# Coding club meet-up - sleuth walkthrough

*Lingzi Li*

*20/09/2019*

## Introduction

sleuth computes differential expression using a statistical model that combines variance information from **biological replicates** and **bootstrapped technical replicates** to estimate true biological variance.

## Output from kallisto

```
#!/bin/bash
vds2Dir=/data/home/lingzili/VDS2/NGS_Runs/190807_A00316_0089_AHCM7JDRXX/Data/Intensities/BaseCalls/Demul

# Align pair-end reads with kallisto index from mouse GRCm38.p6 cDNA, bootstrapping 100 times
ls ${vds2Dir} | cut -d"_" -f 1,2 | sort -u | while read id
do
echo "***The sample ID is $id***"
read1=${vds2Dir}/${id}_R1_001.fastq.gz
read2=${vds2Dir}/${id}_R2_001.fastq.gz
echo "***Read 1 is ${read1} and read 2 is ${read2}***"
kallisto quant -t 16 \
-i /data/home/lingzili/mm10_genome/Mus_musculus/UCSC/GRCm38_kallisto/GRCm38.cdna.all.idx \
--gtf /data/home/lingzili/mm10_genome/Mus_musculus/UCSC/GRCm38_kallisto/Mus_musculus.GRCm38.96.gtf.gz \
-o kallisto_${id} \
-b 100 \
$read1 $read2
echo "***${id} is done.***"
done
```

Instead of actual technical replicates, kallisto makes use of **bootstrapped values** which serve as accurate proxies. **Bootstrapping** here is the process of repeating the quantification analysis after resampling with replacement from the original data, in order to simulate random sampling associated with sequencing.

**Effective length** is an effective length with respect to each possible fragment that maps to it. It accounts for the fact that not every transcript in the population can produce a fragment of every length starting at every position.

**Estimated counts** refers to the number of reads after dividing the mass of each read based on the likelihood.

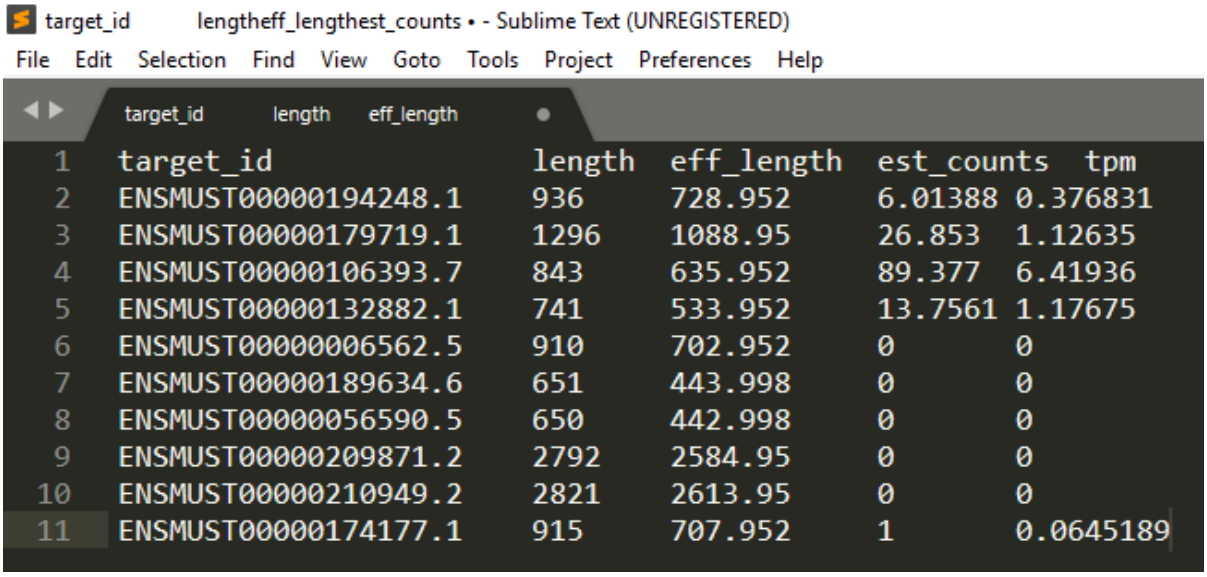
**Transcripts per million (TPM)** is a measurement of the proportion of transcripts in your pool of RNA. Shouldn't compare TPM across experiments.

## Libraries

```
# Load library
library(tidyverse)
library(knitr)
library(sleuth)
```

```
$ find . | sed -e "s/[^-][^\/]*\// |/g" -e "s/|\([^ ]\)/|-\1/"
.
 |-sample_info.csv
 |-SM5078
 | |-abundance.h5
 | |-abundance.tsv
 | |-run_info.json
 |-SM5079
 | |-abundance.h5
 | |-abundance.tsv
 | |-run_info.json
 |-SM5080
 | |-abundance.h5
 | |-abundance.tsv
 | |-run_info.json
 |-SM5081
 | |-abundance.h5
 | |-abundance.tsv
 | |-run_info.json
 |-SM5082
 | |-abundance.h5
 | |-abundance.tsv
 | |-run_info.json
 |-SM5083
 | |-abundance.h5
 | |-abundance.tsv
 | |-run_info.json
```

