Department of Computer Science
Algorithmic Bioinformatics

Universitätsstr. 1 D–40225 Düsseldorf

Heinrich Heine Universität Düsseldorf

# A user-friendly Snakemake pipeline for differential expression analysis

**Julian Kremer**

Bachelor Thesis

| | |
|---|---|
| Submission: | 11.03.2020 |
| Supervisor: | Univ.-Prof. Dr. G. Klau |
| Second Assessor: | Dr.phil. A. Dilthey |
| Advisor: | Philipp Spohr |

# Abstract

RNA sequencing uses high throughput techniques for transcriptome profiling. An important part of RNA sequencing is the analysis of read counts per gene to find differentially expressed genes across samples. This thesis presents an extended version of an existing Snakemake pipeline to find differentially expressed genes based on count data from RNA sequencing. The extended Snakemake pipeline focuses on providing reasonable output for multiple conditions at once, while maintaining user-friendliness, as well as adding new options to choose from.

# Contents


**1 Introduction** **1**

1.1 Contribution . . . . . 1

**2 Background information** **2**

2.1 Biological background . . . . . 2

2.1.1 RNA sequencing . . . . . 2

2.2 Statistical background . . . . . 3

2.2.1 p-value . . . . . 3

2.2.2 Null hypothesis . . . . . 3

2.2.3 False Discovery Rate . . . . . 3

2.2.4 Linear regression . . . . . 5

2.2.5 Formula . . . . . 5

2.2.6 Log2 Fold Change . . . . . 6

2.3 Differential expression analysis . . . . . 6

**3 Tools** **6**

3.1 Snakemake . . . . . 6

3.2 Conda . . . . . 6

3.3 STAR . . . . . 7

3.4 DESeq2 . . . . . 7

3.5 Cutadapt . . . . . 7

3.6 RSeQC . . . . . 7

3.7 IHW . . . . . 7

3.8 MultiQC . . . . . 7

**4 Extending the pipeline** **8**

4.1 Contrasts . . . . . 9

4.1.1 Limitations . . . . . 9

4.1.2 Implementation . . . . . 9

4.2 Plots . . . . . 9

4.2.1 Mean SD Plot . . . . . 9

4.2.2 heatmap . . . . . 10

4.2.3 Dispersion Plot . . . . . 10

4.2.4 p-value histograms . . . 10
4.2.5 Time Course Experiments . . . 10
4.2.6 IHW . . . 10
4.3 Further additions . . . 11
4.3.1 Shrinkage . . . 11
4.3.2 Transformation . . . 11
4.3.3 Grouping . . . 11

**5 Example run** **12**
5.1 Data . . . 12
5.2 Output . . . 12
5.2.1 MA plot . . . 12
5.2.2 PCA . . . 14
5.2.3 Mean SD Plot . . . 15
5.2.4 heatmap . . . 16
5.2.5 Dispersion Plot . . . 17
5.2.6 p-value histogram . . . 18
5.2.7 Time Course Experiment . . . 19
5.2.8 IHW . . . 20
5.2.9 MultiQC and RSeQC plots . . . 21
5.2.10 Shrinkage . . . 23
5.2.11 Formulas and Contrasts . . . 23

**6 Discussion and Outlook** **25**
6.1 Limitations for experienced users . . . 25
6.2 Limitations on plots . . . 25
6.3 Limitations on the input . . . 25
6.4 Other high throughput sequencing input . . . 25

**7 Source Code** **25**

**8 Acknowledgements** **26**

**References** **27**

# 1 Introduction

In the field of transcriptomics, RNA sequencing and differential expression analysis are omnipresent. RNA sequencing is used to get insight into transcriptomes, whereas differential expression analysis is used to identify genes with distinct expression differences under conditions. A condition is a difference between samples, e.g. drug treatment in medicine where the patients are either treated with a drug or not treated with a drug, as well as mutations in genomics where an organism can be a wild type or a specific mutation. Because most tools for these technologies only exist for less than one decade, it is more important than ever to be able to see how conditions impact gene expression. Nevertheless, in practice, it is often not straightforward to perform such analysis due to the sheer amount of tools available and background knowledge needed. Therefore, we present a more user-friendly Snakemake pipeline, by extending an existing pipeline [14].

## 1.1 Contribution

The existing pipeline allows users to do basic differential expression analysis. But has restrictions, such as only being able to perform analysis on one condition at a time or providing the user with a minimal amount of plots. We address those restrictions by extending the pipeline in a way that allows us to create a wider variety of plots, as well as taking care of comparison for multiple conditions and the need to specify formulas, from the formula interface of the R language, which would typically need background knowledge. Further, we decided to add a broad range of advanced options and to keep the options the existing pipeline had to offer for more experienced users, for the sake of convenience. Another important reason to extend the existing pipeline besides being open source, was the reproducibility it provides using the Conda system [1] as it allows for predefined environments and packages, as well as using Snakemake [13] for structuring and managing the pipeline. To provide examples throughout this thesis and give a more detailed explanation of the output this pipeline generates, we used the following published dataset [7] from the GEO database [4] for an example run. This dataset focuses on the identification of differentially expressed genes in the mammary glands of mice, throughout various development stages and different cells.

# 2 Background information

We use this section to focus on introducing information needed to understand later parts of our thesis.

## 2.1 Biological background

To be able to understand the data this pipeline is working on, we present a short breakdown of RNA sequencing.

### 2.1.1 RNA sequencing

RNA sequencing is a procedure that allows us to get insight into the transcriptome of a cell, and thus the ability to see different expression between genes under specific conditions. We can split this procedure into 3 steps, starting with the creation of a RNA sequencing library, then moving on to sequencing and lastly analysis of the data. In the first step, RNA is extracted from biotic material, the mRNA (messenger RNA) is passed on, reverse transcribed into cDNA (complementary DNA) and then sequencing adaptors are ligated to the ends. Finally, the cDNA is amplified with PCR (Polymerase Chain Reaction) or in vitro transcription. This first step can be seen in Figure 1 below.

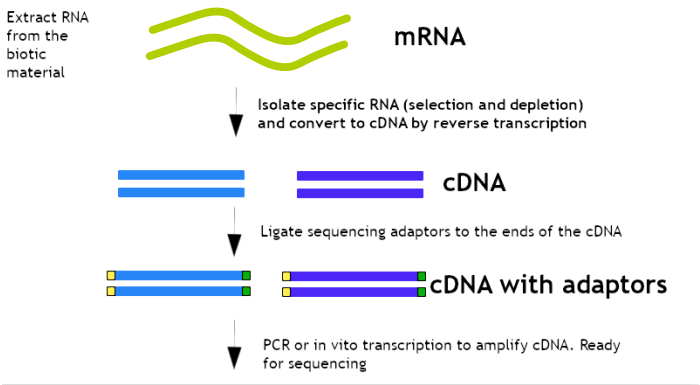



Figure 1: Library preparation of RNA sequencing procedure

The first part of the sequencing step is then performed by high-throughput sequencing platforms using techniques, such as next-generation sequencing to generate sequencing reads. These sequencing reads are then stored in files, such as the fastq files that we use as input in the pipeline. Afterwards, the reads are mapped to a reference genome, resulting in counts, i.e. the number of reads that overlap at specific positions in the genome.

Finally, we can use the read counts to analyse differences between genes, such as differential expression analysis.

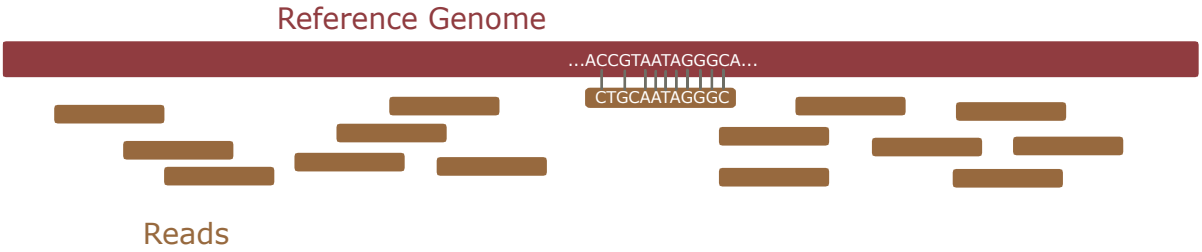



Figure 2: Mapping reads to a reference genome

## 2.2 Statistical background

We use this part of the thesis to introduce our readers to basic statistical concepts to understand the output this pipeline generates, as well as techniques and input for the advanced methods the pipeline has to offer.

### 2.2.1 p-value

A p-value is a form of quantifying the probability of obtaining a difference as extreme as the one observed under the assumption, that the null hypothesis is true. Therefore we conclude, that a small p-value, i.e. less or equal to a given significance level $\alpha$, indicates to reject the null hypothesis and otherwise to accept it. E.g. a significance level of $\alpha = 0.05$ is common in research, but the significance level can be set in this pipeline. There are also ways to correct for p-values, which represent wrong numbers caused by statistical or other biases. These corrected p-values are also known as adjusted p-values.

