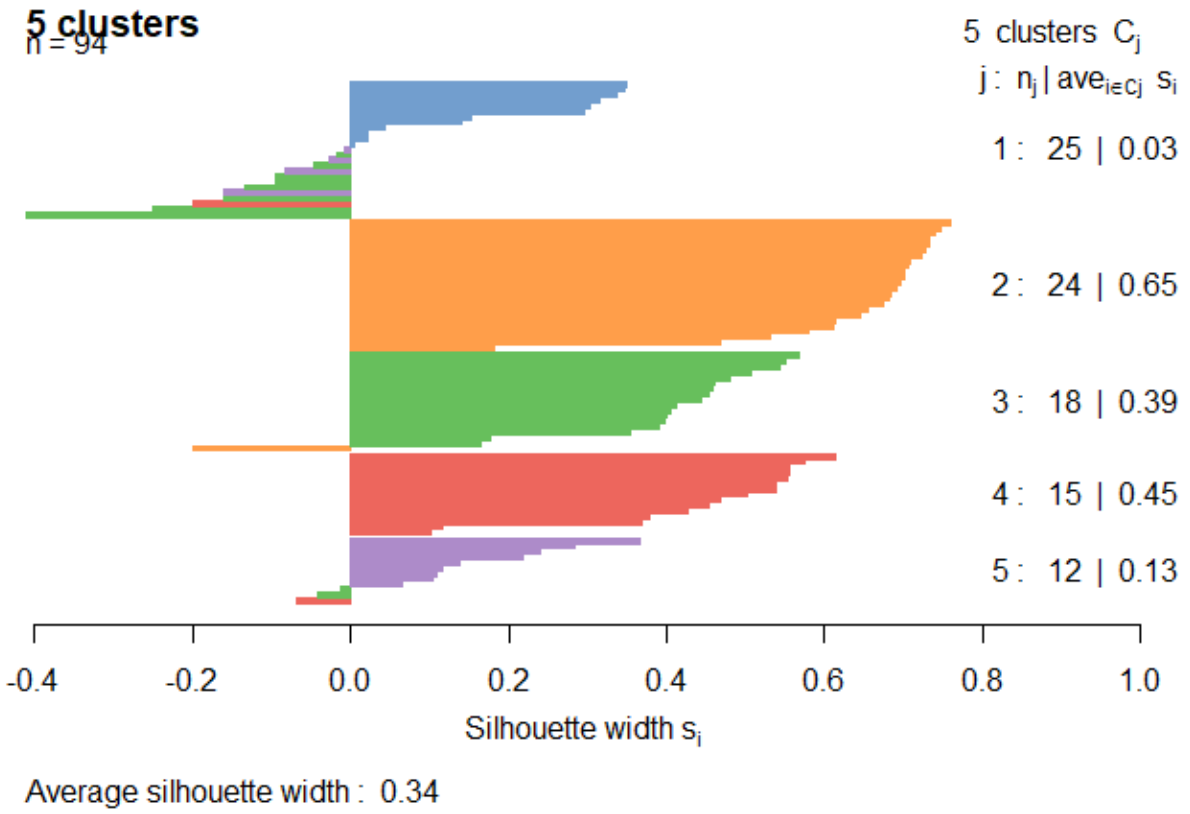
Hide

```
#dev.off()
```

Fig. 23 - t-SNE plot of the denoised PCs of the E4.5 transcripts level dataset

Hide

```
library(cluster)
clust.col <- scater:::.get_palette("tableau10medium")
sil2 <- silhouette(my.clusters2, dist = my.dist)
sil.cols2 <- clust.col[ifelse(sil2[,3] > 0, sil2[,1], sil2[,2])]
sil.cols2 <- sil.cols2[order(-sil2[,1], sil2[,3])]
#png("e45.mRNA_clusters_silhouette.png")
plot(sil2, main = paste(length(unique(my.clusters2)), "clusters"), border=sil.cols2, col=sil.cols
2, do.col.sort=FALSE)
```



Hide

```
#dev.off()
```

Fig. 24 - Barplot of silhouette widths for cells in each cluster for E4.5 transcipts level dataset

Detecting marker transcripts between clusters

Hide

```
markers2 <- findMarkers(sce2, my.clusters2)
marker.set2 <- markers2[["2"]]
top.markers2 <- rownames(marker.set2)[marker.set2$Top <= 10]
```

Define row filters for transcripts of pluripotency genes

Hide

```
nanog <- rownames(sce2)[grep("Nanog_", rownames(sce2))]
esrrb <- rownames(sce2)[grep("Esrrb_", rownames(sce2))]
oct4 <- rownames(sce2)[grep("Pou5f1_", rownames(sce2))]
sox2 <- rownames(sce2)[grep("Sox2_", rownames(sce2))]
pluriset <- c(oct4, sox2, nanog, esrrb)
```

Visualize the expression distribution of cluster 2 top markers together with the pluripotency gene transcripts

Identifying HVGs from the normalized log-expression

Hide

```
hvg.out2 <- var.out2[which(var.out2$FDR <= 0.05 & var.out2$bio >= 0.5),]
hvg.out2 <- hvg.out2[order(hvg.out2$bio, decreasing=TRUE),]
write.table(file="e45_mRNA_HVGs.tsv", hvg.out2, sep="\t", quote=FALSE, col.names=NA)
```

Check if some Nanog transcripts have been included among the HVGs

Hide

```
rownames(hvg.out2)[grep("Nanog_", rownames(hvg.out2))]
```

```
[1] "Nanog_ENSMUST00000012540.4"
```

Identify correlated transcripts using Spearman's ranking

Hide

```
set.seed(100)
var.cor2 <- correlatePairs(sce2, iters=1e8, subset.row=rownames(hvg.out2))
write.table(file="e45_mRNA_cor.tsv", var.cor2, sep="\t", quote=FALSE, row.names=FALSE)
```

Retrieve transcripts correlated with Nanog from correlate pairs

Hide

```
nanog.rows2 <- var.cor2$gene1 == "Nanog_ENSMUST00000012540.4" | var.cor2$gene2 == "Nanog_ENSMUST0
0000012540.4"
var.cor.nanog2 <- var.cor2[nanog.rows2,]
write.table(file="e45_mRNA_cor_nanog.tsv", var.cor.nanog2, sep="\t", quote=FALSE, row.names=FALSE
)
```

Find the top Nanog correlated pairs by significance (FDR <= 5%) and high strength of association (rho > [0.4]).

Hide

```
sig.cor.nanog2 <- var.cor.nanog2$FDR <= 0.05 & abs(var.cor.nanog2$rho) >= 0.4

summary(sig.cor.nanog2)
```

Visualize the expression distribution of the top Nanog correlated transcripts

subset SingleCellExperiment to Cluster 2 (Epiblast)

Hide

```
epi.sce2 <- sce2[,sce2$cluster == 2]

dim(epi.sce2)
```

Hide

```
epi.sce2 <- computeSumFactors(epi.sce2, sizes=c(5, 10, 15, 20, 24))
summary(sizeFactors(epi.sce2))
```

Applying the size factors to normalize gene expression

Hide

```
epi.sce2 <- normalize(epi.sce2)
```

Identify correlated HVGs transcripts in the epiblast using Spearman's ranking

Hide

```
head(epi.var.cor2)
```

```
DataFrame with 6 rows and 6 columns
                         gene1                          gene2               rho         p.value
              FDR   limited
                      <character>                    <character>         <numeric>       <numeric>
        <numeric> <logical>
1    Vamp8_ENSMUST00000059983.9    Vamp8_ENSMUST00000192980.1                   1  1.99999998e-08
0.0315884096841159      TRUE
2 Eif4ebp1_ENSMUST00000210959.1 Eif4ebp1_ENSMUST00000033880.6 0.919130434782609  1.99999998e-08
0.0315884096841159      TRUE
3    Ifi30_ENSMUST00000213938.1    Ifi30_ENSMUST00000229632.1              0.88   9.9999999e-08
0.105294698947053     FALSE
4     Bub3_ENSMUST00000084502.6    Top2a_ENSMUST00000068031.7 0.865217391304348 1.799999982e-07
0.142147843578522     FALSE
5   H2afy_ENSMUST00000016081.12    Ppp4c_ENSMUST00000206570.1 0.847826086956522 5.199999948e-07
0.328519460714805     FALSE
6   Dnmt3b_ENSMUST00000103151.7  Foxred1_ENSMUST00000127996.7 0.842608695652174  6.99999993e-07
0.366425552335744     FALSE
```

Retrieve epiblast transcripts correlated with Nanog from correlate pairs

Hide

```
epi.nanog.rows2 <- epi.var.cor2$gene1 == "Nanog_ENSMUST00000012540.4" | epi.var.cor2$gene2 == "Na
nog_ENSMUST00000012540.4"
epi.var.cor.nanog2 <- epi.var.cor2[epi.nanog.rows2,]
write.table(file="epi.e45_mRNA_cor_nanog.tsv", epi.var.cor.nanog2, sep="\t", quote=FALSE, row.nam
es=FALSE)
head(epi.var.cor2)
```

```
DataFrame with 6 rows and 6 columns
                         gene1                          gene2               rho         p.value
              FDR   limited
                      <character>                    <character>         <numeric>       <numeric>
        <numeric> <logical>
1    Vamp8_ENSMUST00000059983.9    Vamp8_ENSMUST00000192980.1                   1  1.99999998e-08
0.0315884096841159      TRUE
2 Eif4ebp1_ENSMUST00000210959.1 Eif4ebp1_ENSMUST00000033880.6 0.919130434782609  1.99999998e-08
0.0315884096841159      TRUE
3    Ifi30_ENSMUST00000213938.1    Ifi30_ENSMUST00000229632.1              0.88   9.9999999e-08
0.105294698947053     FALSE
4     Bub3_ENSMUST00000084502.6    Top2a_ENSMUST00000068031.7 0.865217391304348 1.799999982e-07
0.142147843578522     FALSE
5   H2afy_ENSMUST00000016081.12    Ppp4c_ENSMUST00000206570.1 0.847826086956522 5.199999948e-07
0.328519460714805     FALSE
6   Dnmt3b_ENSMUST00000103151.7  Foxred1_ENSMUST00000127996.7 0.842608695652174  6.99999993e-07
0.366425552335744     FALSE
```

Find the top Nanog correlated pairs by significance (FDR <= 20%)

Hide

```
epi.sig.cor.nanog2 <- epi.var.cor.nanog2$FDR <= 0.2
summary(epi.sig.cor.nanog2)
```

```
   Mode   FALSE
logical    2513
```

Compute the correlate pair within the Nanog and Esrrb variants

Hide

```
set.seed(100)
pluri.cor <- correlatePairs(sce2, subset.row=c(nanog, esrrb))
sig.pluri.cor <- pluri.cor$FDR <= 0.05
write.table(file="e45_mRNA_Nanog-Esrrb_cor.tsv", pluri.cor[sig.pluri.cor,], sep="\t", quote=FALSE
, row.names=FALSE)
```

Output and discussion

From the preliminary quality control on the sequencing data performed with “FastQC”, no particular issues have been identified, and the overall quality of the reads appeared to be good, except an adapter contamination deriving from the Illumina mate pair libraries constructed using the Nextera protocol. The raw scRNA-seq data have been processed with “Trim Galore” for quality filtering of the reads with a low-quality base calls and the adapter trimming.