Figure 1: kallisto output documents




| | target_id | length | eff_length | est_counts | tpm |
|---|---|---|---|---|---|
| 2 | ENSMUST00000194248.1 | 936 | 728.952 | 6.01388 | 0.376831 |
| 3 | ENSMUST00000179719.1 | 1296 | 1088.95 | 26.853 | 1.12635 |
| 4 | ENSMUST00000106393.7 | 843 | 635.952 | 89.377 | 6.41936 |
| 5 | ENSMUST00000132882.1 | 741 | 533.952 | 13.7561 | 1.17675 |
| 6 | ENSMUST00000006562.5 | 910 | 702.952 | 0 | 0 |
| 7 | ENSMUST00000189634.6 | 651 | 443.998 | 0 | 0 |
| 8 | ENSMUST00000056590.5 | 650 | 442.998 | 0 | 0 |
| 9 | ENSMUST00000209871.2 | 2792 | 2584.95 | 0 | 0 |
| 10 | ENSMUST00000210949.2 | 2821 | 2613.95 | 0 | 0 |
| 11 | ENSMUST00000174177.1 | 915 | 707.952 | 1 | 0.0645189 |

Figure 2: tsv file after 100 rounds of bootstrapping

```r
library(biomaRt)
library(here)
```

## Experimental design table

```r
# Define file paths for kallisto directories
sample_id <- c("SM5078", "SM5079", "SM5080", "SM5081", "SM5082", "SM5083")

paths <- list(
  "C:/Users/lingzili/Documents/seq_tutorials/data/kallisto/SM5078",
  "C:/Users/lingzili/Documents/seq_tutorials/data/kallisto/SM5079",
  "C:/Users/lingzili/Documents/seq_tutorials/data/kallisto/SM5080",
  "C:/Users/lingzili/Documents/seq_tutorials/data/kallisto/SM5081",
  "C:/Users/lingzili/Documents/seq_tutorials/data/kallisto/SM5082",
  "C:/Users/lingzili/Documents/seq_tutorials/data/kallisto/SM5083"
)

# Add sample names to file paths
names(paths) <- sample_id
```

Note: At least one column needs to be labeled '**sample**'

```r
# Load experimental design
s2c <- read.csv("~/seq_tutorials/data/kallisto/sample_info.csv")

# Add file path to experimental design
s2c <- mutate(s2c, path = paths)
s2c[] <- lapply(s2c, as.character)

s2c
```

```
##   sample condition
## 1 SM5078      Chow
## 2 SM5079      Chow
## 3 SM5080      Chow
## 4 SM5081       HFD
## 5 SM5082       HFD
## 6 SM5083       HFD
##                                                                 path
## 1 C:/Users/lingzili/Documents/seq_tutorials/data/kallisto/SM5078
## 2 C:/Users/lingzili/Documents/seq_tutorials/data/kallisto/SM5079
## 3 C:/Users/lingzili/Documents/seq_tutorials/data/kallisto/SM5080
## 4 C:/Users/lingzili/Documents/seq_tutorials/data/kallisto/SM5081
## 5 C:/Users/lingzili/Documents/seq_tutorials/data/kallisto/SM5082
## 6 C:/Users/lingzili/Documents/seq_tutorials/data/kallisto/SM5083
```

## Get mouse gene names (GRCm38.p6)

Since the gene names are not automatically in the annotation of kallisto, we can get them from biomaRt.

```r
# Load mouse gene names from Ensembl
mart <- biomaRt::useMart("ensembl", dataset = "mmusculus_gene_ensembl")
t2g <- biomaRt::getBM(attributes = c("ensembl_transcript_id", "ensembl_gene_id", "external_gene_name"),
```

```r
# Rename the columns
t2g <- dplyr::rename(t2g, target_id = ensembl_transcript_id, ens_gene = ensembl_gene_id, ext_gene = ext

head(t2g)
```

```
##            target_id           ens_gene ext_gene
## 1 ENSMUST00000082423 ENSMUSG00000064372    mt-Tp
## 2 ENSMUST00000082422 ENSMUSG00000064371    mt-Tt
## 3 ENSMUST00000082421 ENSMUSG00000064370  mt-Cytb
## 4 ENSMUST00000082420 ENSMUSG00000064369    mt-Te
## 5 ENSMUST00000082419 ENSMUSG00000064368   mt-Nd6
## 6 ENSMUST00000082418 ENSMUSG00000064367   mt-Nd5
```

## Create the sleuth object (so)

```r
so <- sleuth_prep(s2c, target_mapping = t2g, extra_bootstrap_summary = TRUE, read_bootstrap_tpm = TRUE)
```

```
## reading in kallisto results
```