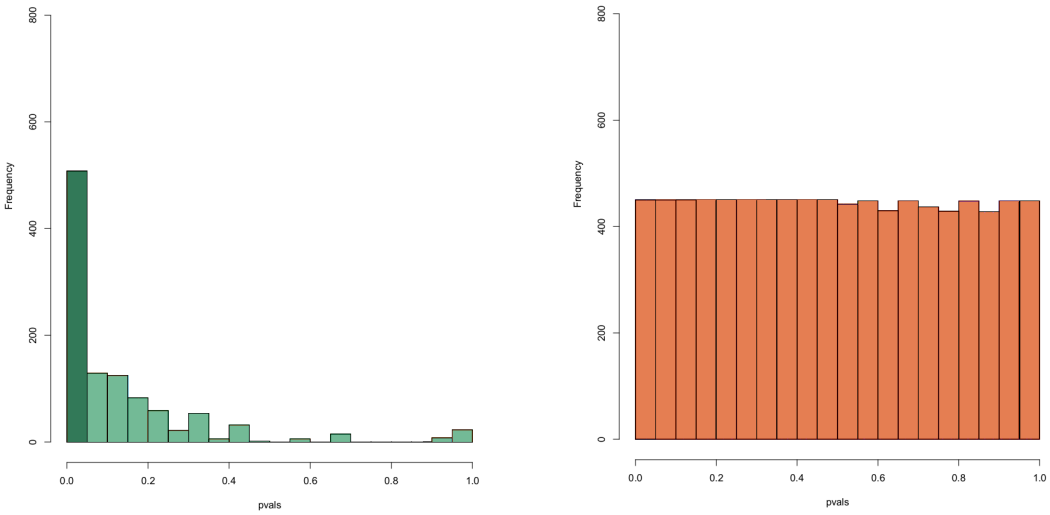
### 2.2.2 Null hypothesis

The null hypothesis is used in science as a hypothesis to represent the point of view, that there is no significant effect present in an experiment. In this case, the null hypothesis is that there are no differences between the read counts of different conditions for a gene. The results in the form of p-values this pipeline generates are all created under this null hypothesis.

### 2.2.3 False Discovery Rate

The False Discovery Rate provides us with the ability to sort out false positives. A false positive indicates that a significant result is true when in reality is not. False positives can be explained by assuming that p-values under the null hypothesis should be uniformly distributed, and thus some of the p-values under the significance level are wrongly viewed as significant. To give an example: If we compare samples under a condition, such as treated and untreated, we might have of genes that differ due to the treatment between the samples, those genes will likely show up as small p-values in research, see 3a. When comparing the other genes, which were not affected by the treatment we get p-values which are going to be uniformly distributed, because there might be genes that are not overlapping even if they were untreated, see 3b. This is problematic because now there are several

false positives under our significance level, see brown in 4. For this reason, False Discovery Rate is used. The original procedure to control for False Discovery Rate at a significance level $\alpha$ was first described by Yoav Benjamini and Yosef Hochberg [2]. The procedure will result in adjusted p-values where false positives are higher than the significance level, i.e. we get a smaller number of observed p-values (because of the adjusted p-values) under the significance level.




(a) genes that differentiate

(b) genes that do not differentiate

Figure 3: Examples for False Discovery Rate

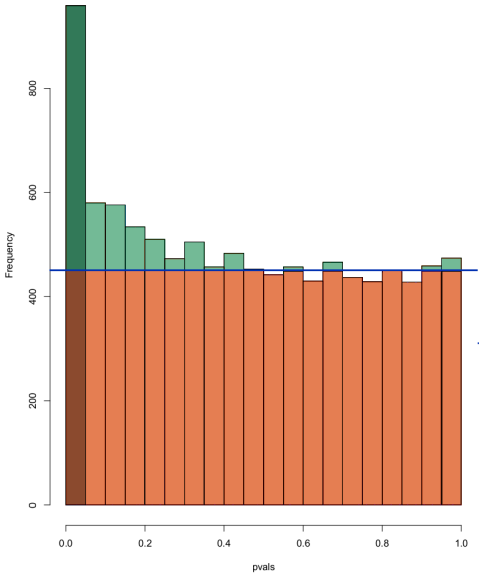



Figure 4: Both 3a and 3b combined

### 2.2.4 Linear regression

Linear regression allows to model the relationship between dependent and independent variables by fitting a data model in the model coefficients. We define a linear regression model as following:

$$Y_i = \beta_0 + \beta_1 X_{i,1} + \beta_2 X_{i,2} + \ldots + \beta_p X_{i,p} + \varepsilon_i \text{ and } i = 1, \ldots, n \quad (1)$$

Where $Y$ is the dependent variable, $X$ are vectors of independent variables, $\beta_0$ being the intercept term and $\beta$ the coefficients. Further, $p$ refers to the number of independent variables and $\varepsilon_i$ are i.i.d. (independent and identically distributed) errors.

### 2.2.5 Formula

The R programming language provides us with a formula interface. A formula can be seen as syntax that allows us to indicate which columns should be used when fitting statistical models. The syntax can be described as follows:

| **Operator** | **Example** | **Meaning** |
|---|---|---|
| + | +condition | Include condition |
| - | -condition | Exclude condition |
| : | conditionA:conditionB | Interaction between both conditions |
| * | conditionA*conditionB | Short for:conditionA+conditionB+conditionA:conditionB |
| ^ | (conditionA+conditionB)^2 | Short for: (conditionA+conditionB)*(conditioA+conditionB) |
| I() | I(conditionA+conditionB) | Function to use brackets in arithmetic way |
| 1 | condition-1 | Removes the intercept term |
| 0 | condition+0 | Specifies the model to have no intercept |

Table 1: Formula operators

Every single formula can be written either $a \sim b$ or $\sim b$, where $a \sim b$ refers to modelling $a$ as a function of $b$ and $\sim b$ just refers to $b$ being a independent variable. For this thesis, we will only focus on the use of right-sided formulas such as $\sim b$, because it is the only valid input for our pipeline.

To give a small example of how we use formulas in our pipeline, imagine we have a dataset as input with two columns **condition** and **status**. In case we want to compare the effects of **condition** our formula would have the following design *~condition*. Further, if we want to compare multiple conditions, i.e. both conditions in our example we would use a formula like *~condition+status*. Lastly, we can also compare interactions across conditions, i.e. *~condition*status*. For each of these formulas several different possible contrasts to choose from are created by the pipeline, later in our example run, we present some of the contrasts as a result of different formulas specified. This should give a better understanding of how formulas are used in combination with the pipeline.

### 2.2.6 Log2 Fold Change

Fold Change is a form of measurement for how much a quantity changes between initial and final values. It can be calculated using the following formula, where A is the initial and B the final value: $\frac{B}{A}$. Because we get fold changes between 0 and 1 for values where the initial value is bigger than the final value, we use $\log_2$ to represent those small fold changes as negative values. This is done, because it allows for better visualization between value ratios, rather than having values with small fold changes converge to the limit 0. To put this into perspective with an example, if we have a ratio of 2 between two gene counts and another ratio of 0.5 we would get the values $\log_2 2 = 1$ and $\log_2 0.5 = -1$.

## 2.3 Differential expression analysis

Differential expression analysis allows us to find differences in gene expression between two or more conditions across samples. As mentioned earlier, a condition refers to differences, such as drug treatment as a condition with the effects treated or untreated or in our example later on, status with the effects pregnant, lactate or virgin. This process is generally performed on counts created through an aligner. To determine, if the counts for a gene show differences higher than expected between conditions, tools such as DESeq2 are used to estimate these differences and as a result estimate, the significance of differences found. In the case of this pipeline, we get a list of p-values for all of the genes, as well as adjusted p-values that should correct for False Discovery Rate. Throughout all ways to investigate RNA sequencing data, differential expression analysis is a more simple yet powerful and common way.

# 3 Tools

In the following, we introduce the tools used to build the pipeline and to ensure reproducibility.

## 3.1 Snakemake

Snakemake [13] is a tool to create and manage workflows. Through several tasks so-called *rules*, we can set up the order in which these tasks are executed by letting them depend on each other. The resulting workflow is automated and the input is set dynamically. For this bachelor thesis, we extend the number of rules and options in the existing Snakemake workflow.

## 3.2 Conda

Conda [1] is an open source system for environment and package management. It allows to find, install and update multiple packages and save them in an environment as well as switching between environments. Therefore, we use Conda to have a persistent environment for our pipeline and thus ensure reproducibility.

### 3.3 STAR

STAR (Spliced Transcripts Alignment to a Reference)[3] is an open source software for RNA sequence alignment. It allows us to align reads to a reference genome and outperforms other aligners by a factor of > 50 in mapping speed. But it is memory consuming, and thus it is advised to run on high performance clusters. We use it in our pipeline to align reads in fastq files to a reference genome to get read counts for each of the genes in our sample.

### 3.4 DESeq2

DESeq2 [17] is a package for differential expression analysis of count data based on the R language. It uses statistical methods to get better estimates and focuses on the analysis of expression strength. Further, the DESeq2 package provides us with the functions and statistical backup to explore the differences in gene expression between multiple conditions, i.e. the contrasts.

### 3.5 Cutadapt

Cutadapt [18] is a tool that allows error-tolerant adapter trimming before read mapping as well as specifying the adapter sequence that should be trimmed. In fact we have Cutadapt as an optional setting in our pipeline, to allow for trimming of adapter sequences for our high-throughput sequencing reads.

### 3.6 RSeQC

RSeQC [20] is a package created for quality control of RNA sequencing experiments. It allows multiple ways of quality control, such as sequence quality, read distribution over the genome structure and many others. Therefore, it provides users with output to ensure that the RNA sequencing data is suitable for subsequent analysis.

### 3.7 IHW

IHW (Independent Hypothesis Weighting)[10] is a package to calculate adjusted p-values by running several statistical methods, such as False Discovery Rate. We use it as a way to control for False Discovery Rate and display the results in plots.