Following the QC, in alternative to the classic alignment based tools (such as tophat, STAR, bowtie, HISAT) the allignment free trasciptome quantification has been preferred due to the purpose of this study (involving a differential expression analysis, without necessity to identify novel transcripts). The pseudoallignment tool used “kallisto”, break up reads into k-mers before assigning them to transcripts. This results in a substantial gain in speed compared to the alignment based workflows. The workflows also differ in how the expression abundance is estimated, enabling quantification on transcript level (Everaert et al. 2017).

After the transcriptome index preparation and the abundance quantification, the kallisto output has been imported into the R environmente using “tximport”, generating SingleCellExperiment (sce) files for the Bioconductor analysis pipeline. The analysis output reported here is focused on the peri-implatation stages E3.5 and E4.5. The full dataset of the original study has been subjected to pseudoalignment with the mouse transcriptome and a principal component analysis has been performed, showing that the dataset separates by developmental stages.

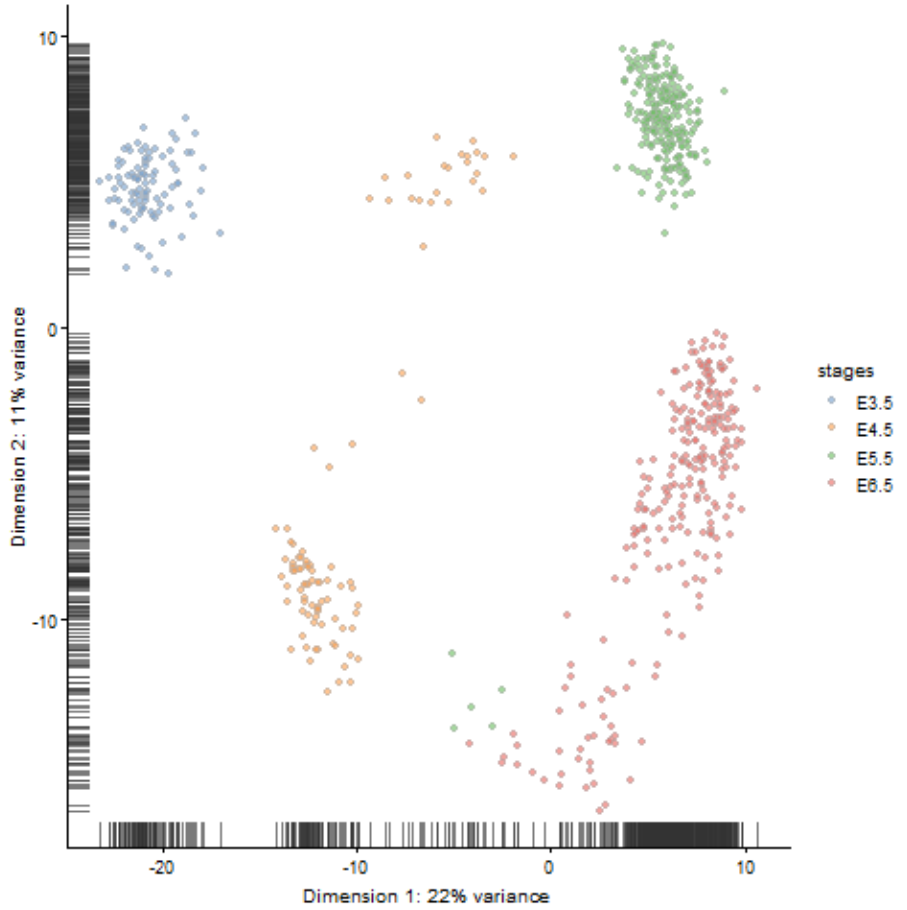


Fig. 25 - PCA plot for 2 dimensions, after QC and normalization of cell-specific biases, of the full dataset in the original study of Mohammed et al. 2017. The cells has been colored by developmental stage.

By colouring differentially the cells on the PCA plot, it is possible to inspect the expression profile associated with gene markers reported in literature (Tam and Loebel 2007) allowing the identification of different lineages.

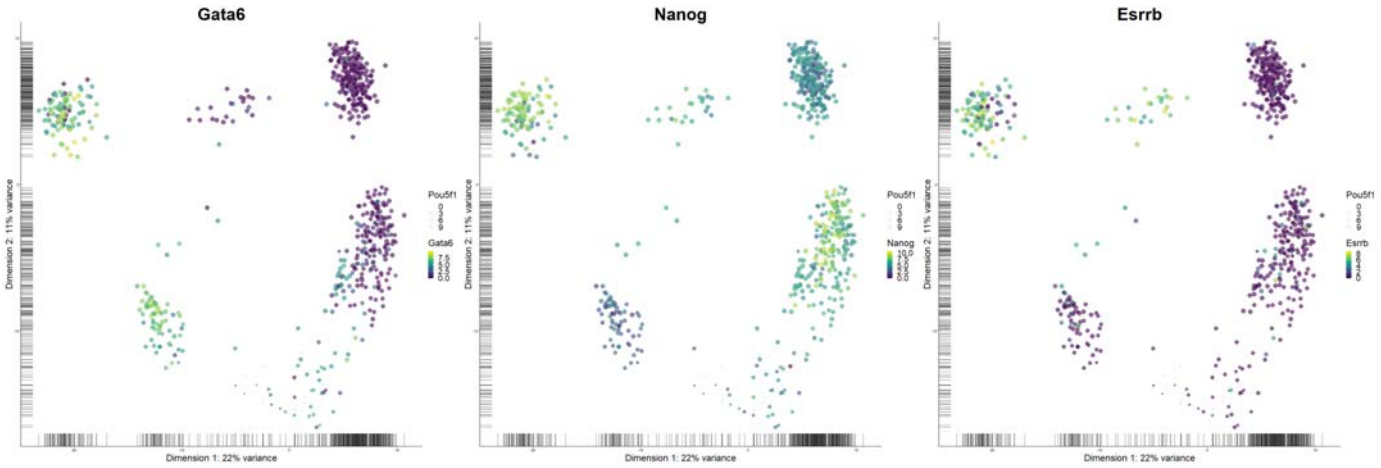


Fig. 26 - PCA plot of the full dataset colored by gene expression levels (logcounts) of selected marker genes - Nanog (ICM/epiblast), Gata6 (PrE/VE), Esrrb (naive pluripotency). Cells are also sized by expression levels of Oct4 (core pluripotency).

Examining the figures 30 and 31, the population of cells at E3.5 appear homogeneous, without a distinct lineage indentities. But colouring some marker genes (Fig. 31), it is clearly visible a separation of two populations with different features, among the pluripotent cells (high Oct4) of the stage E4.5, characterized by opposite expression profiles of Nanog and Gata6, respectively associated to the Epiblast (Epi) and primitive endoderm (PrE) in previous studies (Tam and Loebel 2007). Esrrb expression distribution is concentrated on the stages E3.5 and E4.5 and the expression levels seem to follow the newely forming epiblast cell population.

Single cell consensus clustering (SC3) has been used to explore the data, find a reasonable estimate of the number of clusters and calculate the biological features based on the identified cell clusters:

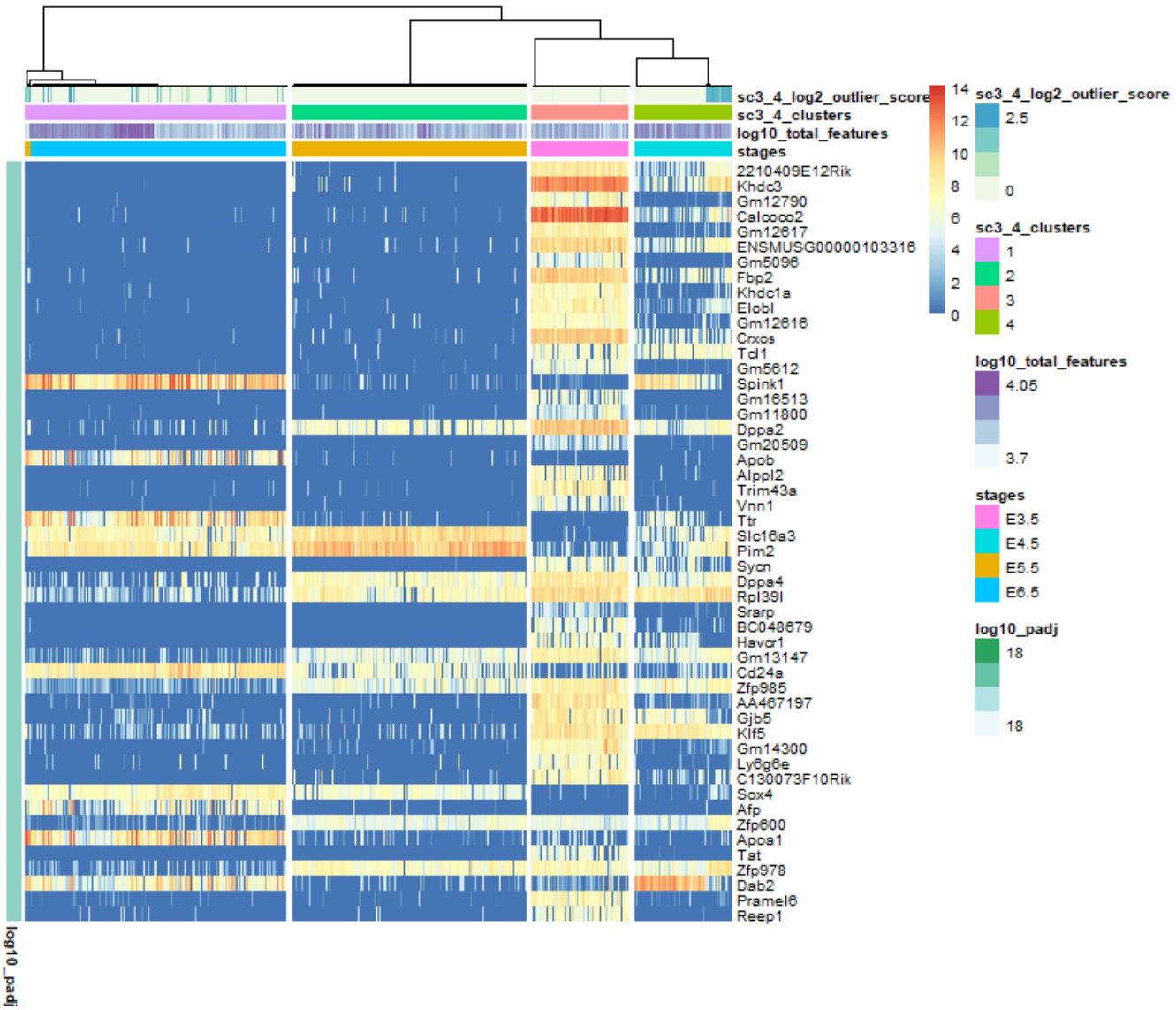


Fig. 27 - Heatmap showing the differential expression between clusters coincident to developmental stages, calculated using non-parametric Kruskal-Wallis test.