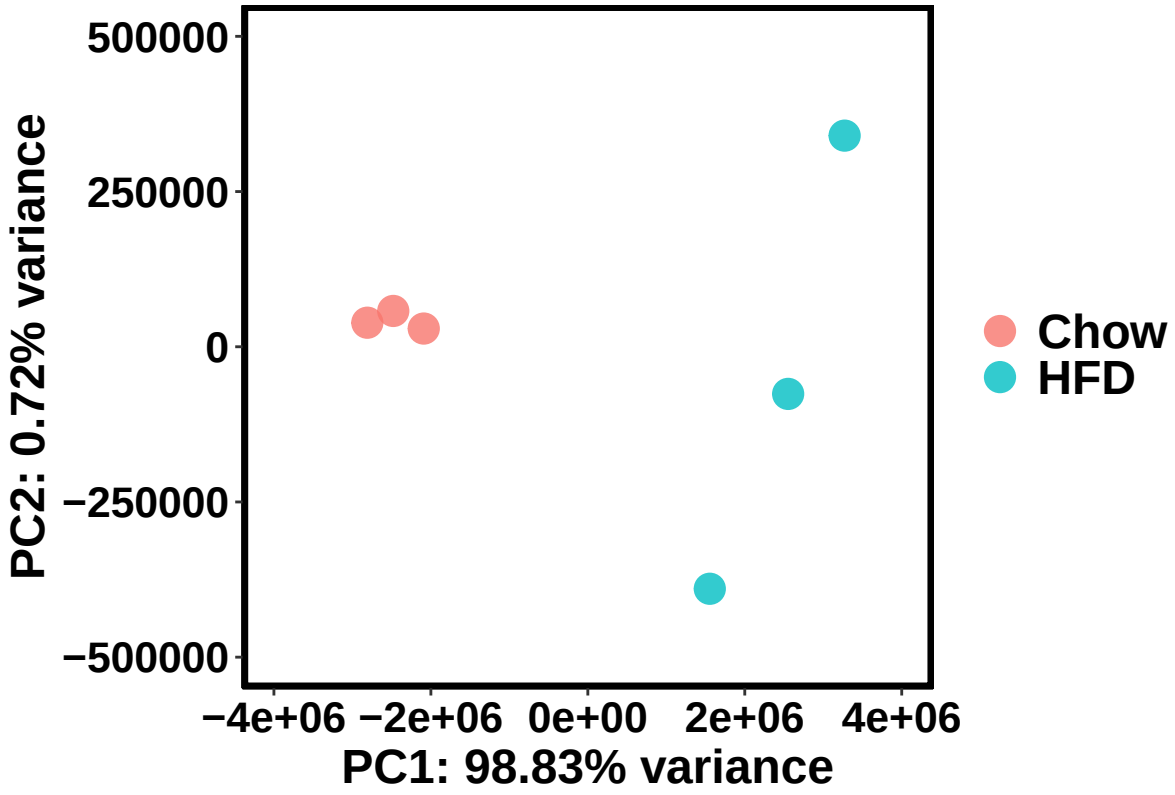
```
## dropping unused factor levels
```

```
## ......
## normalizing est_counts
## 36573 targets passed the filter
## normalizing tpm
## merging in metadata
## summarizing bootstraps
## ......
```