### 3.8 MultiQC

MultiQC [6] is a tool to summarize the output of the RSeQC package and create a single report from the data for readability.

# 4 Extending the pipeline

This part of the thesis focuses on explaining the reasoning behind extending the pipeline to its current state and what has been extended, as well as an example run of our pipeline, particularly explaining the input and output. To give a visual representation of how the pipeline works, we start by showing the reduced DAG (directed acyclic graph) created by us for this pipeline, see Figure 5. Every vertex of the DAG represents several tools or scripts. Whereas all the directed edges represent the order in which the vertices are called, i.e. vertices with outgoing edges distribute their output to the respective successor. Thus, starting the pipeline at vertices with no incoming edges and the other vertices are only executed when they got all the outputs from their predecessors. The output is marked with a blue border. The red vertices refer to optional choices and the green vertices refer to obligatory steps.

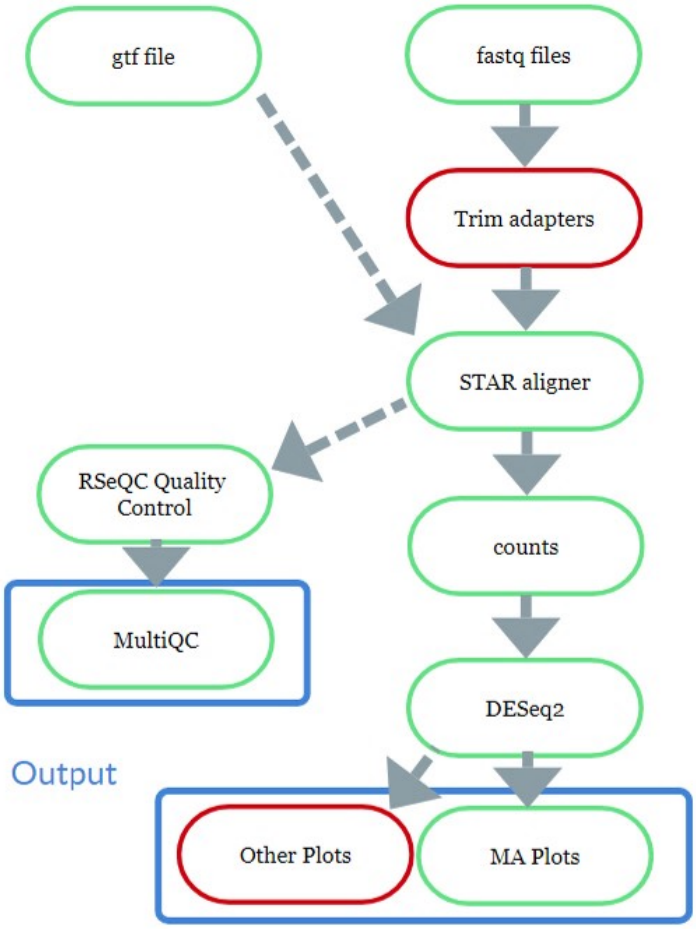



Figure 5: DAG of our pipeline

## 4.1 Contrasts

A contrast is used to define the different effects of one condition and pass them onto the pipeline for evaluation. E.g. a condition could be **status** with the effects *lactate*, *pregnant* or *virgin*. Therefore, all possible contrasts would be *pregnant vs lactate*, *pregnant vs virgin* and *lactate vs virgin*. We would then pass them onto the pipeline to see if any genes under the condition **status**, and between those contrasts are differentially expressed.

### 4.1.1 Limitations

For each gene there could be multiple conditions, that cause differential expression across samples. The original implementation could only test for one of those conditions at a time with the constraint to specify contrasts and a design formula. More precisely the DESeq2 package we use needs us to set such contrasts beforehand and can only create contrasts in combination with a design formula that also has to be set beforehand. To remove this level of complexity for the inexperienced user we implemented the following functionality. That is to say, there is no possible way for us to know which conditions the user wants to test for a given set of samples and the sheer amount of possible design formulas in regarding the samples' conditions. Therefore, we opted to create all possible contrasts for the user to choose from and as well taking care of the design formulas needed.

### 4.1.2 Implementation

As just mentioned, we have to specify several contrasts and design formulas concerning the samples as the first step of our pipeline. To compute the contrasts we take the columns within the sample file and prepare all possible contrasts for each column representing the condition and each row representing an effect due to this condition, e.g. on our example run we would take **status** as a condition and *virgin*, *pregnant* and *lactate* as the unique effects. Additionally, we create a new imaginary column representing the combination of conditions, i.e. the combination of the given columns and effects and prepare contrasts for this new column, we will refer to it as **group**. E.g. column **group** with the unique effects *basalvirgin*, *basalpregnant*, *basallactate*, *luminalvirgin* and so on. This preparation of contrasts is performed in the Snakefile, and therefore written in Python. After preparation, we pass the contrasts on to DESeq2 to create 2 data sets with different design formulas, based on our sample file, one for the original columns and another one for our group column.

## 4.2 Plots

### 4.2.1 Mean SD Plot

The Mean SD Plot is part of the vsn [9] package. We used it to extend the pipeline, as it allows us to plot the standard deviation of the transformed data, against the mean across all samples used. Therefore, we can use the plot to verify variance stabilisation.
The red line in the plot represents the running median of the standard deviation and should be horizontal with only a few fluctuations to ensure that there is no variance-mean dependency. Variance-

mean dependency would let us know that there is some sort of confounding and would not allow for further meaningful expression analysis [5].

#### 4.2.2 heatmap

The heatmap is part of the pheatmap [11] package. We chose to extend the pipeline by a heatmap of the count matrix because it is another important step in quality assurance to find outliers in samples that might be caused by anomalies. In particular, the added heatmap shows us the expression of the 20 most expressed genes of log2 normalized counts. The color range is ranging from the lowest to the highest of the normalized values.

#### 4.2.3 Dispersion Plot

Another diagnostic plot is the Dispersion Plot because we can see the result of the DESeq2 *Estimation of dispersion* process, where the final gene estimates (blue) are shrunken towards the fitted (red) estimates except for outliers. For this reason, we also chose to extend the pipeline by adding this plot.

#### 4.2.4 p-value histograms

Histograms of p-values are a common way to view data. We extended the pipeline with p-value histograms for every contrast before using independent hypothesis weighting on them. This gives us a first look at the original p-values (not the adjusted p-values) of our data and could hint in favour of rejecting or not rejecting the null hypothesis.

#### 4.2.5 Time Course Experiments

To provide a way to test for Time Course Experiments, we extended the pipeline with a possibility, that we found in one of the DESeq2 vignettes. This approach offers to find genes that react in a condition specific manner over time. In particular, we let DESeq2 perform a likelihood ratio test, by specifying a reduced formula to remove the condition specific differences over time, which should result in genes with low p-values as the ones that showed condition specific effects. Since our example data did not include a condition offering time differences, the example in Figure 12 was created with the following data package [15]. The data is used to find differences of fission yeast in response to oxidative stress at different times and between a wild type group and a group with deletion of aft21.

#### 4.2.6 IHW

For a more powerful way to control for False Discovery Rate and thus being able to conclude if genes might have expressed differentially due to given conditions, we added the IHW (independent hypothesis weighting) package from Bioconductor to the pipeline and followed an example in the respective vignette on how to use it and how to interpret the received plots. It is an optional setting in this pipeline. The IHW package calculates weights for all p-values to let them average

out at 1 and computes adjusted p-values by applying the Benjamini Hochberg procedure on all weighted p-values, it also uses other techniques, such as splitting the hypothesis into different covariates and then randomly splitting them into folds, to avoid overfitting the data. As a result, we get two diagnostic plots. The first one represents the different weights for the folds across the covariates, where genes with a high baseMean get higher weights than genes with low baseMeans. A baseMean [16] refers to the mean of normalized counts, divided by size factors, across all samples. And the second plot shows those folds on the unweighted p-values.

## 4.3 Further additions

### 4.3.1 Shrinkage

Shrinkage is a way to control for low gene counts that tend to be *noisy*, as a result of the strong variance of $\log_2$ Fold Change estimates (*Empirical Bayes shrinkage for fold-change estimation* [17]). To address the problem, we added different shrinkage options to choose from in the pipeline. The options for shrinkage are: normal (which can not be used for formulas with interaction terms, that being said formulas including the **:** operator) and ashr for all formulas. Normal is the shrinkage estimator provided by DESeq2 using an adaptive normal distribution and ashr is the shrinkage estimator from the ashr package [19].

### 4.3.2 Transformation

Transformation is another option added to the pipeline and offers the ability to transform the count data, i.e. replacing each value with a transformed value by using functions to compute the new values. These functions are normal, rlog and vst. Normal uses $\log_2$ with a pseudo count of 1 on the normalized counts, whereas rlog uses the *regularized logarithm* [17] and vst uses a variance stabilizing transformation concept [8], but needs at least 1000 genes to work. Later on, we add brackets to captions of the plot to let the reader know which transformation was used during plotting.

### 4.3.3 Grouping

We added grouping to the pipeline, as it allows for more complex contrasts by taking multiple conditions and creating one condition. The grouping options refer to the combination of all columns and creating an imaginary column as explained earlier, but now can be set optional for an experienced user that decides to use their formulas and choose specific contrasts. In particular, by using grouping in this pipeline all conditions of the provided data, the columns, are combined into a single condition, and contrasts can be specified for this new column.