After visually examining the consensus matrices generated for a range of clusters between 2 and 6, the optimal separation of 4 clusters has been chosen for downstream analysis, because segregates better with the four developmental stages constituing the dataset. SC3 has been used to create a list of all differentially expressed (DE) genes between the clusters, with adjusted p-values < 0.01 and plots gene expression profiles of the 50 genes with the lowest p-values. Interestingly, in coincidence with the stage E4.5, a subpopulation with strong positive DE for Dab2 is visible. The endocytic adaptor protein Dab2 mediates directional vesicular trafficking required for the genesis of an apical polarity, which is known to be crucial for the sorting and positioning of the PrE cells at the surface of the inner cell mass (ICM) during the embryo implantation (Moore et al. 2014).

Based on the mean expression values of the genes, marker genes for each cluster/stage have been defined.

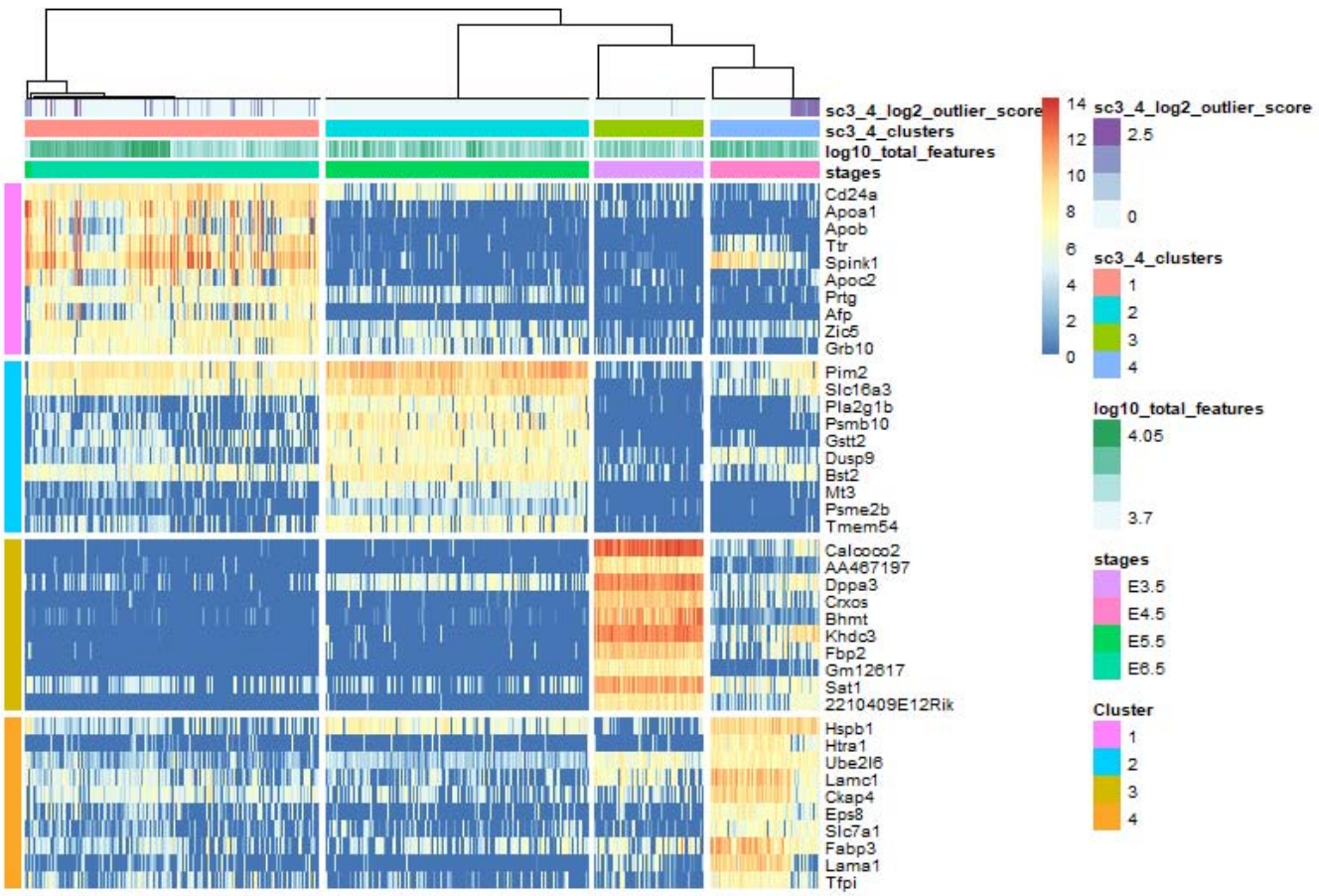


Fig. 28 - Heatmap showing the marker genes identified for each cluster. To find marker genes, for each gene a binary classifier is constructed based on the mean cluster expression values. The classifier prediction is then calculated using the gene expression ranks. The area under the receiver operating characteristic (ROC) curve is used to quantify the accuracy of the prediction. A p-value is assigned to each gene by using the Wilcoxon signed rank test. By default the genes with the area under the ROC curve (AUROC) > 0.85 and with the p-value < 0.01 are selected and the top 10 marker genes of each cluster are visualized in this heatmap.

Notably, an high expression of developmental pluripotency-associated protein 3 (Dppa3) characterizes the cluster 3, correspondent to the preimplantation E3.5 stage, while the cluster 4, correspondent to the post-implantation stage E4.5 express high levels of laminin (Lama1 and Lamc1) required for the attachment, migration and organization of cells into tissues.

Following the gereral characterization of the dataset, already described by the authors (Mohammed et al. 2017), specific SingleCellExperiment files have been generated from the abundance output for the stage E3.5 and E4.5, and the same Bioconductor pipeline (Lun, McCarthy, and Marioni 2018) has been performed on both datasets, in order to detect highly variable genes, significantly correlated genes and subpopulation-specific marker genes characteristic of each stage.

The approach used for dimensionality reduction is the t-stochastic neighbour embedding (tSNE) method (Maaten and Hinton 2008), which showed to be more suitable for capturing the non-linear relationships in the high dimensional space and separating subpopulations by expressing features. A hierarchical clustering on the Euclidean distances between cells was performed, generating a dendrogram that groups together cells with similar expression patterns. Then, clusters have been explicitly defined by applying a dynamic tree cut (Langfelder and Horvath 2008) to the dendrogram.

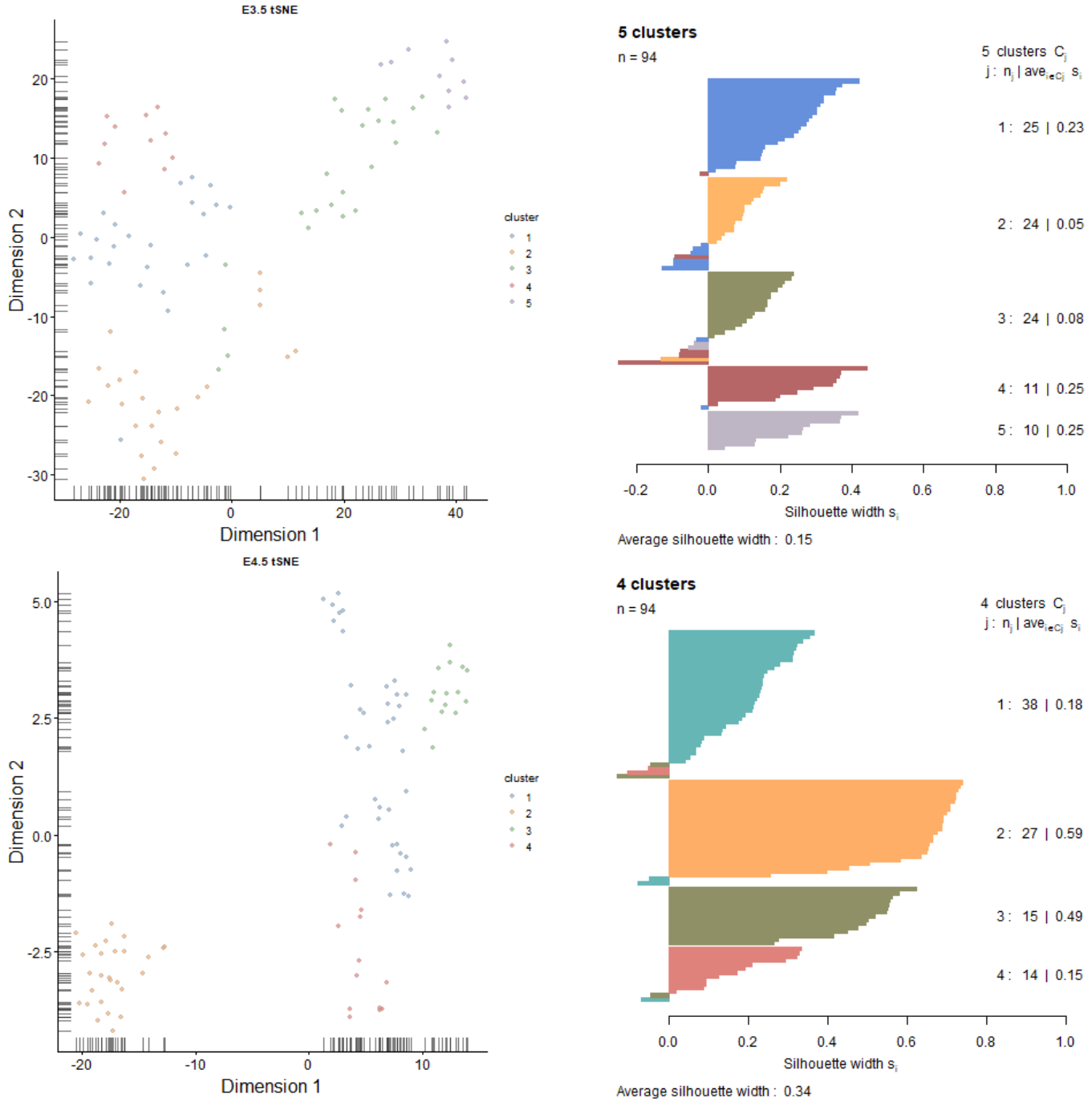
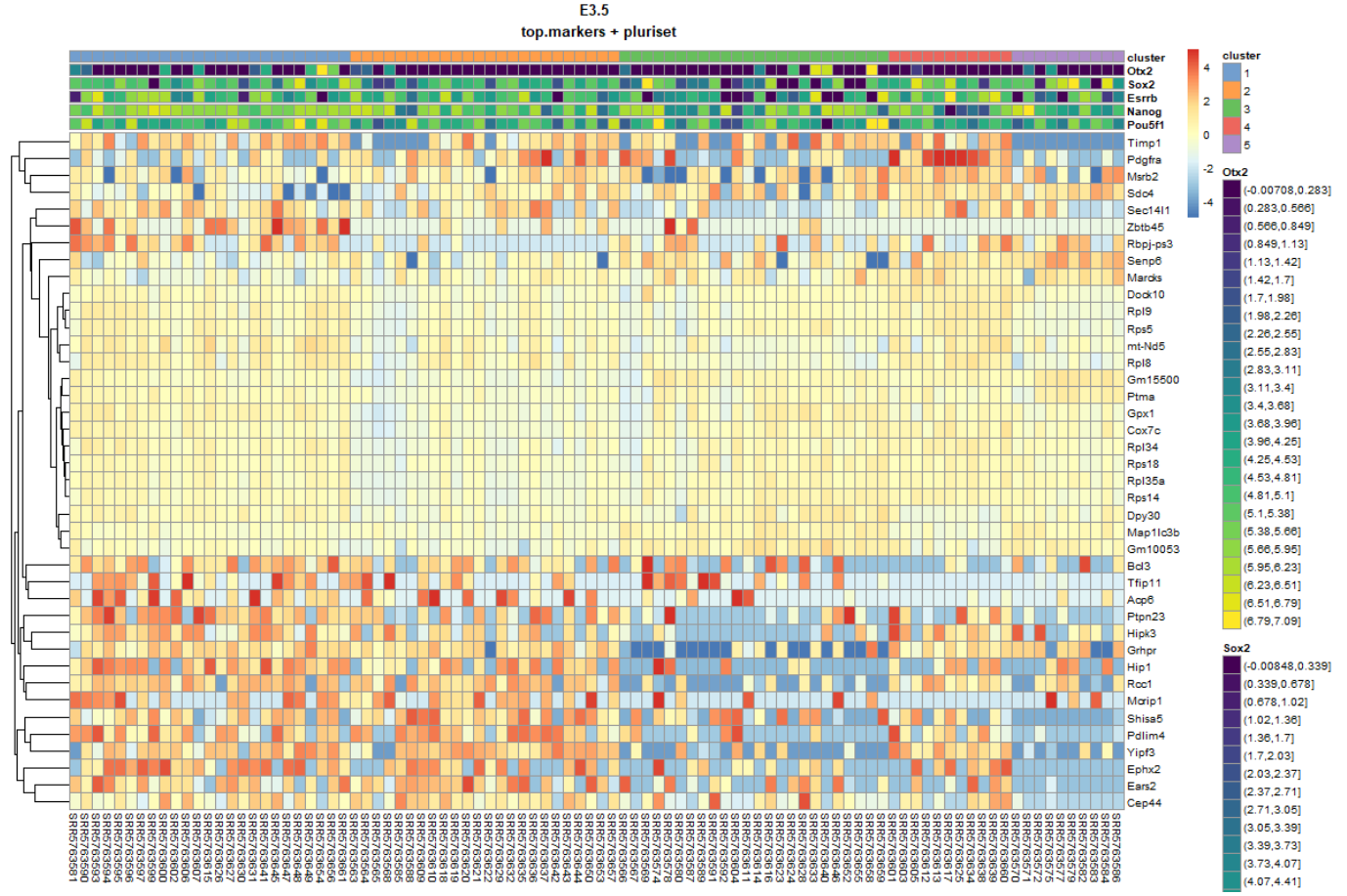


