Output description for rnaseq-nf pipeline

rnaseq-nf is a bioinformatics best-practice analysis pipeline used for RNA-seq data. The pipeline is focused in counts and differential expression analyses.

Pipeline overview

The pipeline is built using Nextflow and processes data using the following steps:

- FastQC v0.11.8 - read quality control
- Trimmomatic v.0.38 - adapter and low quality trimming.
- STAR v2.6.1d - alignment
- RSeQC v3.0.0 - RNA quality control metrics
  - Bam stat
  - Clipping profile
  - Gene body coverage
  - Infer Experiment
  - Inner distance
  - Junction annotation
  - Junction saturation
  - Read distribution
  - Read duplication
- Preseq v2.0.3 - library complexity
- Picard v2.18.27 - Identify duplicate reads
- dupRadar v1.12.1 - technical / biological read duplication
- Subread(Featurecounts) v1.6.4 - gene counts, biotype counts, rRNA estimation.
- StringTie v1.3.5 - FPKMs for genes and transcripts
- edgeR v3.24.1 - create MDS plot and sample pairwise distance heatmap / dendrogram
- MultiQC v1.7 - aggregate report, describing results of the whole pipeline
- Custom analysis:
  - DESeq2 v1.18.1 - Differential expression analysis and plots

Preprocessing

FastQC

FastQC [1] gives general quality metrics about your reads. It provides information about the quality score distribution across your reads, the per base sequence content (%T/A/G/C). You get information about adapter contamination and other over-represented sequences.

For further reading and documentation see the FastQC help.

Note:The FastQC plots displayed in the MultiQC report shows *untrimmed* reads. They may contain adapter sequence and potentially regions with low quality. To see how your reads look after trimming, look at the FastQC reports in the 02-preprocessing/FastQC directory.

Output directory: `01-fastqc/``

- `logs/`
  - Folder with the log files per sample of the FastQC.
- `zips/`
  - `{sample_id}_R[12]_fastqc.zip`
  - zip file containing the FastQC report, tab-delimited data file and plot images

Trimming

Trimmomatic [2] is used for removal of adapter contamination and trimming of low quality regions. Parameters included for trimming are:

- Nucleotides with phred quality < 10 in 3'end.
- Mean phred quality < 20 in a 4 nucleotide window.
- Read lenght < 50

MultiQC reports the percentage of bases removed by trimming in bar plot showing percentage or reads trimmed in forward and reverse.

Results directory: `02-preprocessing/`

- `FastQC/`
  - `{sample_id}_filtered_R[12].fastqc.html`: html report of the trimmed reads.
  - `{sample_id}_filtered_R[12].fastqc.html.zip`: zip compression of above file.

- `logs/`
    - Folder with the log files per sample of the FastQC.

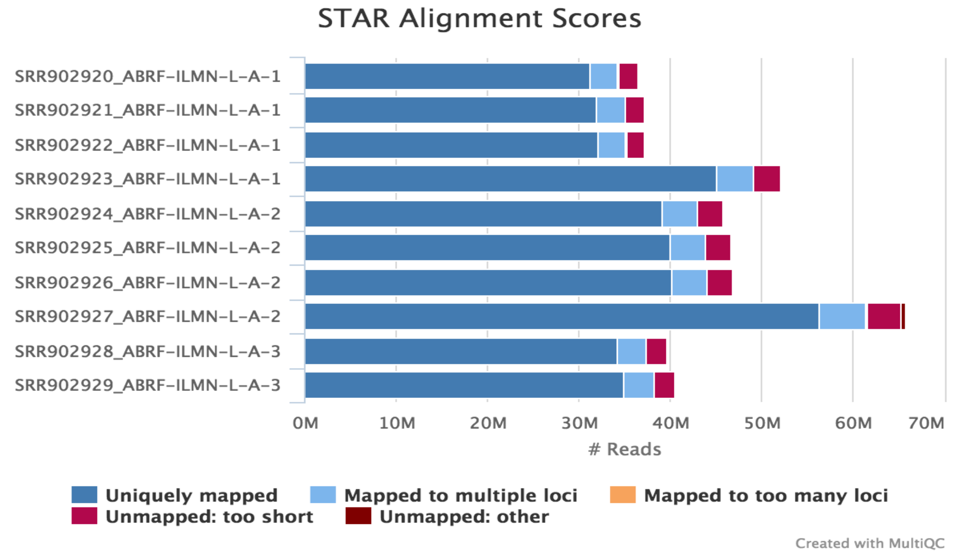
NOTE: Trimmed reads are not delivered to the researcher by default due to disk space issues. If you are interested in using them, please contact us and we will add them to your delivery.