# 5 Example run

For a visual representation of the plots, we now continue with an example run of the pipeline, particularly explaining the input and output. We show all of the options available in the pipeline in the following.

## 5.1 Data

The experiment we use in our example run, focuses on the differential expression of genes in luminal and basal cells from mammary glands of virgin, 18.5 day pregnant and 2 day lactating mice (*Mus musculus*), respectively. For reproducibility, the files were obtained from popular sites. More precisely, the fastq files were obtained from https://www.ebi.ac.uk, the fasta file was obtained from http://www.ensembl.org and the gtf file was obtained from https://www.ncbi.nlm.nih.gov.
To represent the data more coherently and to show how our sample sheet looks, please see Table 2.

Table 2: sample data

| **sample** | **condition** | **status** |
|---|---|---|
| MCL1.DG | basal | virgin |
| MCL1.DH | basal | virgin |
| MCL1.DI | basal | pregnant |
| MCL1.DJ | basal | pregnant |
| MCL1.DK | basal | lactate |
| MCL1.DL | basal | lactate |
| MCL1.LA | luminal | virgin |
| MCL1.LB | luminal | virgin |
| MCL1.LC | luminal | pregnant |
| MCL1.LD | luminal | pregnant |
| MCL1.LE | luminal | lactate |
| MCL1.LF | luminal | lactate |

## 5.2 Output

In the following, we will explain the output the pipeline created. For illustration purposes, we include some of the plots generated by the pipeline run on the example data.

### 5.2.1 MA plot

MA (M=Intensity Ratio and A=Average Intensity) plots have already been included in the existing pipeline. They are used to represent how different the read counts of two groups are. Each of the dots represents a gene. The red dots represent the genes with a (through DESeq2 adjusted) p-value smaller than what we specified in our config, in this particular case we used the standard

0.1. Therefore, the grey dots represent genes with p-values higher than the significance level. The x-axis is used to represent the mean of normalized counts, whereas the y-axis represents the log2 fold change. It is also important to note, that we provided DESeq2 with unnormalized counts because DESeq2 normalized them on its own. We extended the pipeline with the option to use shrinkage on MA plots to remove the noise caused by genes with low counts. Figure 6 shows MA plots from our example run created through not grouped contrasts, i.e. *basal vs luminal*.

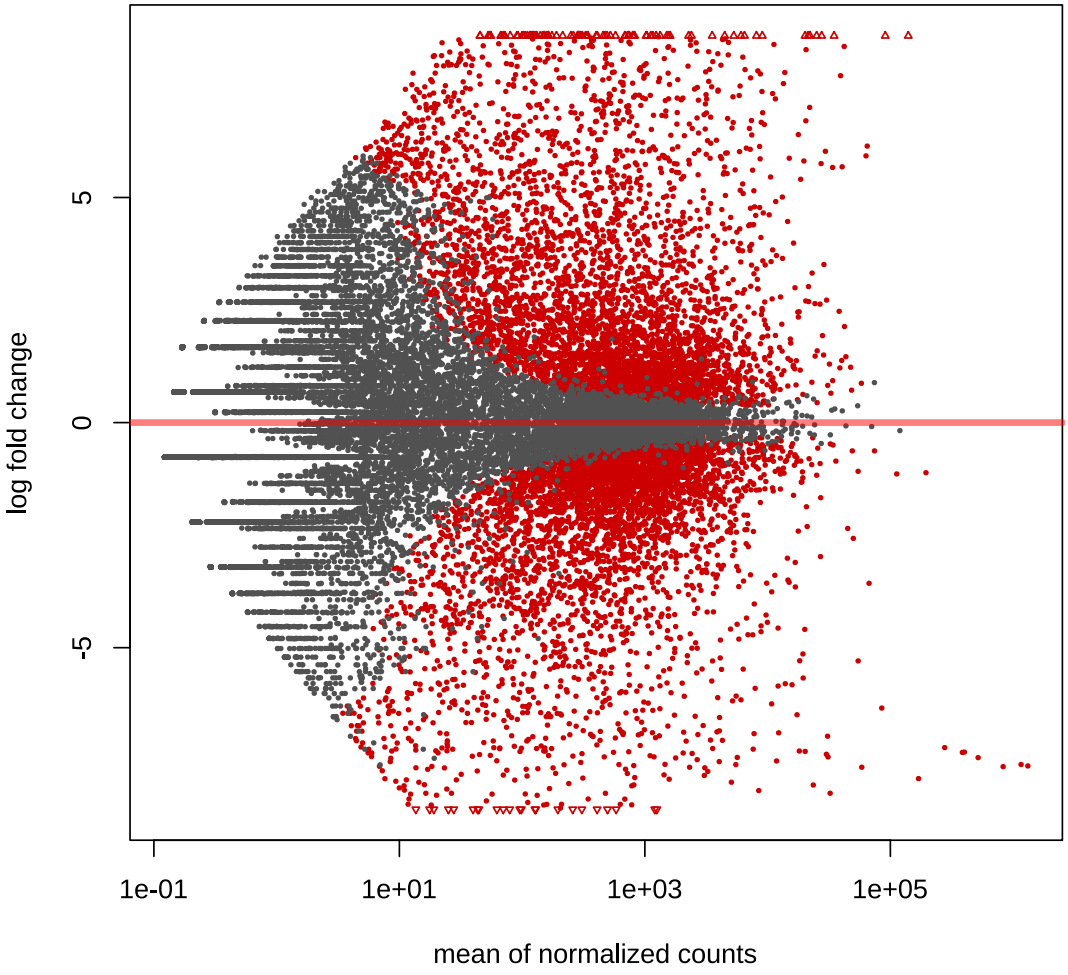



Figure 6: basal vs luminal

### 5.2.2 PCA

The PCA (Principal component analysis) plot has also been included in the existing pipeline and allows users to create a PCA plot for each condition present, by using the PCA techniques of dimensionality reduction where one condition would normally add one dimension. To simplify it, this is done by clustering each correlation among all conditions and thus samples into a 2D-graph. Highly correlated ones should cluster together. It is also important to note that the PC1-axis is more important than the PC2-axis, i.e. differences along the PC1-axis can be seen as more important than differences with the same distance along the PC2-axis. This is also seen in the percentage on the axes, which displays for how much of the total variation each axis accounts, note that only the two highest percentages are used in the dimension reduction, thus resulting in variances that do not need to account for 100 percent. For the example run we receive the following PCA, showing us that pregnant and lactating mice seemingly are more similar to each other, while virgin mice differentiate the most compared with the others.

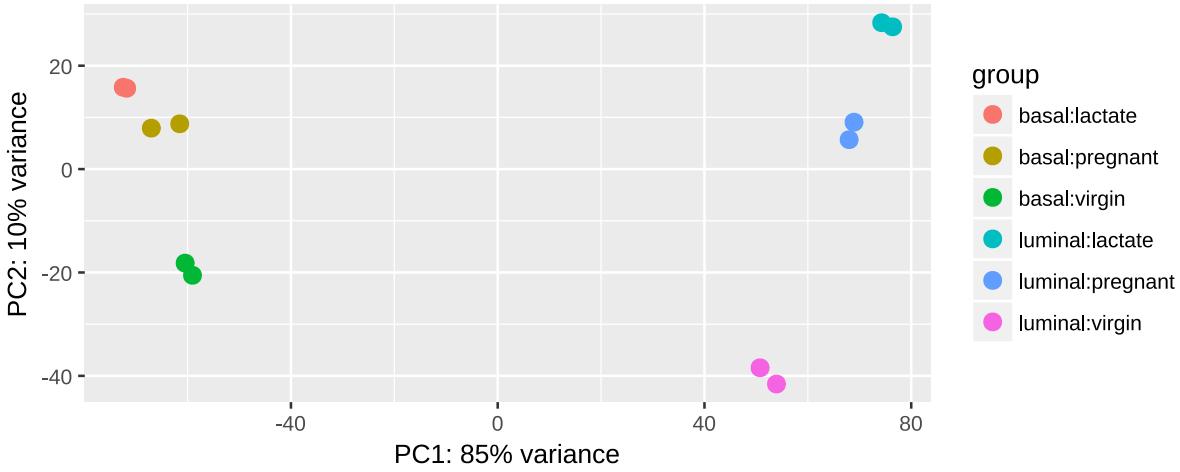



Figure 7: PCA for condition and status (with rlog transformation)

### 5.2.3 Mean SD Plot

The Mean SD Plot on the data in our example run can be seen in Figure 8.