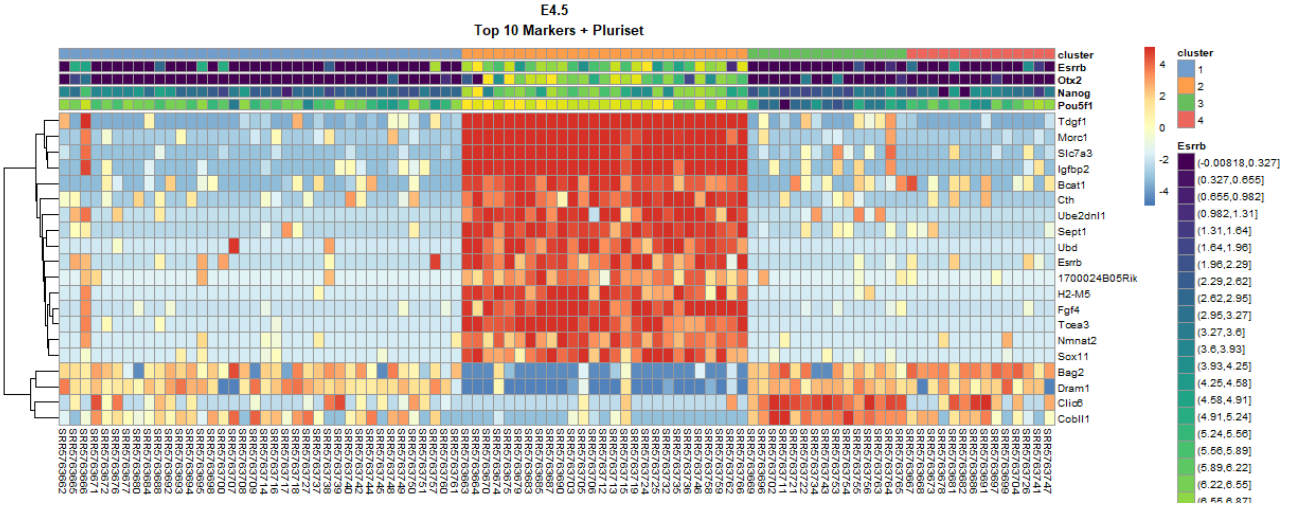
Fig. 29 - tSNE plot of the denoised PCA of the E3.5 dataset (perplexity = 20).

The separatedness of the clusters has been verified using the silhouette width (Fig. 17, Fig. 29) for each cluster marker genes have been identified using limma (Ritchie et al. 2015). Five clusters have been identified on E3.5 but the separatedness is much lower in comparison to the 4 clusters of E4.5 (in particular cluster 2 and 3), so they are not likely to reflect actual populations on E3.5 dataset. The cluster 2 of E4.5 on the other hand, groups a subpopulation apart from the rest of the cells with different features.

For both datasets the clusters have been tested. The top differentially expressed genes are likely to be good candidate markers as they can effectively distinguish between cells in different clusters. The top markers candidates for cluster 1 of E3.5 (which contains the majority of cells and the largest positive silhouette width) and cluster 2 of E4.5, have been chosen to be visualized on a heatmap, together with the expression profile of some pluripotency markers (Oct4, Nanog, Esrrb, Otx2).



30a

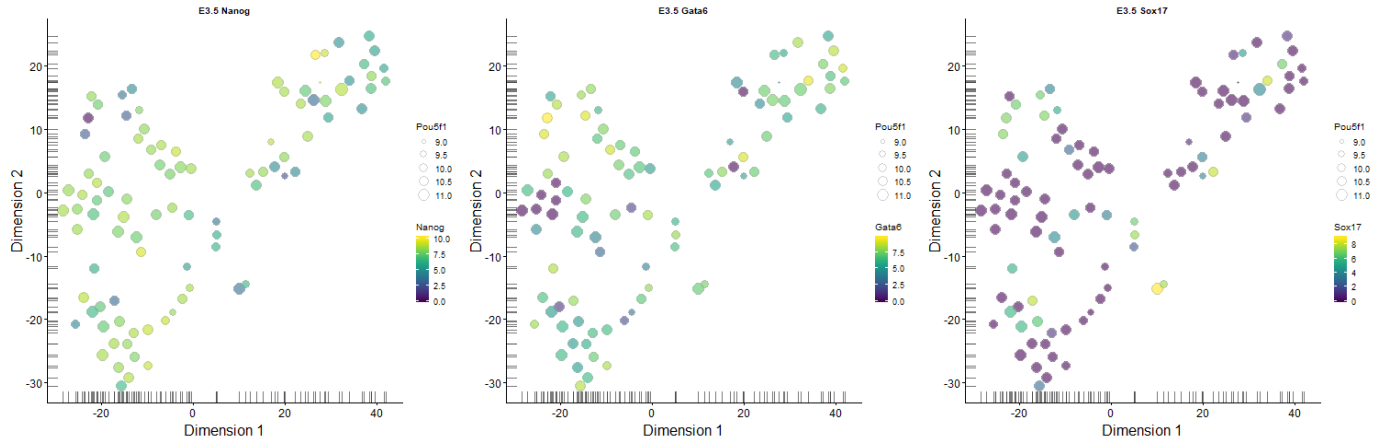


30b

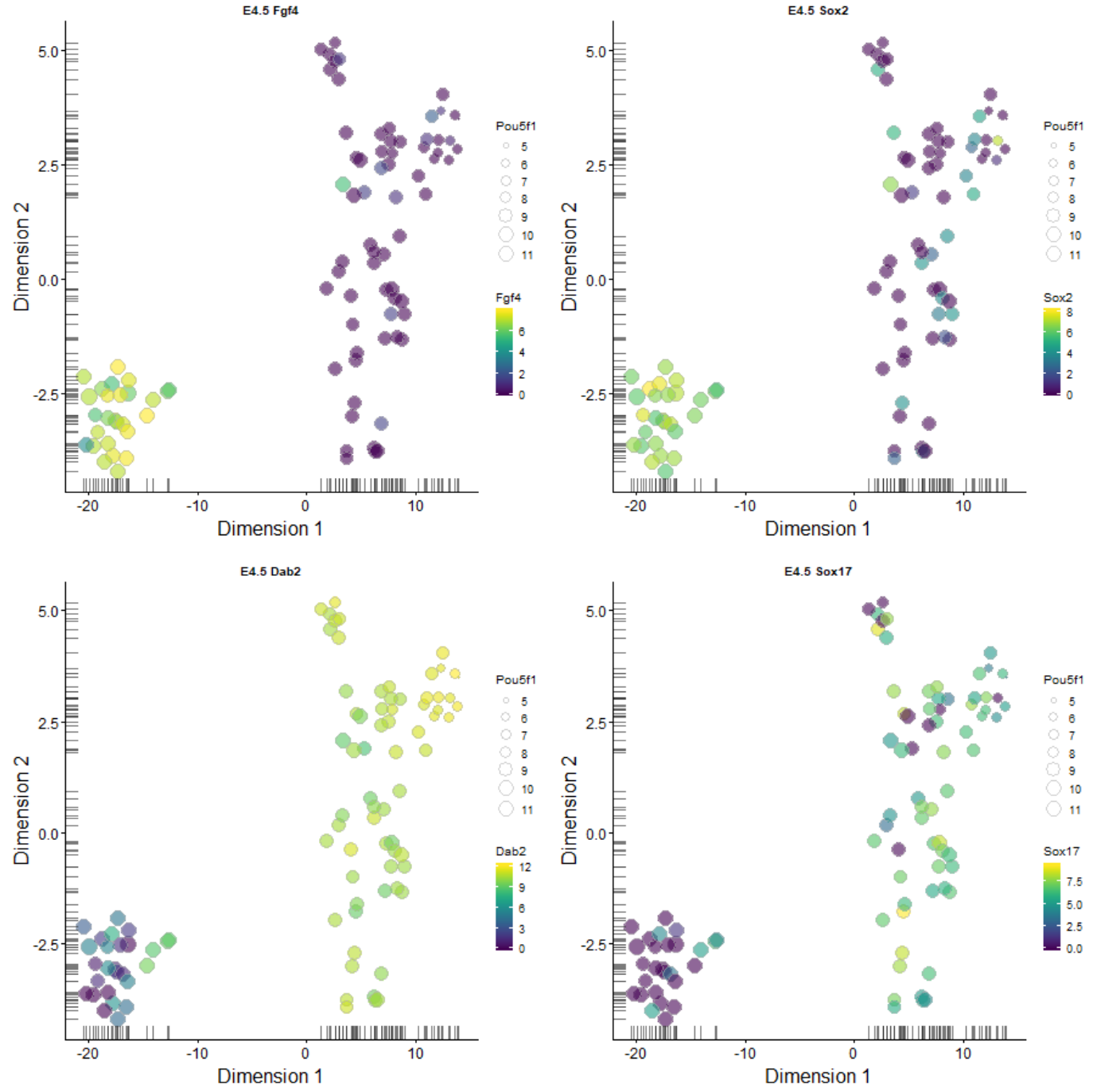
Fig. 30 - Heatmap of mean-centred and normalized log-expression values for the top set of markers for cluster 1 in the E3.5 dataset. Markers of pluripotency and lineage differentiation are also added on top rows.