Alignment

STAR

STAR [3] is a read aligner designed for RNA sequencing. STAR stands for Spliced Transcripts Alignment to a Reference, it produces results comparable to TopHat (the aligned previously used by NGI for RNA alignments) but is much faster.

The STAR section of the MultiQC report shows a bar plot with alignment rates: good samples should have most reads as *Uniquely mapped* and few *Unmapped* reads.



Output directory: 03-alignment/

- `{sample_id}_filteredAligned.sortedByCoord.out.bam`
    - The aligned BAM file
- `{sample_id}_filteredAligned.sortedByCoord.out.bam.bai`
    - The aligned BAM file index
- `logs/`
    - `{sample_id}_filteredLog.final.out`
    - The STAR alignment report, contains mapping results summary
    - `{sample_id}_filteredLog.out` and `{sample_id}filteredLog.progress.out`
    - STAR log files, containing a lot of detailed information about the run. Typically only useful for debugging purposes.
    - `{sample_id}_filteredSJ.out.tab`
    - Filtered splice junctions detected in the mapping

Alignment Quality Control

RSeQC

RSeQC [4] is a package of scripts designed to evaluate the quality of RNA seq data. You can find out more about the package at the RSeQC website.

This pipeline runs several, but not all RSeQC scripts. All of these results are summarized within the MultiQC report and described below.

Output directory: `04-rseqc/`

These are all quality metrics files and contains the raw data used for the plots in the MultiQC report. In general, the `.r` files are R scripts for generating the figures, the `.txt` are summary files, the `.xls` are data tables and the `.pdf` files are summary figures.

Bam stat

Output: bam_stat/

This script gives numerous statistics about the aligned BAM files produced by STAR. A typical output looks as follows:

From the `{sample_id}.bam_stat.txt` file:

```
#Output (all numbers are read count)
#==================================================
Total records:                                  41465027
QC failed:                                      0
Optical/PCR duplicate:                          0
Non Primary Hits                                8720455
Unmapped reads:                                 0

mapq < mapq_cut (non-unique):                   3127757
mapq >= mapq_cut (unique):                      29616815
Read-1:                                         14841738
Read-2:                                         14775077
Reads map to '+':                               14805391
Reads map to '-':                               14811424
Non-splice reads:                               25455360
Splice reads:                                   4161455
Reads mapped in proper pairs:                   21856264
Proper-paired reads map to different chrom:     7648
```

MultiQC plots each of these statistics in a dot plot. Each sample in the project is a dot - hover to see the sample highlighted across all fields.

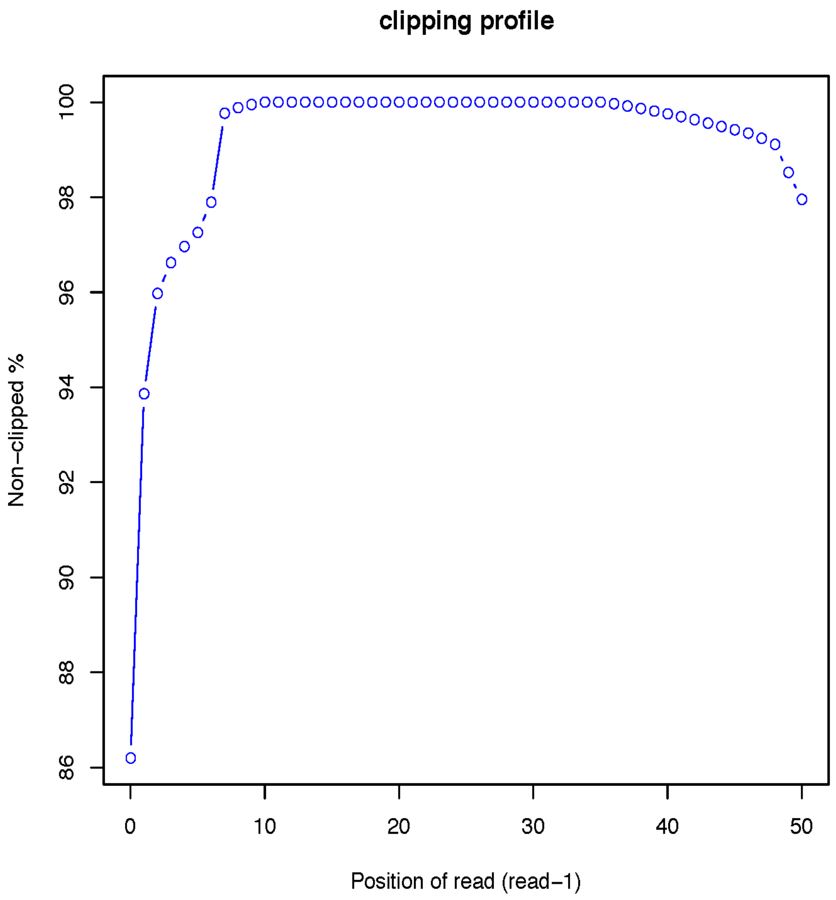
RSeQC documentation: bam_stat.py

Clipping Profile

Output directory: clipping_profile/

This program is used to estimate clipping profile of RNA-seq reads from BAM or SAM file. Note that to use this function, CIGAR strings within SAM/BAM file should have ‘S’ operation (This means your reads aligner should support clipped mapping).