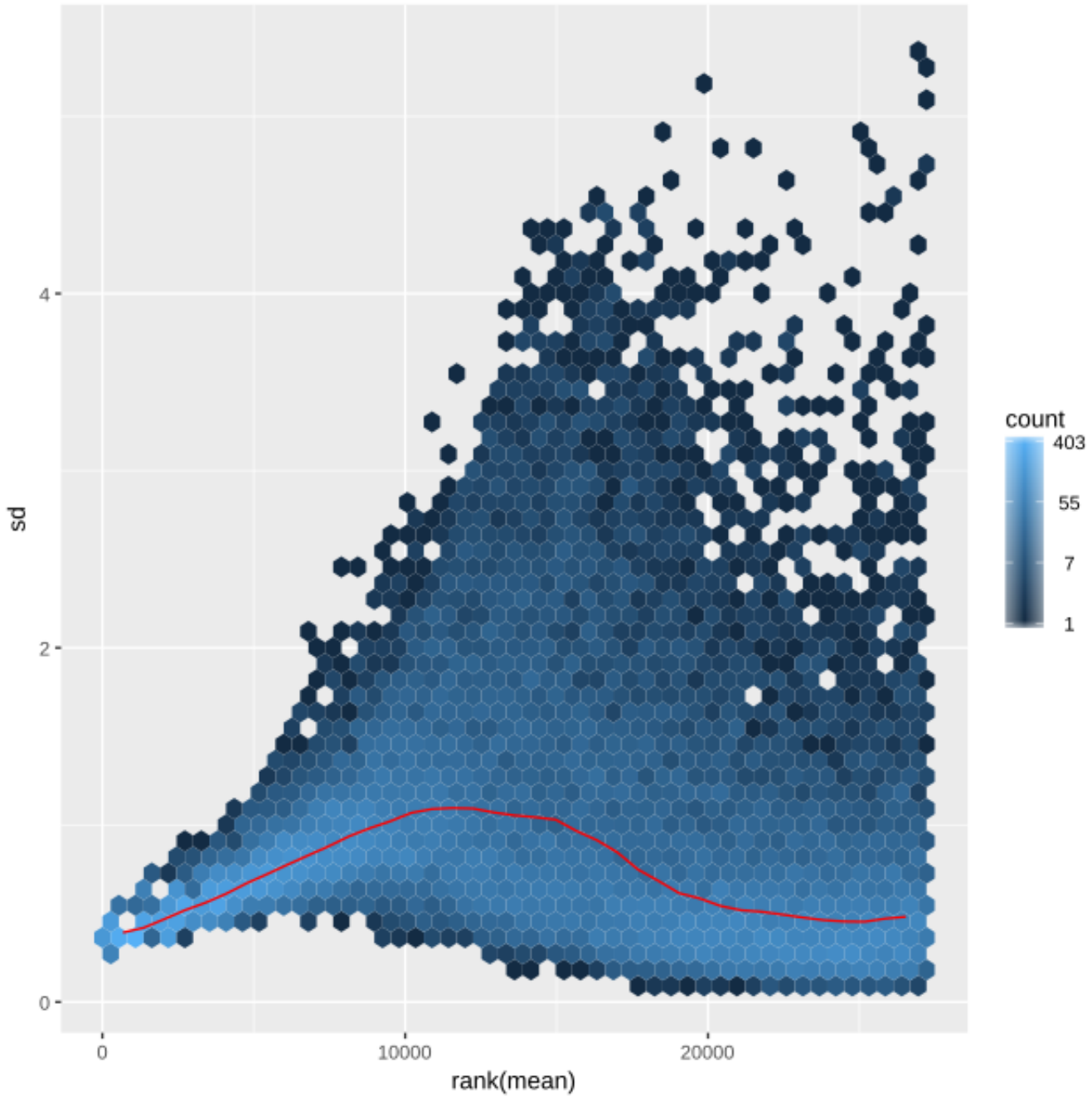



Figure 8: meanSdPlot

### 5.2.4 heatmap

See Figure 9 for the heatmap on our example run, presenting the top 20 most expressed genes.

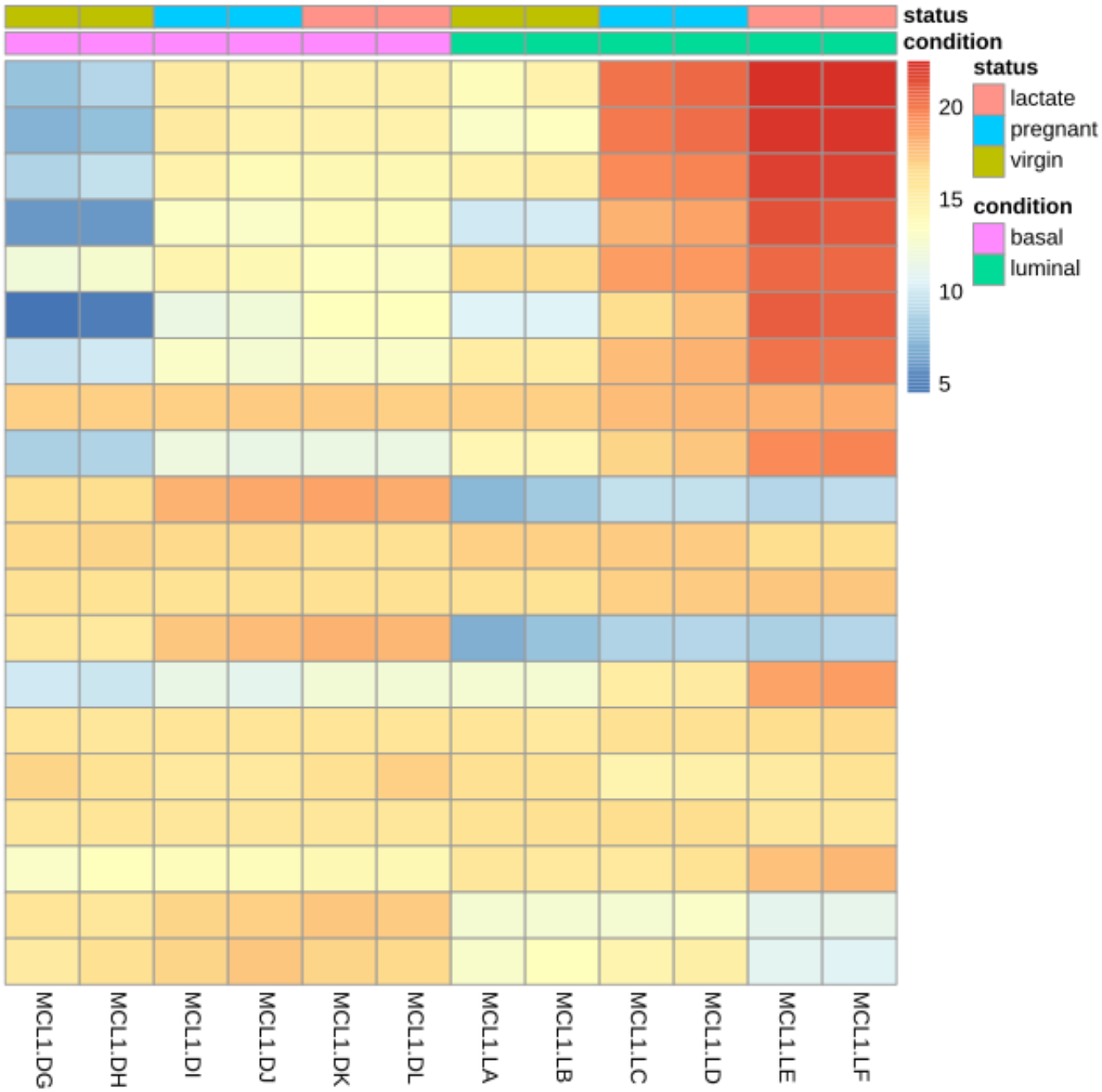



Figure 9: Heatmap of top 20 most expressed genes

### 5.2.5 Dispersion Plot

For an example, see Figure 10 created during our example run.