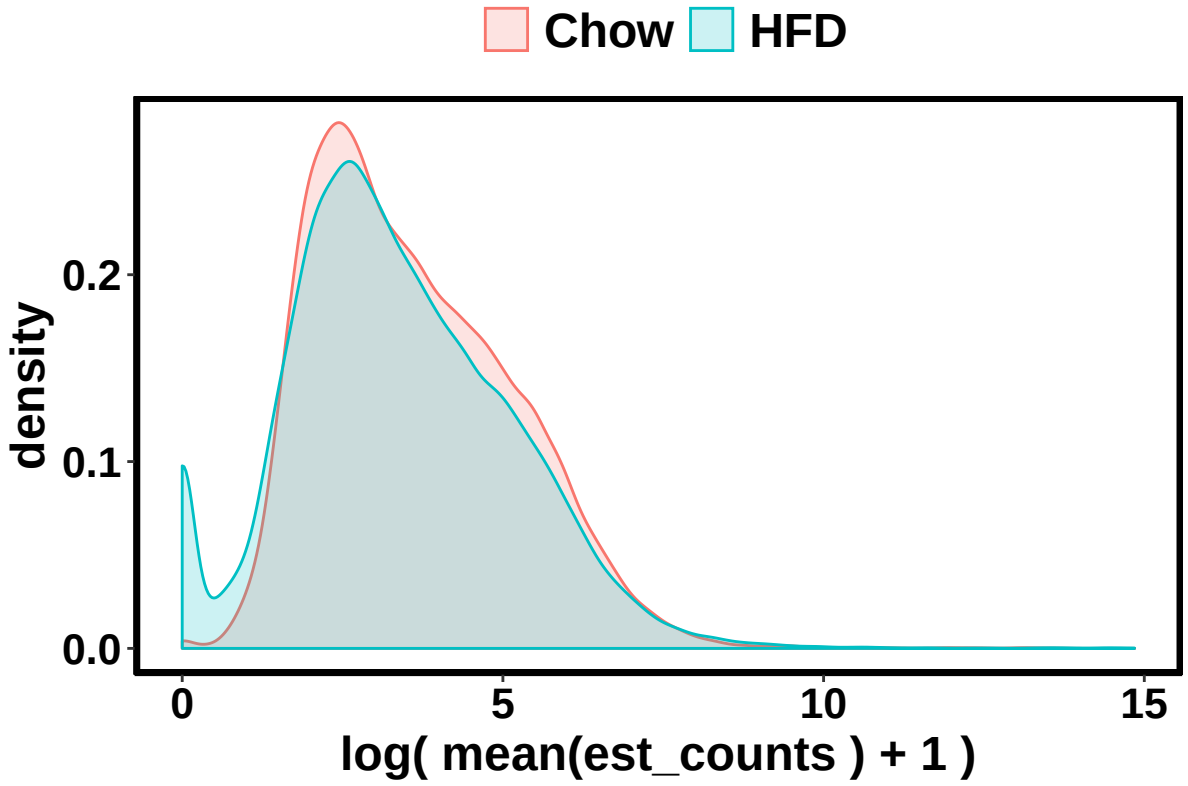
## PCA and group density

```r
# Define standard plot themne
standard_theme <- theme(
  axis.line = element_line(colour = "black"),
  axis.text.x = element_text(color = "black", size = 16, face = "bold"),
  axis.text.y = element_text(color = "black", size = 16, face = "bold"),
  axis.title.x = element_text(color = "black", size = 18, face = "bold"),
  axis.title.y = element_text(color = "black", size = 18, face = "bold"),
  legend.title = element_blank(),
  legend.text = element_text(color = "black", size = 18, face = "bold"),
  legend.key = element_rect(fill = "white"), # Remove grey background of the legend
  strip.text.x = element_blank(),
  strip.background = element_rect(fill = "white"),
  panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
  panel.background = element_blank(),
  panel.border = element_rect(colour = "black", fill = NA, size = 2),
  plot.margin = unit(c(0.5, 0.5, 0.5, 0.5), "cm"),
  plot.title = element_text(color = "black", size = 20, face = "bold")
)

# Calculate PC variance
pc_variance <- plot_pc_variance(so)
```

```
list_variance <- pc_variance$data$var

# PCA plot
pca_p1 <- plot_pca(so, color_by = "condition", text_labels = FALSE, point_size = 5)

pca_p2 <- pca_p1 +
  standard_theme +
  xlab(paste0("PC1: ", format(round(list_variance[1], 2), nsmall = 2), "% variance")) +
  ylab(paste0("PC2: ", format(round(list_variance[2], 2), nsmall = 2), "% variance")) +
  xlim(-4e+06, 4e+06) +
  ylim(-5e+05, 5e+05)

pca_p2
```




```
# Plot group density
plot_group_density(so, use_filtered = TRUE, trans = "log", grouping = "condition", offset = 1) +
  standard_theme +
  theme(legend.position = "top")
```




## DE analysis

Briefly, the likelihood ratio test (lrt) models the likelihood of the data given 2 models:

**full**: transcript abundance affected on one or more dependent variables (here just being treated or not)

**reduced**: transcript abundance unaffected by the treatment (null hypothesis)

```
# First fit a full model that includes a paramter for the condition
so <- sleuth_fit(so, ~condition, "full")
```

```
## fitting measurement error models
```

```
## shrinkage estimation
```

```
## 13 NA values were found during variance shrinkage estimation due to mean observation values outside
## The LOESS fit will be repeated using exact computation of the fitted surface to extrapolate the missi
## These are the target ids with NA values: ENSMUST00000127280.7, ENSMUST00000129607.1, ENSMUST0000015
```

```
## computing variance of betas
```

```
# Then fit a reduced model that only includes the intercept
so <- sleuth_fit(so, ~1, "reduced")
```

```
## fitting measurement error models
```

```
## shrinkage estimation
```

```
## 7 NA values were found during variance shrinkage estimation due to mean observation values outside o
## The LOESS fit will be repeated using exact computation of the fitted surface to extrapolate the missi
```

```
## These are the target ids with NA values: ENSMUST00000127280.7, ENSMUST00000129607.1, ENSMUST00000150
## computing variance of betas
```

```
# For each transcript, we perform a likelihood ratio test to determine whether the full model fits the
so <- sleuth_lrt(so, "reduced", "full")

# Make a table of the results
sleuth_table <- sleuth_results(so, "reduced:full", "lrt", show_all = FALSE)

# To check how many transcripts are differentially expressed between the two conditions (q-value <= 0.0
table(sleuth_table$qval <= 0.05)
```

```
##
## FALSE  TRUE
## 32123  4450
```

```
# Save the results
write.table(subset(sleuth_table, qval <= 0.05), file = "sleuth.DE_transcripts.qval_0.05.txt", sep = "\t

# Make a table that only includes the significantly DE transcripts
sleuth_significant <- dplyr::filter(sleuth_table, qval <= 0.05)
```

```
# Top 20 most significant DE transcripts in table and heatmap
head(sleuth_significant, 20) %>%
  dplyr::select(ext_gene, ens_gene, target_id, qval)
```