While no pattern is recognizable on the heatmap for E3.5, the expression profiles for the markers for cluster 2 of E4.5 dataset result strongly associated to a subpopulation of cells expressing typical features of the epiblast phenotype.

This is in accordance to the biological knowledge on the embryo development. As reported in the original study, at stage E3.5 transcriptional variability occurring in the inner cell mass (ICM) in the absence of cell-type substructure indicates the existence of uncoordinated transcriptional heterogeneity or transcriptional noise (Mohammed et al. 2017). But this also confirms previous findings that genes specific of PrE (Gata6, Sox17) and epiblast (Nanog, Sox2) initially result co-expressed in cells within the E3.5 ICM before becoming restricted to specific cell types by E4.5 (Ohnishi et al. 2014).



31a



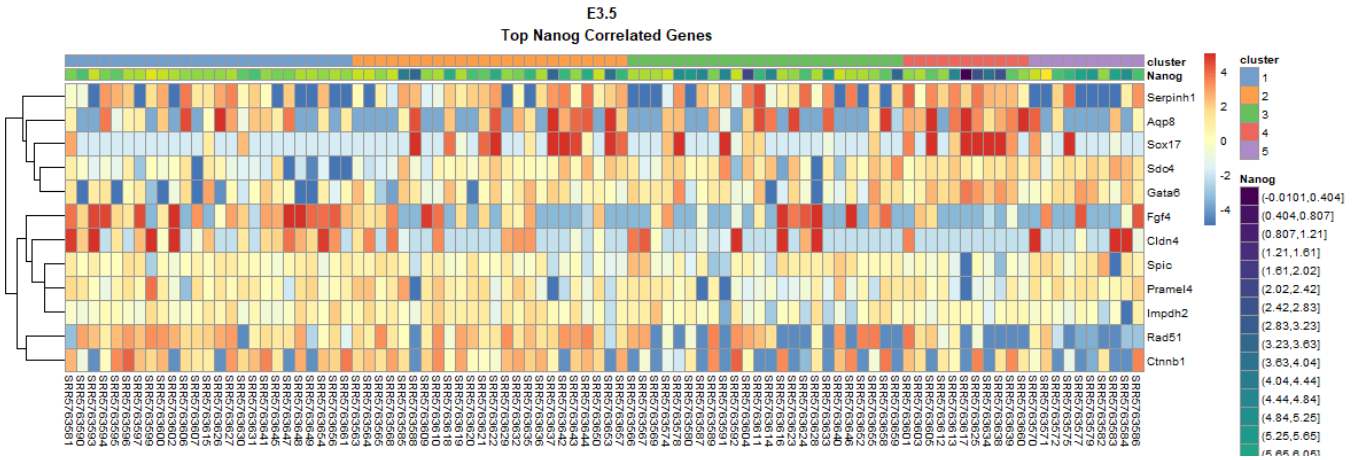
31b

Fig. 31 - t-SNE plot comparison between E3.5 and E4.5 datasets showing the expression distribution of epiblast and primitive endoderm markers across the cells.

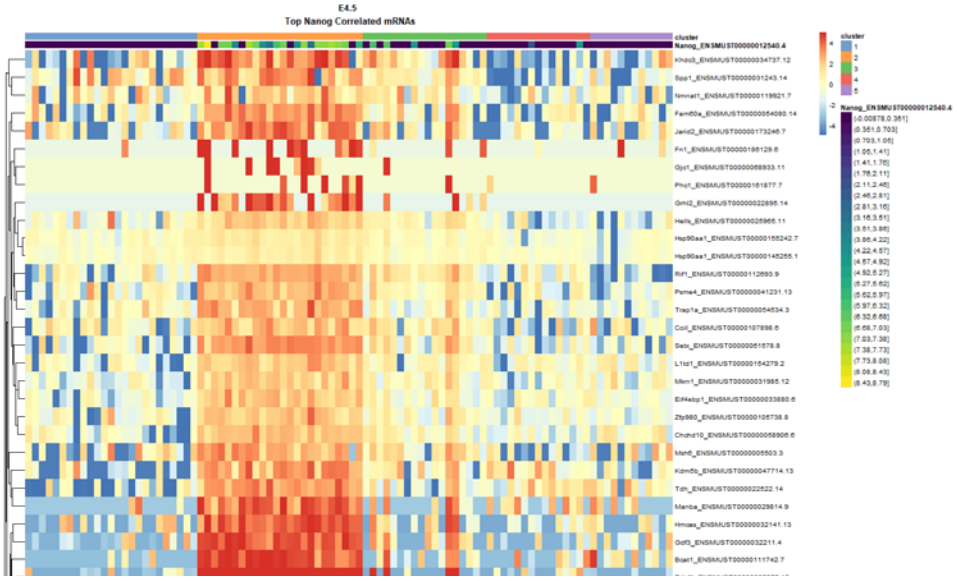
It is also known by studies exploring the cooperation of transcription factors in PrE induction in ESCs, that Sox17/Oct4 partner to co-select specific target genes during commitment to primitive endoderm (Aksoy et al. 2013). A model has been proposed. In pluripotent cells, Oct4 and Sox2 expression levels being high, both factors cooperate and target speci???cally the canonical motif to regulate the expression of genes involved in self-renewal and pluripotency (e.g., Nanog). When these cells are subjected to an endodermal differentiation signal such as FGF4 within the ICM, Sox17 levels increase leading to a switch of Oct4 from an interaction with Sox2 to an interaction with Sox17, and thereby targets speci???c genes containing a compressed motif to trigger the endodermal expression program. The expression profile of these genes across the stages E3.5 and E4.5 in this dataset seems to confirm this model (Fig. 31), supporting the hypotesis that the cluster 2 identified by tSNE at stage E4.5 correspond to the epiblast cell type.

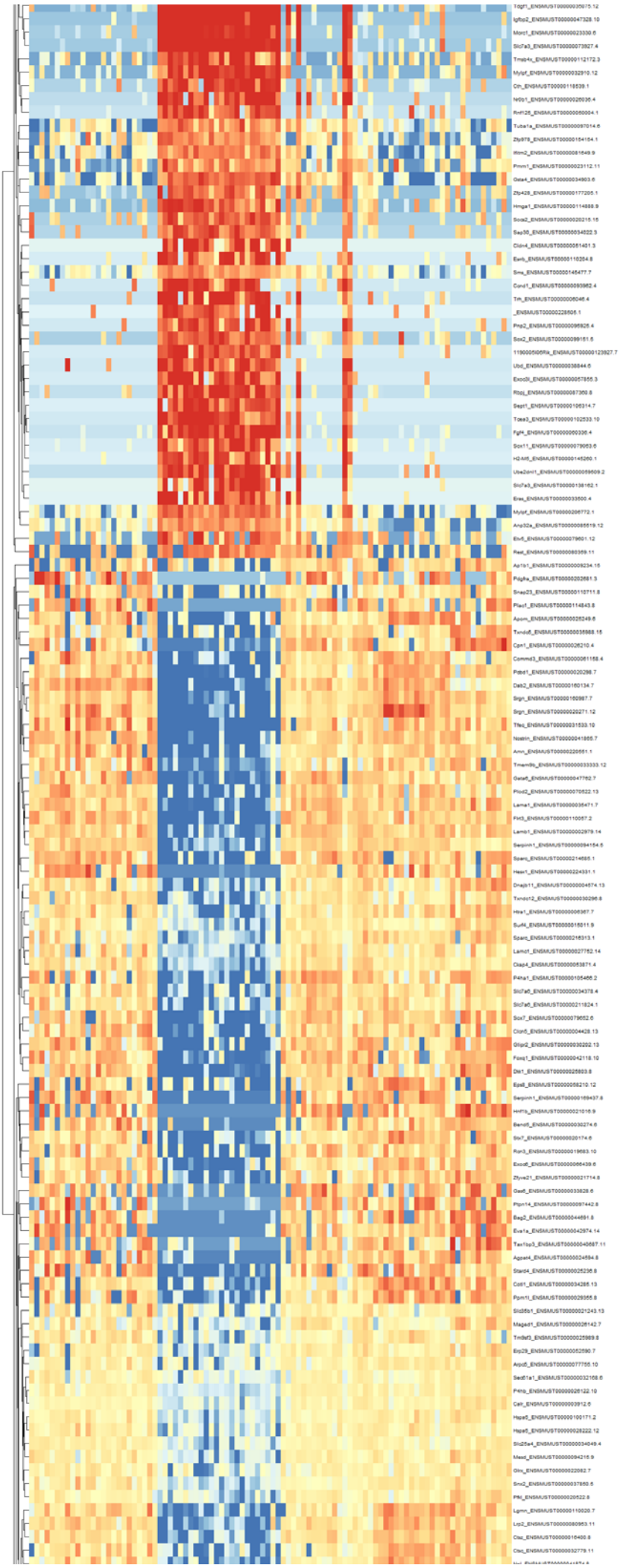
It's also important to notice that Esrrb is included among the top markers of cluster 2 showing an expression profile very similar to Nanog and others pluripotency genes on the dataset. Among the Nanog target genes identified by using a Nanog knockout system on mouse ESCs, Esrrb showed the strongest transcriptional induction. Moreover the existance of partial funcional overlap between Nanog and Esrrb has been proposed (Festuccia et al. 2012).