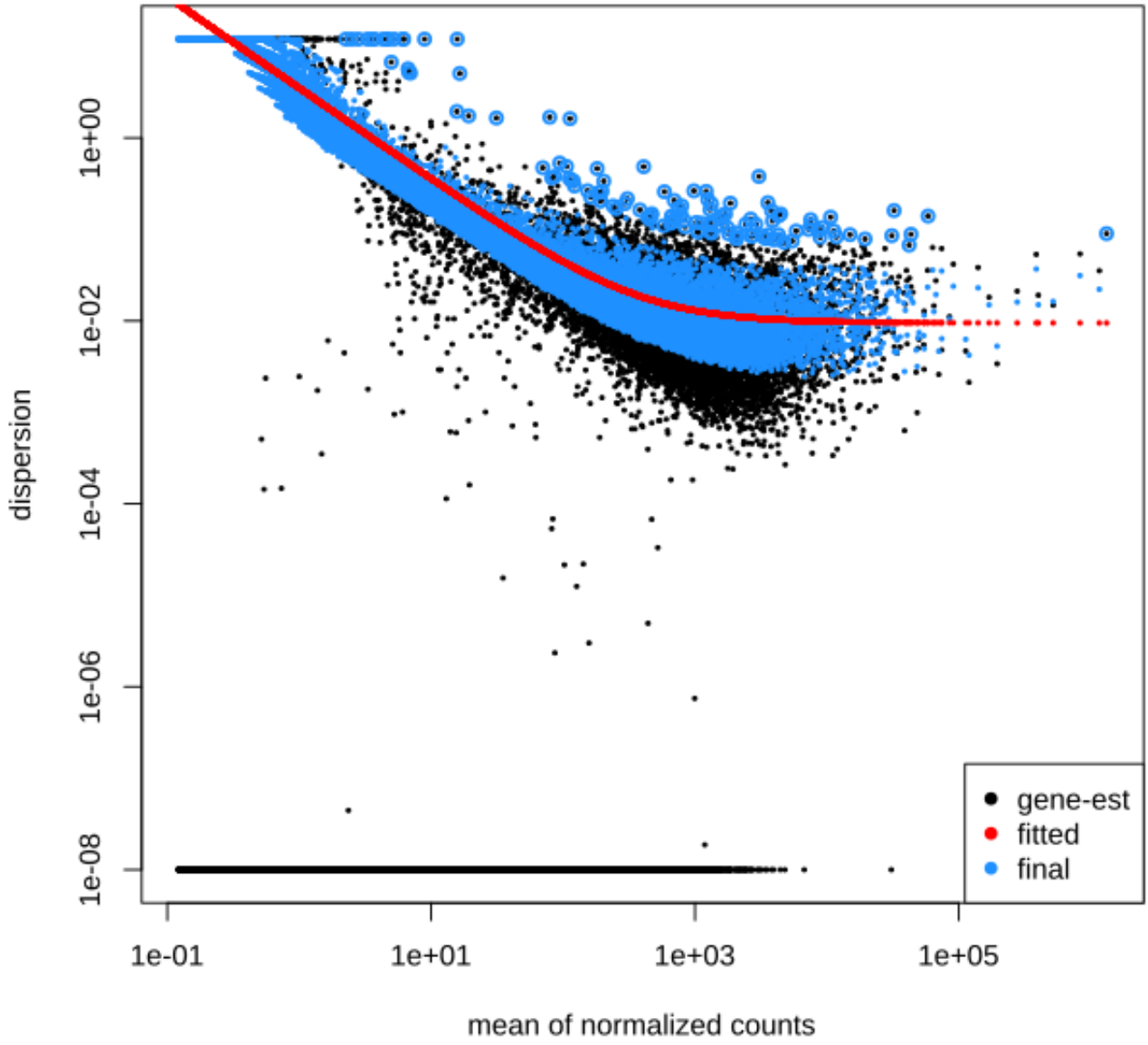



Figure 10: Dispersion Plot

### 5.2.6 p-value histogram

In the following, we provide an example p-value histogram received with the virgin vs pregnant contrast in our example run, see Figure 11.

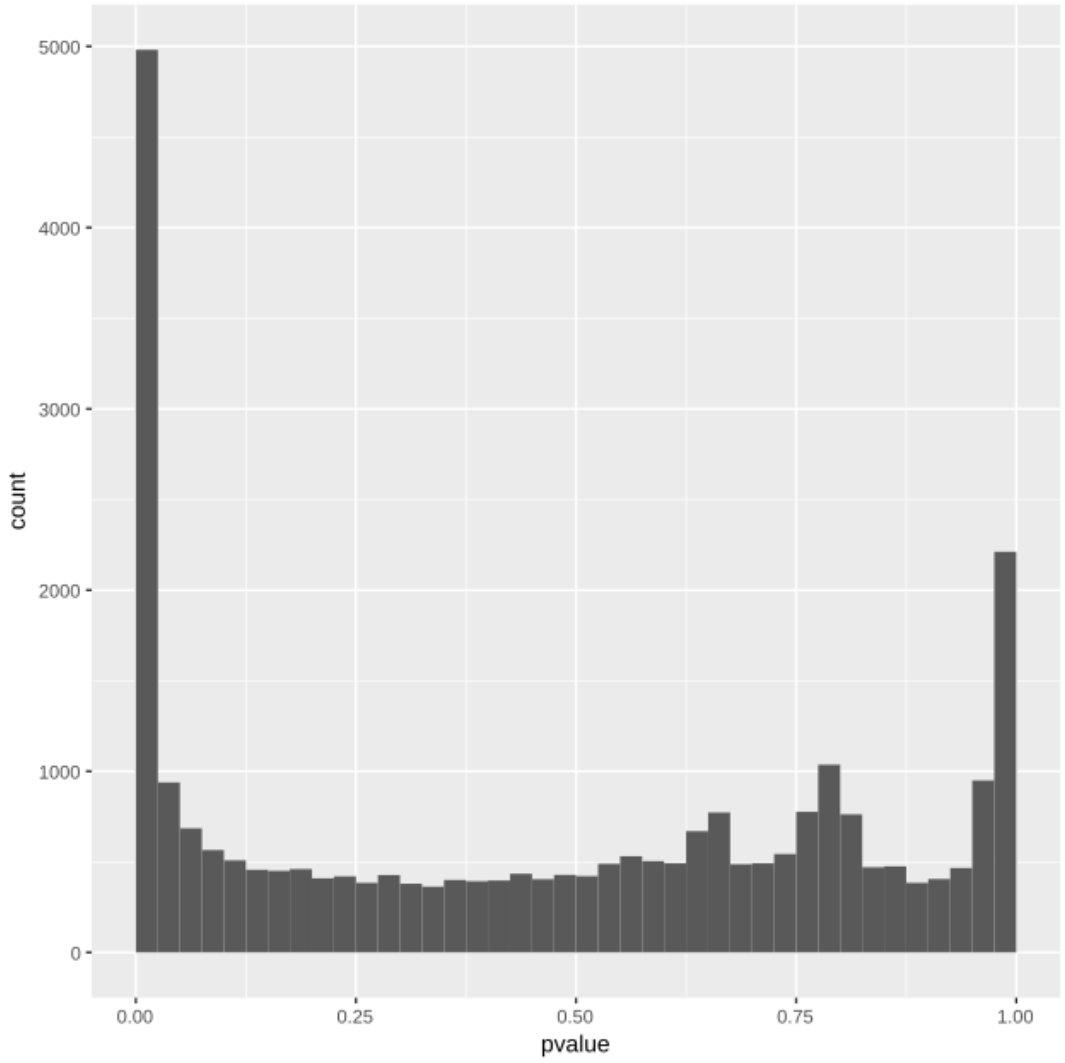



Figure 11: p-value histogram for virgin vs pregnant

### 5.2.7 Time Course Experiment

The example in Figure 12 was created with the following data package [15], to find differences of fission yeast in response to oxidative stress at different times and between a wild type group and a group with deletion of aft21. Please note, that we did not use all samples and just subset the data for the differences in the first 60 minutes.

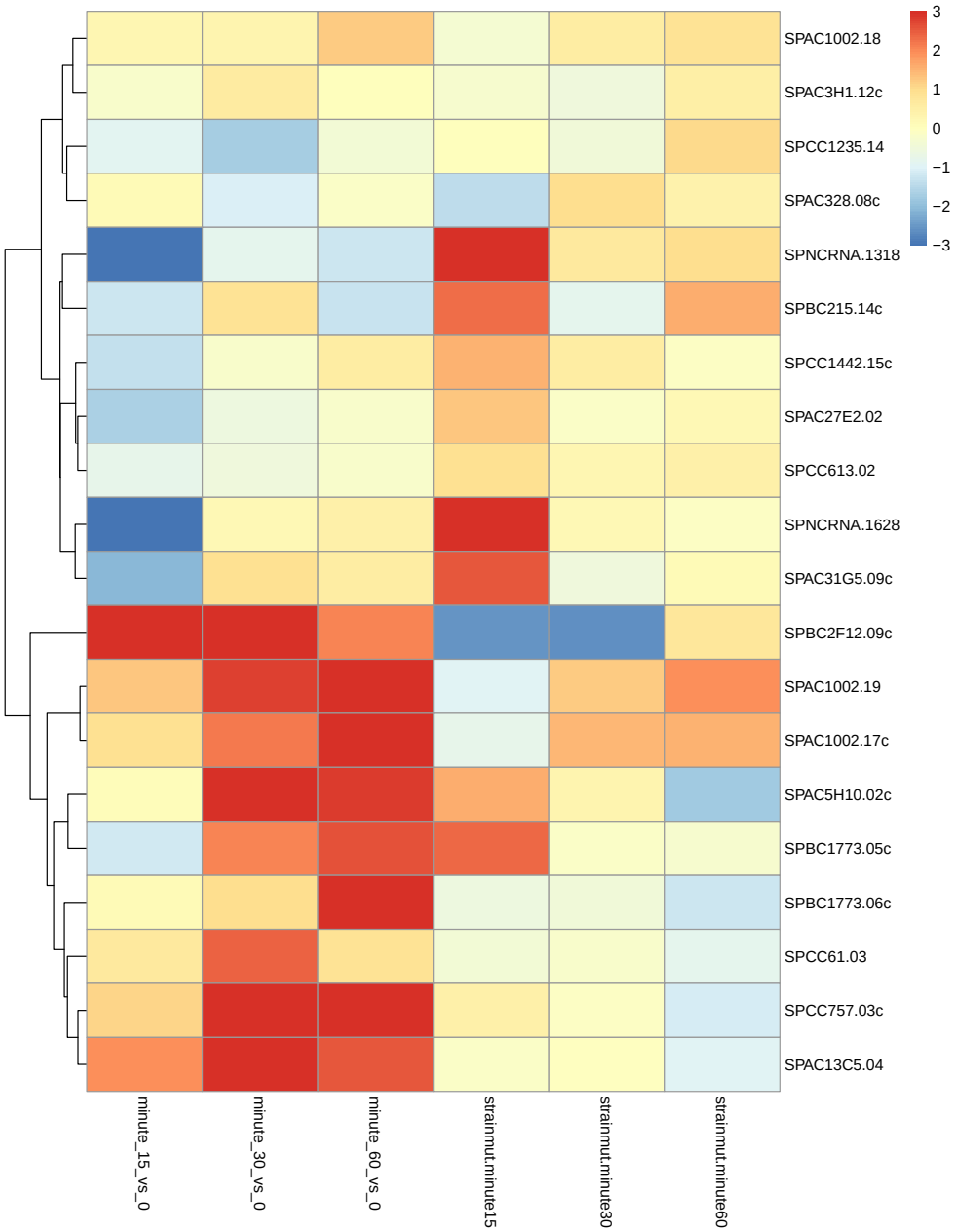
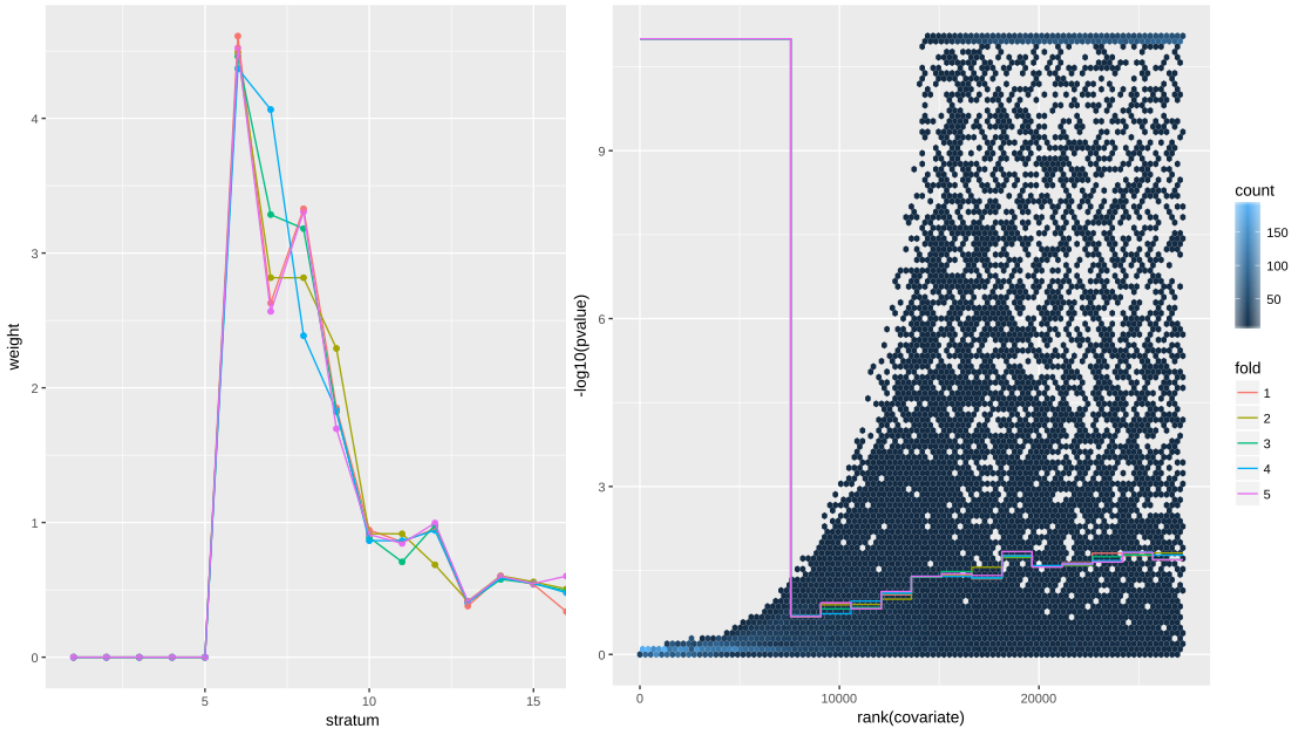