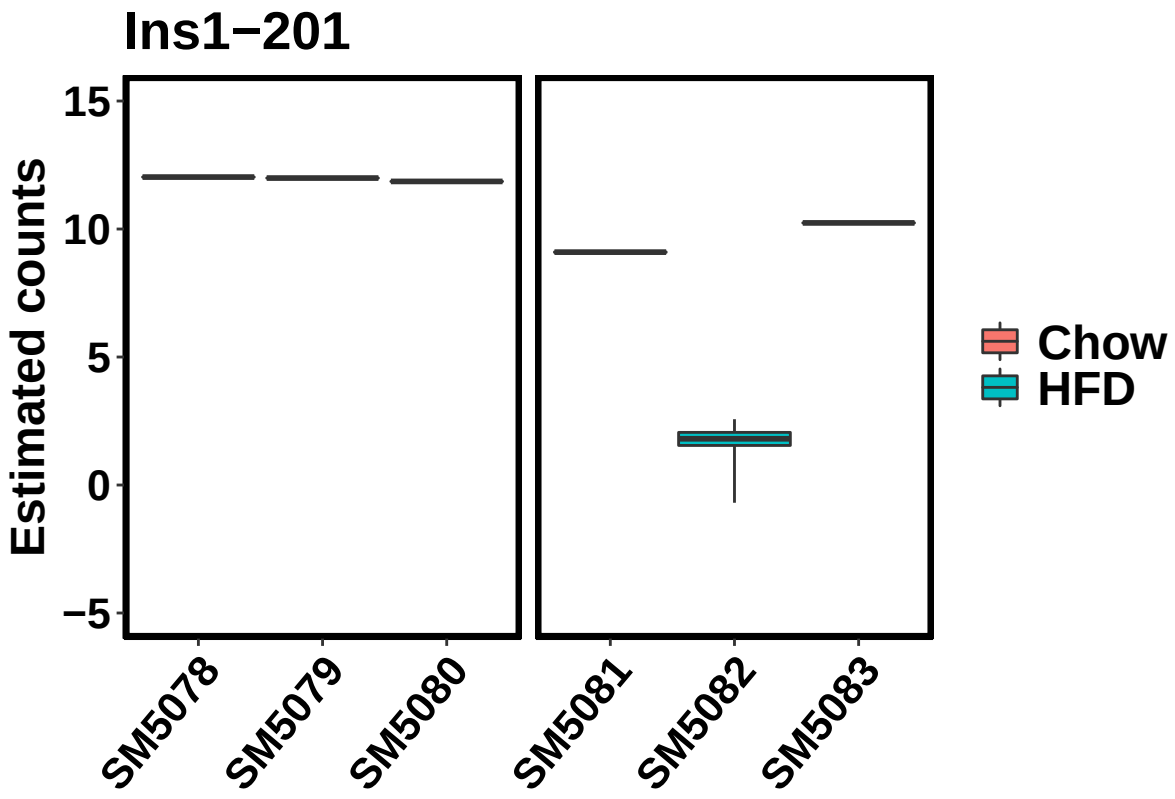
```
##    ext_gene           ens_gene             target_id        qval
## 1   Camk2n1 ENSMUSG00000046447  ENSMUST00000050918.3 0.001228901
## 2     Dmbt1 ENSMUSG00000047517  ENSMUST00000213064.1 0.001228901
## 3      Snd1 ENSMUSG00000001424 ENSMUST00000001460.13 0.001430032
## 4      Calr ENSMUSG00000003814  ENSMUST00000003912.6 0.001430032
## 5      Rps5 ENSMUSG00000012848 ENSMUST00000004554.13 0.001430032
## 6    Eif2b2 ENSMUSG00000004788 ENSMUST00000004910.11 0.001430032
## 7      Gcat ENSMUSG00000006378  ENSMUST00000006544.8 0.001430032
## 8       Srf ENSMUSG00000015605  ENSMUST00000015749.6 0.001430032
## 9     Rbpjl ENSMUSG00000017007  ENSMUST00000017151.1 0.001430032
## 10    Cd164 ENSMUSG00000019818 ENSMUST00000019962.14 0.001430032
## 11    Zwint ENSMUSG00000019923 ENSMUST00000020081.10 0.001430032
## 12  Hsp90b1 ENSMUSG00000020048 ENSMUST00000020238.13 0.001430032
## 13    Rack1 ENSMUSG00000020372  ENSMUST00000020640.7 0.001430032
## 14      Lbh ENSMUSG00000024063 ENSMUST00000024857.13 0.001430032
## 15  Arhgdig ENSMUSG00000073433  ENSMUST00000025019.8 0.001430032
## 16    Eif3a ENSMUSG00000024991  ENSMUST00000025955.7 0.001430032
## 17    Pdia3 ENSMUSG00000027248 ENSMUST00000028683.13 0.001430032
## 18   Rps3a1 ENSMUSG00000028081  ENSMUST00000029722.6 0.001430032
## 19     Rpl6 ENSMUSG00000029614 ENSMUST00000031617.12 0.001430032
## 20    Erp27 ENSMUSG00000030219  ENSMUST00000032343.6 0.001430032
```