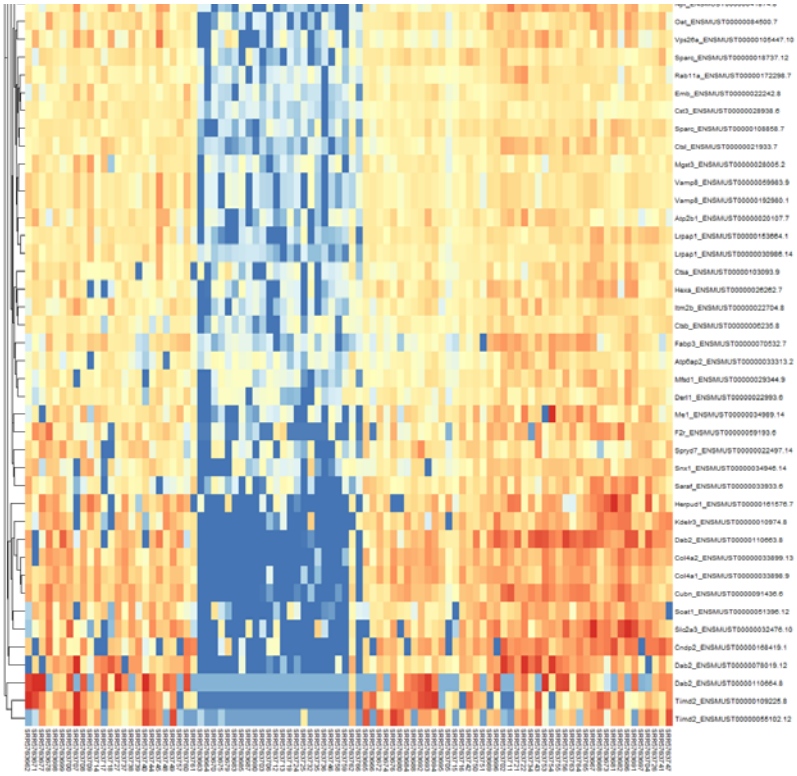
On the next step the attention has been focused on the genes that are driving heterogeneity across the population of cells, by identifying highly variable genes (HVGs): those genes with the largest biological components of the variance in expression. In order to distinguish between HVGs caused by random noise and those involved in driving systematic differences between subpopulations, gene pairs with significantly positive or negative correlation have been quantified by computing Spearman's rho(rho = 1 means a perfect positive correlation and the value rho = -1 means a perfect negataive correlation), which accommodates non-linear relationships in the expression values, therefore has been preferred to the classical Pearson's correlation coefficients. On the E3.5 dataset only 12 out of 2601 significantly correlated genes pairs (FDR <= 5%) include Nanog, while 341 out of 119754 genes resulted significantly correlated to Nanog in the E4.5 dataset.



32a




Tdgf1_ENSMUST00000035075.12
Igfbp2_ENSMUST00000047328.10
Morc1_ENSMUST00000023330.6
Slc7a3_ENSMUST00000073927.4
Tmsb4x_ENSMUST00000112172.3
Mylpf_ENSMUST00000032910.12
Cth_ENSMUST00000118539.1
Nr0b1_ENSMUST00000026036.4
Rnf125_ENSMUST00000050004.1
Tuba1a_ENSMUST00000097014.6
Zfp978_ENSMUST00000154154.1
Ifitm2_ENSMUST00000081549.9
Pmm1_ENSMUST00000023112.11
Gata4_ENSMUST00000034903.6
Zfp428_ENSMUST00000177205.1
Hmga1_ENSMUST00000114888.9
Sox2_ENSMUST00000020215.15
Sap30_ENSMUST00000034022.3
Cldn4_ENSMUST00000051451.3
Ear6_ENSMUST00000110264.8
Sns_ENSMUST00000145477.7
Cond1_ENSMUST00000093962.4
Tfr_ENSMUST00000008046.4
_ENSMUST00000228505.1
Pnp2_ENSMUST00000096826.4
Sox2_ENSMUST00000099151.6
1190005I06Rik_ENSMUST00000123927.7
Ubd_ENSMUST00000038844.6
Exoc3l_ENSMUST00000057855.3
Rbpj_ENSMUST00000087360.8
Sept1_ENSMUST00000106314.7
Tceb3_ENSMUST00000102533.10
Fgf4_ENSMUST00000060336.4
Sox11_ENSMUST00000079063.6
H2-Ab1_ENSMUST00000146260.1
Ube2dnl1_ENSMUST00000059609.2
Slc7a3_ENSMUST00000138162.1
Eras_ENSMUST00000033500.4
Mylpf_ENSMUST00000206772.1
Arp32a_ENSMUST00000085519.12
Elvl5_ENSMUST00000079601.12
Rest_ENSMUST00000093359.11
Ap1b1_ENSMUST00000009234.15
Pdgfra_ENSMUST00000202681.3
Snap23_ENSMUST00000110711.8
Plac1_ENSMUST00000114843.8
Apom_ENSMUST00000026249.6
Tyndc6_ENSMUST00000035988.15
Cpn1_ENSMUST00000026210.4
Commd3_ENSMUST00000061168.4
Pdbd1_ENSMUST00000020286.7
Dab2_ENSMUST00000160134.7
Srgn_ENSMUST00000160967.7
Srgn_ENSMUST00000029271.12
Tfec_ENSMUST00000031533.10
Nostrin_ENSMUST00000041966.7
Anxn_ENSMUST00000220651.1
Tmem8b_ENSMUST00000033333.12
Gata6_ENSMUST00000047762.7
Plod2_ENSMUST00000070522.13
Lama1_ENSMUST00000035471.7
Fkh3_ENSMUST00000110067.2
Lamb1_ENSMUST00000002979.14
Serpinh1_ENSMUST00000094154.5
Sparc_ENSMUST00000214685.1
Hesx1_ENSMUST00000224331.1
Dnajb11_ENSMUST00000004574.13
Tyndc12_ENSMUST00000030296.8
Hira1_ENSMUST00000006367.7
Sulf4_ENSMUST00000015011.9
Sparc_ENSMUST00000216313.1
Lamc1_ENSMUST00000027752.14
Ckap4_ENSMUST00000053871.4
P4ha1_ENSMUST00000105466.2
Slc7a5_ENSMUST00000034378.4
Slc7a5_ENSMUST00000211824.1
Sox7_ENSMUST00000079652.6
Clcn5_ENSMUST00000004428.13
Glipr2_ENSMUST00000030202.13
Fosq1_ENSMUST00000042118.10
Dlx1_ENSMUST00000025803.8
Eps8_ENSMUST00000056210.12
Serpinh1_ENSMUST00000169437.8
Hnf1b_ENSMUST00000021016.9
Bend5_ENSMUST00000030274.6
Six7_ENSMUST00000020174.6
Ror3_ENSMUST00000019683.10
Exoc6_ENSMUST00000066439.6
Zfyve21_ENSMUST00000021714.8
Gas6_ENSMUST00000033828.6
Ptpn14_ENSMUST00000097442.8
Bag2_ENSMUST00000044691.8
Eva1a_ENSMUST00000042974.14
Tax1bp3_ENSMUST00000040687.11
Agpat4_ENSMUST00000024594.8
Star94_ENSMUST00000025236.8
Col1_ENSMUST00000034285.13
Ppm1l_ENSMUST00000029355.8
Slc35b1_ENSMUST00000021243.13
Magad1_ENSMUST00000026142.7
Tmbd3_ENSMUST00000025989.8
Eip29_ENSMUST00000052590.7
Aqp5_ENSMUST00000077755.10
Sec61a1_ENSMUST00000032108.6
P4ha_ENSMUST00000026122.10
Cell_ENSMUST00000003912.6
Hspa5_ENSMUST00000100171.2
Hspa5_ENSMUST00000028222.12
Slc25a4_ENSMUST00000034049.4
Mesd_ENSMUST00000094215.9
Glrx_ENSMUST00000022082.7
Sns2_ENSMUST00000037850.5
Phl_ENSMUST00000020522.8
Lgmn_ENSMUST00000110020.7
Lrp2_ENSMUST00000080953.11
Ctsc_ENSMUST00000016400.8
Ctsc_ENSMUST00000032779.11



32b

Fig. 32 - Heatmap of mean-centred and normalized log-expression values for the top HVGs correlated to Nanog in the E3.5 and E4.5 datasets

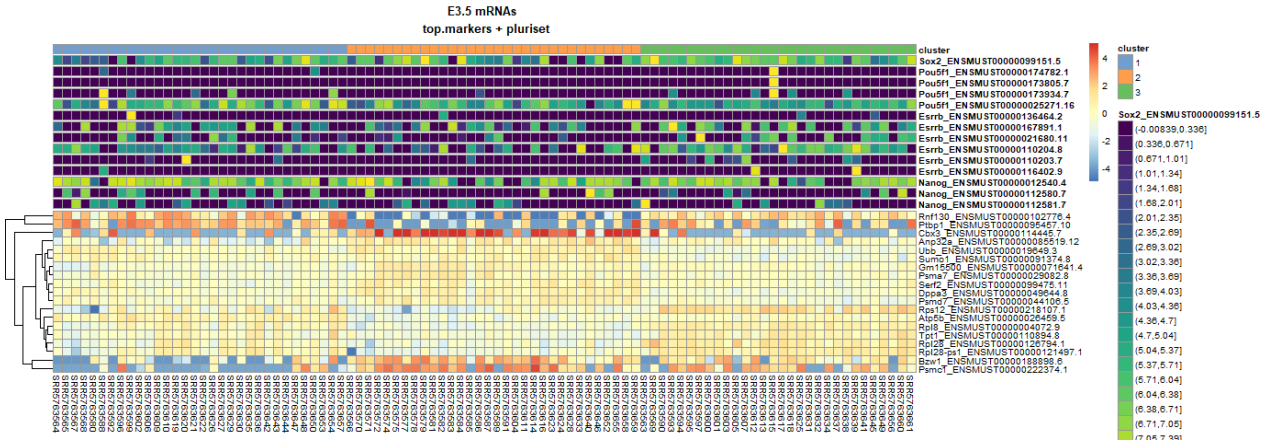
Among the list of genes positively correlated to Nanog at stage E3.5 it's important to report Rad51 (rho = 0.42), a key component of the DNA double-strand breaks repair through homologous recombination and has been shown to increase the reprogramming efficiency during the generation of induced pluripotent stem cells by facilitating mesenchymal-to-epithelial transition during the early phase of the reprogramming process (J.-Y. Lee et al. (2016)), and Fgf4 (rho = 0.39), which contributes to the ICM lineage segregation (Ohnishi et al. 2014). Notably, Sox17 and Gata6 results negatively correlated to Nanog (rho = -0.41 / -0.36), anticipating the emergence of these two TFs among the PrE precursors at stage E4.5.

The same correletion pattern can in fact been found on the list of Nanog correlatet genes at stage E4.5, with Gata6 showing a stronger negative association (rho = -0.54) on conjunction with specific PrE phenotype markers like Dab2 (rho = -0.71) and Sox17 (rho = -0.44). Observing the heatmap on figure 32b it is possible to spot a group of genes with high positive correlation to Nanog overlapping with the top marker genes used to define the cluster 2 on the previous step (Tdgf1, Morc1, Slc7a3, Igfbp2) attributable to the epiblast subpopulation. The presence of Esrrb within the Nanog correlated genes (rho = 0.57) and its coexpression with Nanog in the epiblast subpopulation confirms the functional overlap observed in the ESCs model (Festuccia et al. 2012).