Figure 12: heatmap for Time Course Experiment

### 5.2.8 IHW

As a reminder, the first plot represents the different weights for the folds across the covariates, where genes with a high baseMean get higher weights than genes with low baseMeans. And the second plot shows the folds on the unweighted p-values. For the two plots on the contrast basal vs luminal, see Figure 13.

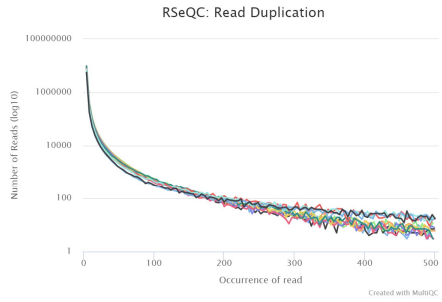



(a) Estimated weights (b) asd

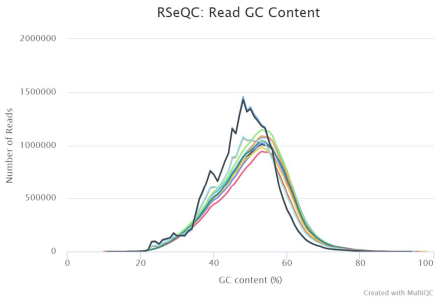
Figure 13: Decision boundaries for unweighted p-values

### 5.2.9 MultiQC and RSeQC plots

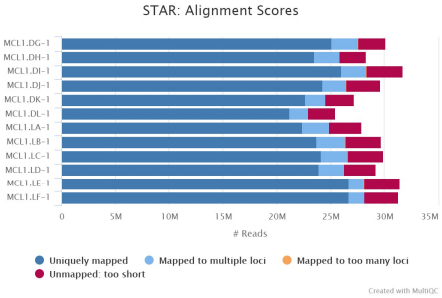
The plots created in RSeQC are all used for quality control. MultiQC then summarizes all of the RSeQC plots into one plot for each different part of the quality controls. In the following, we will give a breakdown of the plots:
Figure 14a shows how many alignment positions have exact duplicates.
Figure 14b shows the amount of GC (guanine-cytosine) content with respect to the number of reads.
Figure 14c shows how the reads have been or have not been mapped in regard to the samples, by using the STAR Aligner.
Figure 14d shows how the gene counts in regard to the samples, by using the STAR Aligner.
Figure 14e shows how the reads are mapped over the genome features.
Figure 14f shows the percentage of reads and read pairs that match the strandedness of overlapping transcripts. It can be used to infer whether RNA-seq library preps are stranded.
Figure 14g shows the compared, detected splice junctions to a reference gene model. An RNA read can be spliced 2 or more times, each time is called a splicing event.
Figure 14h shows the number of known splicing junctions that are observed in each dataset. If sequencing depth is sufficient, all (annotated) splice junctions should be rediscovered, resulting in a curve that reaches a plateau. Missing low abundance splice junctions can affect downstream analysis.



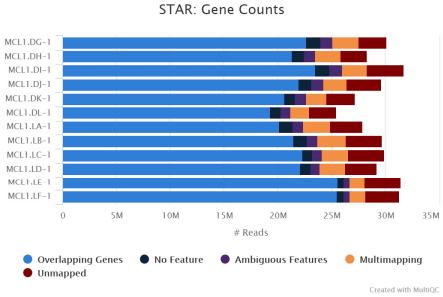

(a)



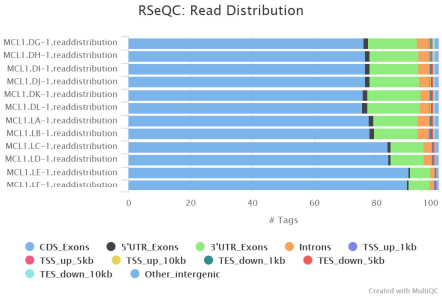

(b)



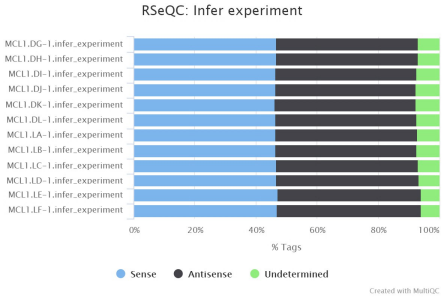

(c)



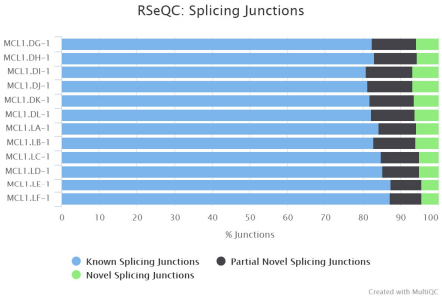

(d)



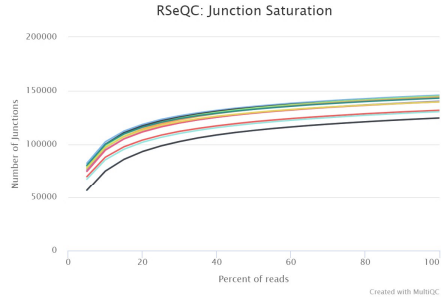

(e)




(f)




(g)




(h)

Figure 14: MultiQC plots for RSeQC and STAR results

### 5.2.10 Shrinkage

The shrinkage option only applies to the results created from DESeq2. In our example run those results are given by the MA plot. To give an example of shrinkage using the ashr option and the same contrasts as used in the MA plot with normal shrinkage, see Figure 15.