## Heatmap

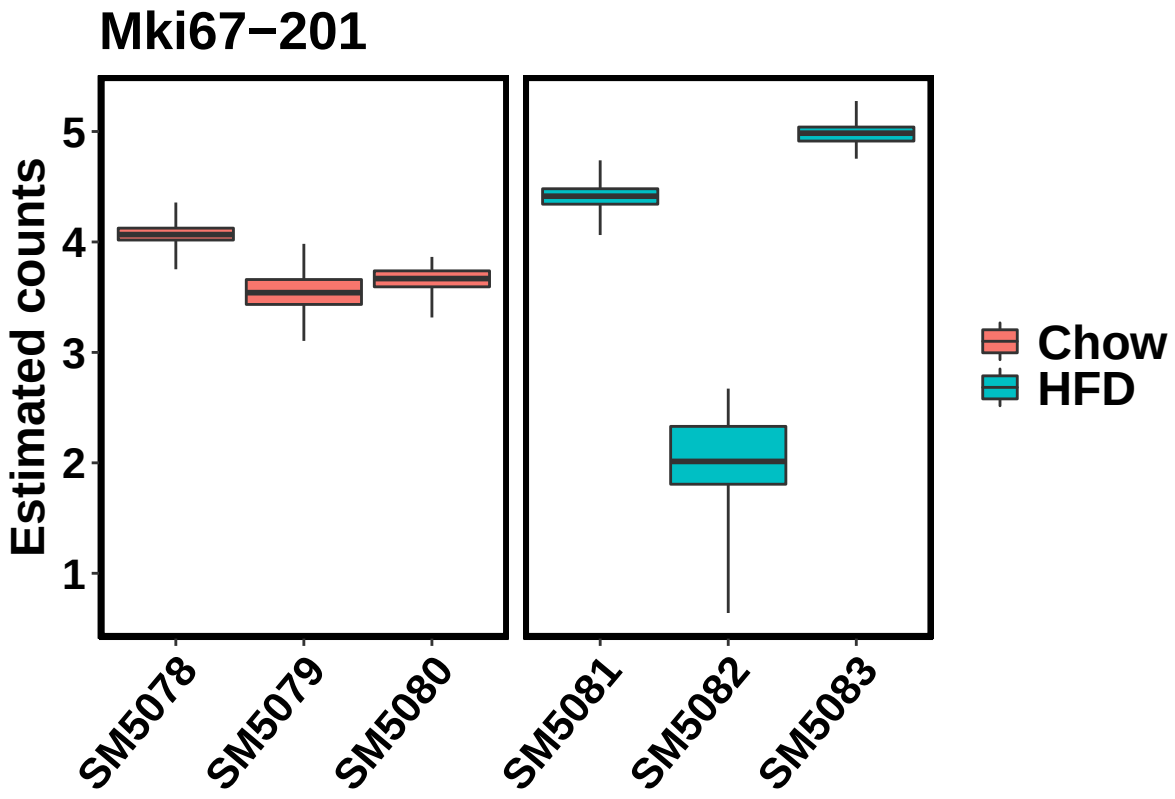
```r
plot_transcript_heatmap(so, head(sleuth_significant, 20)$target_id, "est_counts")
```