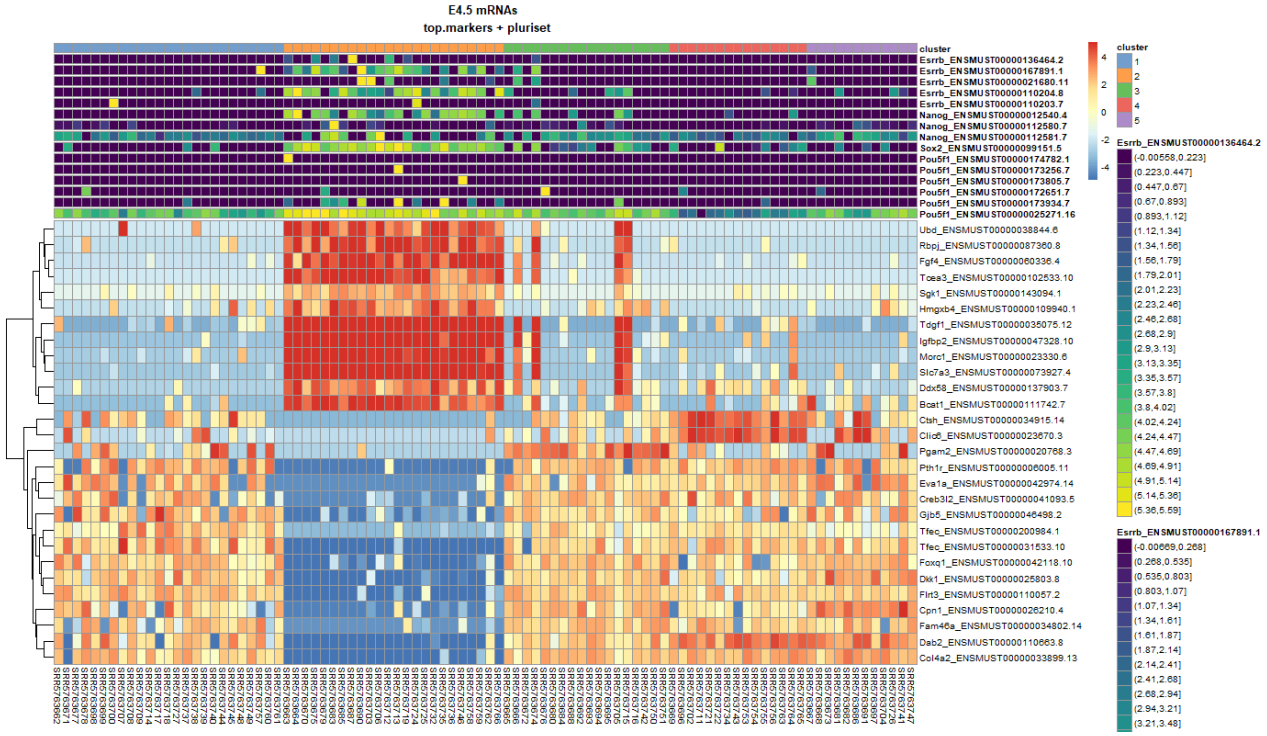
It's also curious to notice at this stage, a positive correlation between Nanog and Otx2 (rho = 0.49) showing a similar expression profile characterized by high levels in the epiblast subpopulation (see Fig. 30b). Nanog is a direct target of Otx2 and previous studies suggested how Otx2 regulation of Nanog contributes to ICM differentiation of the epiblast, observing that stage E4.5 is characterized by contemporary and frequently complementary expression of Nanog and Otx2, resembling the identity of ESCs cultured in LIF/FBS (Acampora et al. 2016), where through mutual antagonism, these TFs specify the heterogeneous identity of ESCs predisposing them for optimal response to naive or primed inducing factors (Acampora et al. 2017).

More than 300 genes resulted significantly correlated to Nanog at stage E4.5. In order to understand which of these genes is driving the specification of the epiblast, the downstream analyais has been focused only on the cluster 2, subsetting the dataset to include only cells showing elevated expression for Oct4 and others naive pluripotency markers. However the Searman's rank correlation performed on this subset produced correlated a list of gene pair with a low significance (FDR greater than 35%) which cannot lead to strong biological conclusions.

At the end of the study a transcription level comparison has been berformed, without summarizing the expression at gene level, in order to get some insights into the distribution of expression profile among the trascripts of interesting genes.



33a



33b

Fig. 33 - Heatmap of mean-centred and normalized log-expression values for the top set of markers transcripts for E3.5 and E4.5 dataset. Marker transcripts of pluripotency and lineage differentiation are also added on top rows.

Surprisingly, many alternative spliced variants for the naive pluripotency genes can be observed with this approach, as well as their differential expression profile. Six protein-coding transcripts were detected for Oct4 but only one able of biological function (ENSMUST00000025271.16).

Nanog includes 3 protein-coding transcripts. One of them is results highly expressed at stage E3.5, the other two variants have been characterized to show attenuated capacities for self-renewal and pluripotency in ESCs (Das, Jena, and Levasseur 2011). However two Nanog isoforms (ENSMUST00000012540.4 and ENSMUST00000012540.4) curiously showed a complementary expression profile at stage E4.5 for this dataset (Fig. 33b) which is not considered during the gene level analysis.

The mouse Esrrb gene has six coding exons, with evidence for six alternatively spliced Esrrb mRNAs in the ENSEMBL EST databases, but one of them is not protein coding (ENSMUST00000136464.2). the abundance for all the six isoforms have been quantified by kallisto and the most abundant transcripts on the E4.5 stage is Esrrb-

203 (ENSMUST00000110204.8, total logcounts = 170.7892) followed by Esrrb-206 (ENSMUST00000167891.1, total logcounts = 104.4746).

To understand the biological relationship among these alternative transcripts, the Spearman's rho has been computed to find correlate pairs on the E4.5 dataset.

gene1	gene2	rho	p.value	FDR	limited
Nanog_ENSMUST00000112581.7	Nanog_ENSMUST00000112580.7	0.8539898	0.000002	0.0000560	TRUE
Nanog_ENSMUST00000112580.7	Nanog_ENSMUST00000012540.4	-0.4036484	0.000076	0.0009100	FALSE
Nanog_ENSMUST00000012540.4	Esrrb_ENSMUST00000110204.8	0.3988224	0.000102	0.0009100	FALSE
Esrrb_ENSMUST00000110204.8	Esrrb_ENSMUST00000167891.1	0.3919734	0.000130	0.0009100	FALSE
Nanog_ENSMUST00000012540.4	Esrrb_ENSMUST00000167891.1	0.3686378	0.000338	0.0017733	FALSE
Nanog_ENSMUST00000112581.7	Nanog_ENSMUST00000012540.4	-0.3643608	0.000380	0.0017733	FALSE
Nanog_ENSMUST00000112580.7	Esrrb_ENSMUST00000167891.1	-0.2687209	0.009362	0.0374480	FALSE

Fig. 34 - Spearman's rank correlate pairs computed for Nanog and Esrrb variants within the E4.5 dataset.

Conclusions

On multiple studies ChIP-seq has been used to map the locations of speci???c TFs considerred crucial for the transcriptional regulatory networks in embryonic stem cells. These factors are known to play different roles in ES-cell biology as components of the LIF and BMP signaling pathways, self-renewal regulators, and key reprogramming factors and a combinatorial control of transcription factors has been observer (X. Chen et al. 2008, (**???**)). Examining the binding pro???les, has been found that a subset of binding sites was bound by many of these TFs, and by clustered the peak sites was possible to define multiple transcription factor-binding loci (MTL). In particular a Nanog-Oct4-Sox2-speci???c MTL has been characterized, and exhibits features of enhanceosomes by enhancing transcription from a distance.

Other studies analyzed the effect of acute depletion of Oct4 or Nanog, in order to understand the mechanisms underlying the transcriptional modulation of the naive pluripotency of ESCs (Hall et al. 2009),(Festuccia et al. 2012).

Less studies however have been carried out on the embryo.

The list of Nanog correlated genes identified in this single-cell dataset for the stages E3.5 and E4.5 of the mouse embryo development, will be used for a comparison study to a set of 64 genes responding to Nanog identified using an ESCs Nanog -/- model (Festuccia et al. 2012). The Spearman's rank correlation coefficient assigned to each gene will be used to discover the strength of a link between two sets of data. As a control for this comparison the expression variation of Esrrb in correlation with Nanog will be used as reference, being Esrrb reported experimentally to have the strongest transcriptional induction as a result of Nanog binding (Festuccia et al. 2012).

If the in-vitro ESCs Nanog -/- model will be proven to be consistent with the embryo expression analysis reported here, this study will hopefully lead to a better understanding of the pluripotency gene regulatory network.

Bibliography

Acampora, Dario, Luca Giovanni Di Giovannantonio, Arcomaria Garofalo, Vincenzo Nigro, Daniela Omodei, Alessia Lombardi, Jingchao Zhang, Ian Chambers, and Antonio Simeone. 2017. "Functional Antagonism Between OTX2 and NANOG Specifies a Spectrum of Heterogeneous Identities in Embryonic Stem Cells." *Stem Cell Reports* 9 (5): 1642–59. doi:10.1016/j.stemcr.2017.09.019 (https://doi.org/10.1016/j.stemcr.2017.09.019).

Acampora, Dario, Daniela Omodei, Giuseppe Petrosino, Arcomaria Garofalo, Marco Savarese, Vincenzo Nigro, Luca Giovanni Di Giovannantonio, Vincenzo Mercadante, and Antonio Simeone. 2016. “Loss of the Otx2-Binding Site in the Nanog Promoter Affects the Integrity of Embryonic Stem Cell Subtypes and Specification of Inner Cell Mass-Derived Epiblast.” *Cell Reports* 15 (12): 2651–64. doi:10.1016/j.celrep.2016.05.041 (https://doi.org/10.1016/j.celrep.2016.05.041).

Aksoy, Irene, Ralf Jauch, Jiaxuan Chen, Mateusz Dyla, Ushashree Divakar, Gireesh K. Bogu, Roy Teo, et al. 2013. “Oct4 Switches Partnering from Sox2 to Sox17 to Reinterpret the Enhancer Code and Specify Endoderm.” *The EMBO Journal* 32 (7): 938–53. doi:10.1038/emboj.2013.31 (https://doi.org/10.1038/emboj.2013.31).

Andrews, Simon. 2018. “FastQC - a Quality Control Tool for High Throughput Sequence Data. Babraham Bioinformatics.” January 10. http://www.bioinformatics.babraham.ac.uk/projects/fastqc/ (http://www.bioinformatics.babraham.ac.uk/projects/fastqc/).

Boyer, Laurie A., Tong Ihn Lee, Megan F. Cole, Sarah E. Johnstone, Stuart S. Levine, Jacob P. Zucker, Matthew G. Guenther, et al. 2005. “Core Transcriptional Regulatory Circuitry in Human Embryonic Stem Cells.” *Cell* 122 (6): 947–56. doi:10.1016/j.cell.2005.08.020 (https://doi.org/10.1016/j.cell.2005.08.020).