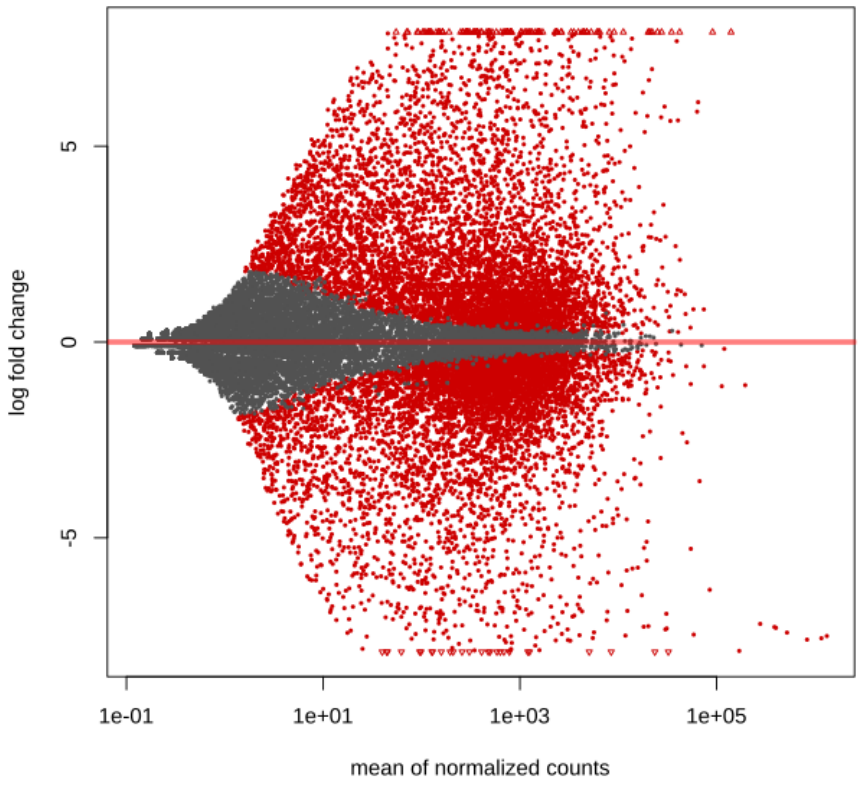



Figure 15: basal vs luminal

### 5.2.11 Formulas and Contrasts

As mentioned earlier, we now give examples for formulas and the resulting contrasts that could be specified on the example data.
If we want to create contrasts for the **status** column we would choose a formula, such as *~status*. As a result of this, we can now set contrasts between the unique effects of the column, i.e. *virgin vs pregnant*, *virgin vs lactate* and *pregnant vs lactate*. Further, if we want contrasts for multiple conditions we could use a formula, such as *~status*condition* or equally *~status+condition+status:condition*. Possible contrasts would be the ones we just saw, i.e. *virgin vs pregnant*, *virgin vs lactate* and *pregnant vs lactate*, as well as additionally the contrasts for the **condition** column, i.e. *basal vs luminal*. Further, possible contrasts should now be available because of the interaction term *status:condition*, but in reality, the interaction terms now result in a

string of condition and effect from the interactions, i.e. ***condition****basal.****status****virgin* would be such a name. We decided against the use of names and can use the grouping option for such a case. When using the grouping options, we now are able to get the same interaction, i.e. *basalvirgin vs luminalvirgin* instead of ***condition****basal.****status****virgin*.

# 6 Discussion and Outlook

## 6.1 Limitations for experienced users

While we extended the pipeline with a wide variety of options for an inexperienced user, there are still plenty of options for experienced users, that we did not implement. In fact, the DESeq2 package provides many more parameters an experienced user might want to use to do more advanced analysis or create other forms of output. Conversely, the challenging part with extending the pipeline further by adding options for experienced users is to reach the point, where we need to separate between the usefulness of adding parameter options to a pipeline because a greatly experienced user might decide not to utilize the pipeline at all.

## 6.2 Limitations on plots

As of now, we mainly focus on plots as a way for quality assessment, whereas it can be interesting to have more plots, such as volcano plots to see differential expression across samples or others for independent hypothesis weighting. In conclusion, there are many more plots that can be implemented for the purpose of illustrating results.

## 6.3 Limitations on the input

The current pipeline is limited to the use of providing a gtf file for annotations, while most annotation files are only available as gff or gff3 files. Therefore, another addition would be to provide a script to check and convert the annotations into the gtf format, if necessary. Further, STAR does not necessarily need annotation files, and therefore there should be an option to be able to omit them.

## 6.4 Other high throughput sequencing input

The DESeq2 paper [17] states, that the used statistical methods are also applicable for other comparative HTS assays, i.e. ChIP (chromatin immunoprecipitation) sequencing, chromosome conformation capture, or counting observed taxa in metagenomic studies. Therefore, the scope of the current pipeline could be extended in a way, to use the existing tools and packages and adding others to take care of other comparative HTS assays and allowing to choose between them.

# 7 Source Code

The source code for this pipeline can be found here and is currently going through the progress of a PR (Pull Request): DOI for the pipeline [12].
Direct link to the release:
https://github.com/Jukre111/rna-seq-star-deseq2/releases/tag/v1.1
The link was last visited on March 11, 2020.

# 8 Acknowledgements

I would like to express my gratitude to Prof. Dr. Gunnar Klau for the opportunity to write this bachelor thesis. I also want to thank my advisor Philipp Spohr, for his advice and the support he provided.

## References


[1] Anaconda Software Distribution. *Computer software*. Version Vers. 2-2.4.0.

[2] Yoav Benjamini and Yosef Hochberg. "Controlling the false discovery rate: a practical and powerful approach to multiple testing". In: *Journal of the Royal statistical society: series B (Methodological)* 57.1 (1995), pp. 289–300.

[3] Alexander Dobin, Carrie A Davis, Felix Schlesinger, Jorg Drenkow, Chris Zaleski, Sonali Jha, Philippe Batut, Mark Chaisson, and Thomas R Gingeras. "STAR: ultrafast universal RNA-seq aligner". In: *Bioinformatics* 29.1 (2013), pp. 15–21.

[4] Ron Edgar, Michael Domrachev, and Alex E Lash. "Gene Expression Omnibus: NCBI gene expression and hybridization array data repository". In: *Nucleic acids research* 30.1 (2002), pp. 207–210.

[5] Nils Eling, Arianne C Richard, Sylvia Richardson, John C Marioni, and Catalina A Vallejos. "Correcting the mean-variance dependency for differential variability testing using single-cell RNA sequencing data". In: *Cell systems* 7.3 (2018), pp. 284–294.

[6] Philip Ewels, Måns Magnusson, Sverker Lundin, and Max Käller. "MultiQC: summarize analysis results for multiple tools and samples in a single report". In: *Bioinformatics* 32.19 (2016), pp. 3047–3048.

[7] Nai Yang Fu, Anne C Rios, Bhupinder Pal, Rina Soetanto, Aaron TL Lun, Kevin Liu, Tamara Beck, Sarah A Best, François Vaillant, Philippe Bouillet, et al. "EGF-mediated induction of Mcl-1 at the switch to lactation is essential for alveolar cell survival (data accessible at NCBI GEO database (Edgar et al., 2002), accession GSE60450)". In: *Nature Cell Biology* 17.4 (2015), pp. 365–375.

[8] Wolfgang Huber, Anja Heydebreck, Holger Sültmann, Annemarie Poustka, and Martin Vingron. "Parameter estimation for the calibration and variance stabilization of microarray data". In: *Statistical applications in genetics and molecular biology* 2 (Feb. 2003), Article3.

[9] Wolfgang Huber, Anja Von Heydebreck, Holger Sültmann, Annemarie Poustka, and Martin Vingron. "Variance stabilization applied to microarray data calibration and to the quantification of differential expression". In: *Bioinformatics* 18.suppl_1 (2002), S96–S104.

[10] Nikolaos Ignatiadis, Bernd Klaus, Judith B Zaugg, and Wolfgang Huber. "Data-driven hypothesis weighting increases detection power in genome-scale multiple testing". In: *Nature methods* 13.7 (2016), p. 577.

[11] Raivo Kolde. *pheatmap: Pretty Heatmaps*. R package version 1.0.12. 2019.

[12] Johannes Köster, Julian Kremer, Sebastian Schmeier, and matrs. *Jukre111/rna-seq-star-deseq2: Update for multiple conditions, easier usability, add plots*. https://doi.org/10.5281/zenodo.3704568. 2020.

[13] Johannes Köster and Sven Rahmann. "Snakemake—a scalable bioinformatics workflow engine". In: *Bioinformatics* 28.19 (2012), pp. 2520–2522.

[14] Johannes Köster, Sebastian Schmeier, and Jose Maturana. *Snakemake workflow: rna-seq-star-deseq2*. https://github.com/snakemake-workflows/rna-seq-star-deseq2/. 2019.

[15] Leong, H. S., Dawson, K., Wirth, C., Li, Y., Connolly, Y., Smith, D. L., Wilkinson, C. R., Miller, and C. J. “A global non-coding RNA system modulates fission yeast protein levels in response to stress”. In: *Nat Commun* 5 (2014), p. 3947.

[16] Michael I Love, Simon Anders, Vladislav Kim, and Wolfgang Huber. “differential expression [version 2; peer review: 2 approved]”. In: ().

[17] Michael I Love, Wolfgang Huber, and Simon Anders. “Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2”. In: *Genome biology* 15.12 (2014), p. 550.

[18] Marcel Martin. “Cutadapt removes adapter sequences from high-throughput sequencing reads”. In: *EMBnet. journal* 17.1 (2011), pp. 10–12.

[19] Matthew Stephens, Peter Carbonetto, David Gerard, Mengyin Lu, Lei Sun, Jason Willwerscheid, and Nan Xiao. *ashr: Methods for Adaptive Shrinkage, using Empirical Bayes*. R package version 2.2-47. 2020.

[20] Liguo Wang, Shengqin Wang, and Wei Li. “RSeQC: quality control of RNA-seq experiments”. In: *Bioinformatics* 28.16 (2012), pp. 2184–2185.