## Boxplot with gene expression counts

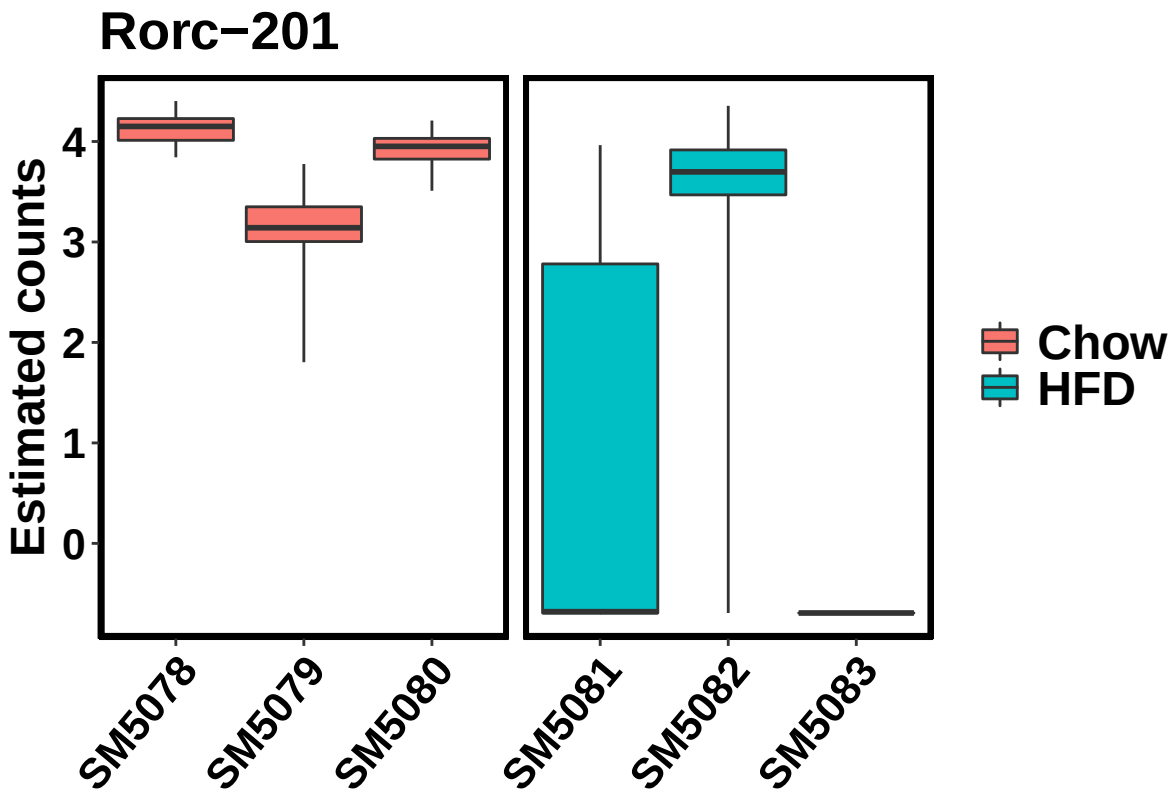
```r
# Define standard plot themne
standard_theme <- theme(
  axis.line = element_line(colour = "black"),
  axis.text.x = element_text(color = "black", size = 16, face = "bold"),
  axis.text.y = element_text(color = "black", size = 16, face = "bold"),
  axis.title.x = element_text(color = "black", size = 18, face = "bold"),
  axis.title.y = element_text(color = "black", size = 18, face = "bold"),
  legend.title = element_blank(),
  legend.text = element_text(color = "black", size = 18, face = "bold"),
  legend.key = element_rect(fill = "white"), # Remove grey background of the legend
  strip.text.x = element_blank(),
  strip.background = element_rect(fill = "white"),
  panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
  panel.background = element_blank(),
  panel.border = element_rect(colour = "black", fill = NA, size = 2),
  plot.margin = unit(c(0.5, 0.5, 0.5, 0.5), "cm"),
  plot.title = element_text(color = "black", size = 20, face = "bold")
)

# Plot variation in units of estimated counts
## Transcript: Ins1-201
Ins1_p1 <- plot_bootstrap(so, "ENSMUST00000039652.5", units = "est_counts")

Ins1_p2 <- Ins1_p1 +
  standard_theme +
  labs(title = "Ins1-201", x = NULL, y = "Estimated counts") +
  ylim(-5, 15)

Ins1_p2
```




```
## Transcript: Mki67-201
Ki67_p1 <- plot_bootstrap(so, "ENSMUST00000033310.8", units = "est_counts")

Ki67_p2 <- Ki67_p1 +
  standard_theme +
  labs(title = "Mki67-201", x = NULL, y = "Estimated counts")

Ki67_p2
```




```
## Transcript: Rorc-201
Rorc_p1 <- plot_bootstrap(so, "ENSMUST00000029795.9", units = "est_counts")

Rorc_p2 <- Rorc_p1 +
  standard_theme +
  labs(title = "Rorc-201", x = NULL, y = "Estimated counts")

Rorc_p2
```



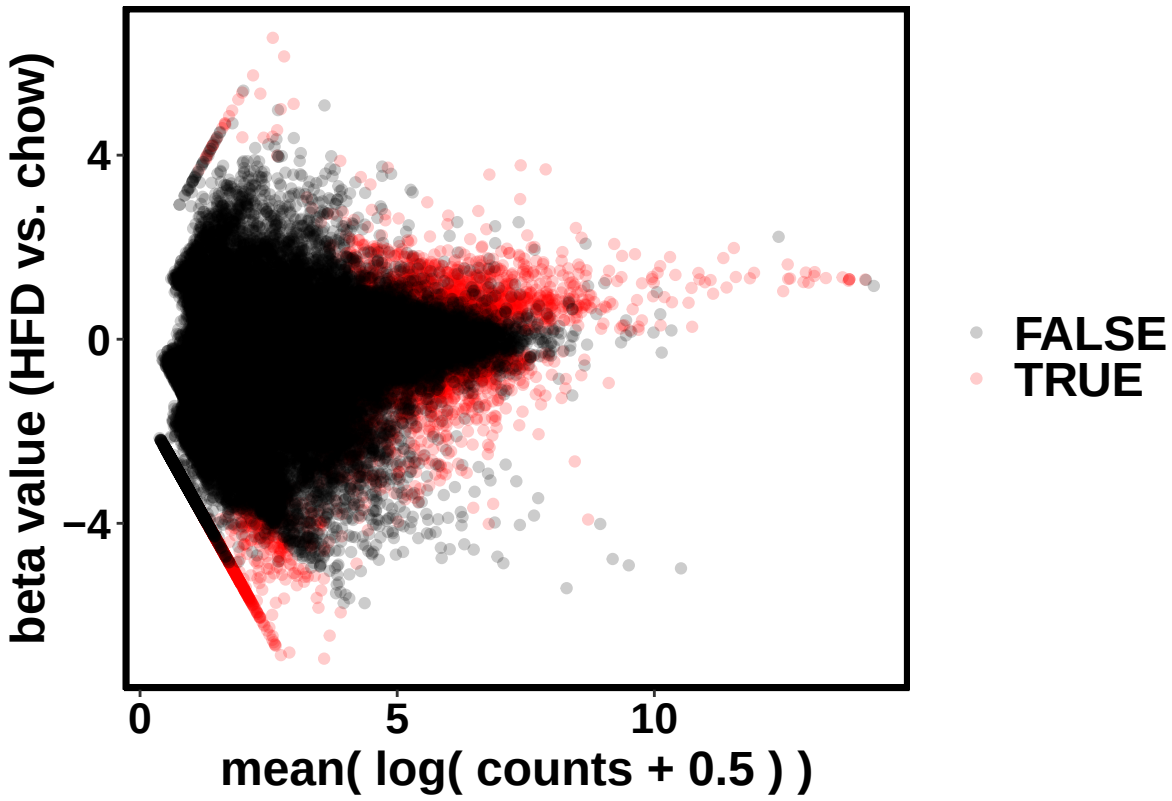

The likelihood ratio test (lrt) does not give a fold change for the transcript, just whether it is differentially expressed or not.