Bray, Nicolas L., Harold Pimentel, Páll Melsted, and Lior Pachter. 2016. “Near-Optimal Probabilistic RNA-Seq Quantification.” *Nature Biotechnology* 34 (5): 525–27. doi:10.1038/nbt.3519 (https://doi.org/10.1038/nbt.3519).

Chambers, Ian, Douglas Colby, Morag Robertson, Jennifer Nichols, Sonia Lee, Susan Tweedie, and Austin Smith. 2003. “Functional Expression Cloning of Nanog, a Pluripotency Sustaining Factor in Embryonic Stem Cells.” *Cell* 113 (5): 643–55. doi:10.1016/S0092-8674(03)00392-1 (https://doi.org/10.1016/S0092-8674(03)00392-1).

Chambers, Ian, Jose Silva, Douglas Colby, Jennifer Nichols, Bianca Nijmeijer, Morag Robertson, Jan Vrana, Ken Jones, Lars Grotewold, and Austin Smith. 2007. “Nanog Safeguards Pluripotency and Mediates Germline Development.” *Nature* 450 (7173): 1230–4. doi:10.1038/nature06403 (https://doi.org/10.1038/nature06403).

Chen, Xi, Han Xu, Ping Yuan, Fang Fang, Mikael Huss, Vinsensius B. Vega, Eleanor Wong, et al. 2008. “Integration of External Signaling Pathways with the Core Transcriptional Network in Embryonic Stem Cells.” *Cell* 133 (6): 1106–17. doi:10.1016/j.cell.2008.04.043 (https://doi.org/10.1016/j.cell.2008.04.043).

Das, Satyabrata, Snehalata Jena, and Dana N. Levasseur. 2011. “Alternative Splicing Produces Nanog Protein Variants with Different Capacities for Self-Renewal and Pluripotency in Embryonic Stem Cells.” *Journal of Biological Chemistry* 286 (49): 42690–42703. doi:10.1074/jbc.M111.290189 (https://doi.org/10.1074/jbc.M111.290189).

Everaert, Celine, Manuel Luypaert, Jesper L. V. Maag, Quek Xiu Cheng, Marcel E. Dinger, Jan Hellemans, and Pieter Mestdagh. 2017. “Benchmarking of RNA-Sequencing Analysis Workflows Using Whole-Transcriptome RT-qPCR Expression Data.” *Scientific Reports* 7 (1): 1559. doi:10.1038/s41598-017-01617-3 (https://doi.org/10.1038/s41598-017-01617-3).

Ewels, Philip, Måns Magnusson, Sverker Lundin, and Max Käller. 2016. “MultiQC: Summarize Analysis Results for Multiple Tools and Samples in a Single Report.” *Bioinformatics* 32 (19): 3047–8. doi:10.1093/bioinformatics/btw354 (https://doi.org/10.1093/bioinformatics/btw354).

Festuccia, Nicola, Rodrigo Osorno, Florian Halbritter, Violetta Karwacki-Neisius, Pablo Navarro, Douglas Colby, Frederick Wong, Adam Yates, Simon R. Tomlinson, and Ian Chambers. 2012. “Esrrb Is a Direct Nanog Target Gene That Can Substitute for Nanog Function in Pluripotent Cells.” *Cell Stem Cell* 11 (4): 477–90. doi:10.1016/j.stem.2012.08.002 (https://doi.org/10.1016/j.stem.2012.08.002).

Grüning, Björn, Ryan Dale, Andreas Sjödin, Brad A. Chapman, Jillian Rowe, Christopher H. Tomkins-Tinch, Renan Valieris, and Johannes Köster. 2018. “Bioconda: Sustainable and Comprehensive Software Distribution for the Life Sciences.” *Nature Methods* 15 (7): 475–76. doi:10.1038/s41592-018-0046-7 (https://doi.org/10.1038/s41592-018-0046-7).

Hall, John, Ge Guo, Jason Wray, Isobel Eyres, Jennifer Nichols, Lars Grotewold, Sofia Morfopoulou, et al. 2009. “Oct4 and LIF/Stat3 Additively Induce Krüppel Factors to Sustain Embryonic Stem Cell Self-Renewal.” *Cell Stem Cell* 5 (6): 597–609. doi:10.1016/j.stem.2009.11.003 (https://doi.org/10.1016/j.stem.2009.11.003).

Huber, Wolfgang, Vincent J. Carey, Robert Gentleman, Simon Anders, Marc Carlson, Benilton S. Carvalho, Hector Corrada Bravo, et al. 2015. “Orchestrating High-Throughput Genomic Analysis with Bioconductor.” *Nature Methods* 12 (2): 115–21. doi:10.1038/nmeth.3252 (https://doi.org/10.1038/nmeth.3252).

Ilicic, Tomislav, Jong Kyoung Kim, Aleksandra A. Kolodziejczyk, Frederik Otzen Bagger, Davis James McCarthy, John C. Marioni, and Sarah A. Teichmann. 2016. “Classification of Low Quality Cells from Single-Cell RNA-Seq Data.” *Genome Biology* 17 (1): 29. doi:10.1186/s13059-016-0888-1 (https://doi.org/10.1186/s13059-016-0888-1).

Islam, Saiful, Amit Zeisel, Simon Joost, Gioele La Manno, Pawel Zajac, Maria Kasper, Peter Lönnerberg, and Sten Linnarsson. 2014. “Quantitative Single-Cell RNA-Seq with Unique Molecular Identifiers.” *Nature Methods* 11 (2): 163–66. doi:10.1038/nmeth.2772 (https://doi.org/10.1038/nmeth.2772).

Krueger, Felix. 2018. “Trim Galore! Babraham Bioinformatics.” June 28. https://www.bioinformatics.babraham.ac.uk/projects/trim_galore/ (https://www.bioinformatics.babraham.ac.uk/projects/trim_galore/).

Langfelder, Peter, and Steve Horvath. 2008. “WGCNA: An R Package for Weighted Correlation Network Analysis.” *BMC Bioinformatics* 9 (1): 559. doi:10.1186/1471-2105-9-559 (https://doi.org/10.1186/1471-2105-9-559).

Lee, Jae-Young, Dae-Kwan Kim, Jeong-Jae Ko, Keun Pil Kim, and Kyung-Soon Park. 2016. “Rad51 Regulates Reprogramming Efficiency Through DNA Repair Pathway.” *Development & Reproduction* 20 (2): 163–69. doi:10.12717/DR.2016.20.2.163 (https://doi.org/10.12717/DR.2016.20.2.163).

Lun, Aaron. 2018. “Scran - Methods for Single-Cell RNA-Seq Data Analysis. Bioconductor.” July 17. http://bioconductor.org/packages/scran/ (http://bioconductor.org/packages/scran/).

Lun, Aaron T. L., Davis J. McCarthy, and John C. Marioni. 2018. “simpleSingleCell - a Step-by-Step Workflow for Low-Level Analysis of Single-Cell RNA-Seq Data with Bioconductor. Bioconductor.” May 25. http://bioconductor.org/packages/simpleSingleCell/ (http://bioconductor.org/packages/simpleSingleCell/).

Maaten, Laurens van der, and Geoffrey Hinton. 2008. “Visualizing Data Using T-SNE.” *Journal of Machine Learning Research* 9 (Nov): 2579–2605. http://www.jmlr.org/papers/v9/vandermaaten08a.html (http://www.jmlr.org/papers/v9/vandermaaten08a.html).

McCarthy, Davis J., Kieran R. Campbell, Aaron T. L. Lun, and Quin F. Wills. 2017. “Scater: Pre-Processing, Quality Control, Normalization and Visualization of Single-Cell RNA-Seq Data in R.” *Bioinformatics* 33 (8): 1179–86. doi:10.1093/bioinformatics/btw777 (https://doi.org/10.1093/bioinformatics/btw777).

Mohammed, Hisham, Irene Hernando-Herraez, Aurora Savino, Antonio Scialdone, Iain Macaulay, Carla Mulas, Tamir Chandra, et al. 2017. “Single-Cell Landscape of Transcriptional Heterogeneity and Cell Fate Decisions During Mouse Early Gastrulation.” *Cell Reports* 20 (5): 1215–28. doi:10.1016/j.celrep.2017.07.009 (https://doi.org/10.1016/j.celrep.2017.07.009).

Moore, R., W. Tao, E. R. Smith, and X.-X. Xu. 2014. “The Primitive Endoderm Segregates from the Epiblast in 1 Integrin-Deficient Early Mouse Embryos.” *Molecular and Cellular Biology* 34 (3): 560–72. doi:10.1128/MCB.00937-13 (https://doi.org/10.1128/MCB.00937-13).

Muñoz Descalzo Silvia, RuÉ Pau, Garcia-Ojalvo Jordi, and Arias Alfonso Martinez. 2012. “Correlations Between the Levels of Oct4 and Nanog as a Signature for Naïve Pluripotency in Mouse Embryonic Stem Cells.” *STEM CELLS* 30 (12): 2683–91. doi:10.1002/stem.1230 (https://doi.org/10.1002/stem.1230).

Ohnishi, Yusuke, Wolfgang Huber, Akiko Tsumura, Minjung Kang, Panagiotis Xenopoulos, Kazuki Kurimoto, Andrzej K. Oleś, et al. 2014. “Cell-to-Cell Expression Variability Followed by Signal Reinforcement Progressively Segregates Early Mouse Lineages.” *Nature Cell Biology* 16 (1): 27–37. doi:10.1038/ncb2881 (https://doi.org/10.1038/ncb2881).

Oliveri, Paola, Qiang Tu, and Eric H. Davidson. 2008. “Global Regulatory Logic for Specification of an Embryonic Cell Lineage.” *Proceedings of the National Academy of Sciences* 105 (16): 5955–62. doi:10.1073/pnas.0711220105 (https://doi.org/10.1073/pnas.0711220105).

Ptashne, M., and A. Gann. 2001. “Transcription Initiation: Imposing Specificity by Localization.” *Essays in Biochemistry* 37: 1–15.

Ritchie, Matthew E., Belinda Phipson, Di Wu, Yifang Hu, Charity W. Law, Wei Shi, and Gordon K. Smyth. 2015. “Limma Powers Differential Expression Analyses for RNA-Sequencing and Microarray Studies.” *Nucleic Acids Research* 43 (7): e47–e47. doi:10.1093/nar/gkv007 (https://doi.org/10.1093/nar/gkv007).

Scialdone, Antonio, Kedar N. Natarajan, Luis R. Saraiva, Valentina Proserpio, Sarah A. Teichmann, Oliver Stegle, John C. Marioni, and Florian Buettner. 2015. “Computational Assignment of Cell-Cycle Stage from Single-Cell Transcriptome Data.” *Methods*, Inferring gene regulatory interactions from quantitative high-throughput measurements, 85 (September): 54–61. doi:10.1016/j.ymeth.2015.06.021 (https://doi.org/10.1016/j.ymeth.2015.06.021).

Tam, Patrick P. L., and David A. F. Loebel. 2007. “Gene Function in Mouse Embryogenesis: Get Set for Gastrulation.” *Nature Reviews Genetics* 8 (5): 368–81. doi:10.1038/nrg2084 (https://doi.org/10.1038/nrg2084).