Sleuth provides another test called **Wald test**, which returns a **beta value** that "it is analogous to, but not equivalent to, the foldchange". Sleuth will transform elements in the condition field to 0s and 1s in alphabetical order. Positive beta values showing transcripts in which expression is greater in condition 1 than in condition 0.

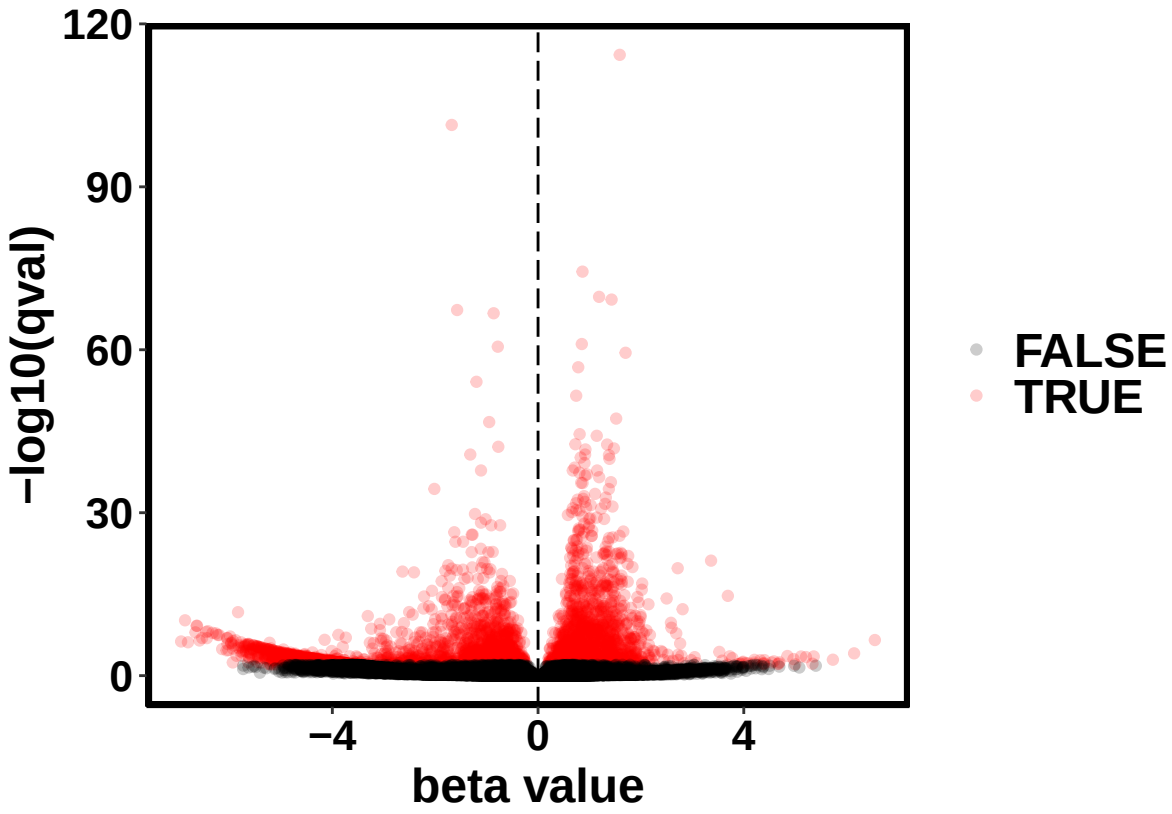
```
# Wald test (wt) returns a beta value that "it is analogous to, but not equivalent to,
so <- sleuth_wt(so, "conditionHFD")

sleuth_table <- sleuth_results(so, "conditionHFD", "wt", show_all = FALSE)
```

## MA plot and volcano plot

```
plot_ma(so, "conditionHFD", test_type = "wt", which_model = "full", sig_level = 0.01, point_alpha = 0.2
```




```
plot_volcano(so, "conditionHFD", test_type = "wt", which_model = "full", sig_level = 0.01, point_alpha =
```




## Interactive analysis

Sleuth live gives you an interactive visualization powered by Shiny.

```
sleuth_live(so)
```

## Reference

sleuth, Pachter Lab https://pachterlab.github.io/sleuth/

Differential Expression Workshop: March 2018 https://informatics.fas.harvard.edu/workshops/HarvardInformatics_DEworkshop_Spring2018.html

Pimentel, H., N.L.Bray, S.Puente, P.Melsted and L.Pachter (2017). "Differential analysis of RNA-seq incorporating quantification uncertainty." Nature Methods 14: 687. https://www.nature.com/articles/nmeth.4324