- `data/`
    - `{sample_id}.clipping_profile.xls`
    - Contains 3 columns: the first column is position (starting from 0) of read in 5’->3’ direction; the second column is the number of reads clipped at this position; the third column is the number of reads non-clipped at this position.
- `plots/`
    - `{sample_id}.clipping_profile.pdf`
    - PDF with the following graph(s)



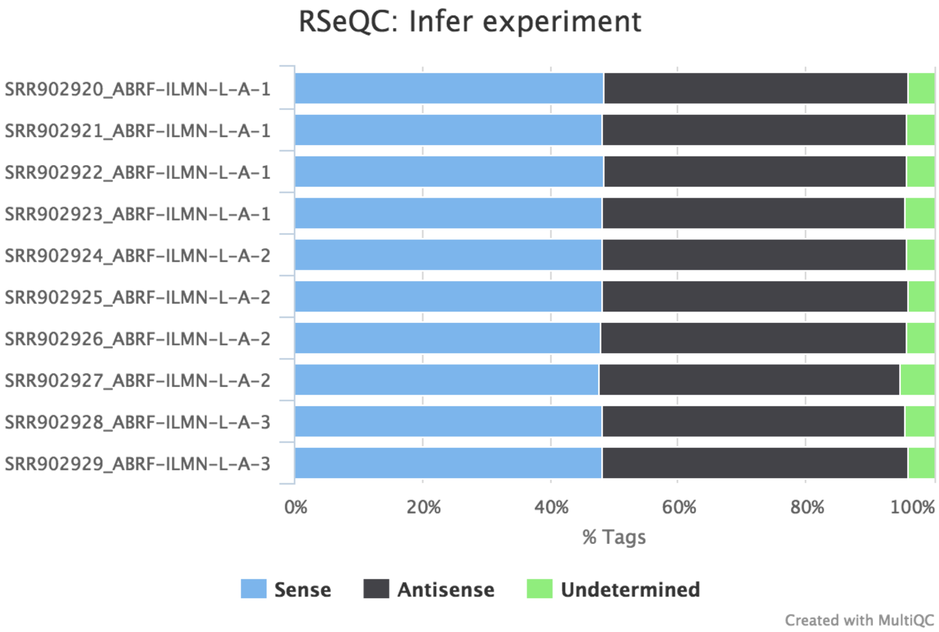
RSeQC documentation: [clipping_profile.py] http://rseqc.sourceforge.net/#clipping-profile-py

Infer Experiment

Output: infer_experiment/

This script predicts the mode of library preparation (sense-stranded or antisense-stranded) according to how aligned reads overlay gene features in the reference genome. Example output from an unstranded (~50% sense/antisense) library of paired end data:

From MultiQC report:



From the `{sample_id}.infer_experiment.txt` file:

```
This is PairEnd Data
Fraction of reads failed to determine: 0.0409
Fraction of reads explained by "1++,1--,2+-,2-+": 0.4839
Fraction of reads explained by "1+-,1-+,2++,2--": 0.4752
```

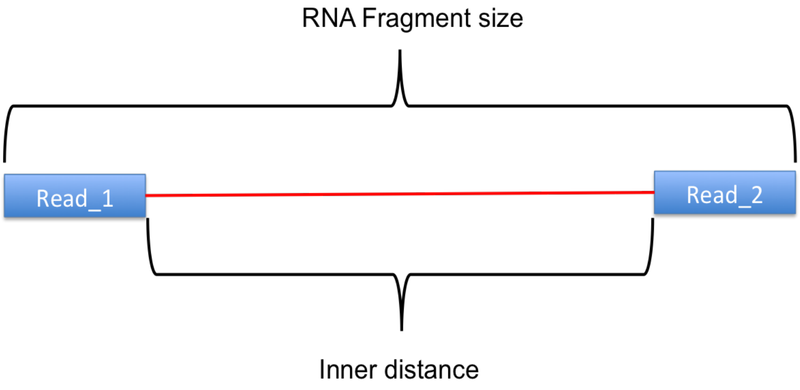
RSeQC documentation: infer_experiment.py

Inner distance

Only for paired-end data. The inner distance script tries to calculate the inner distance between two paired RNA reads. It is the distance between the end of read 1 to the start of read 2, and it is sometimes confused with the insert size (see this blog post).

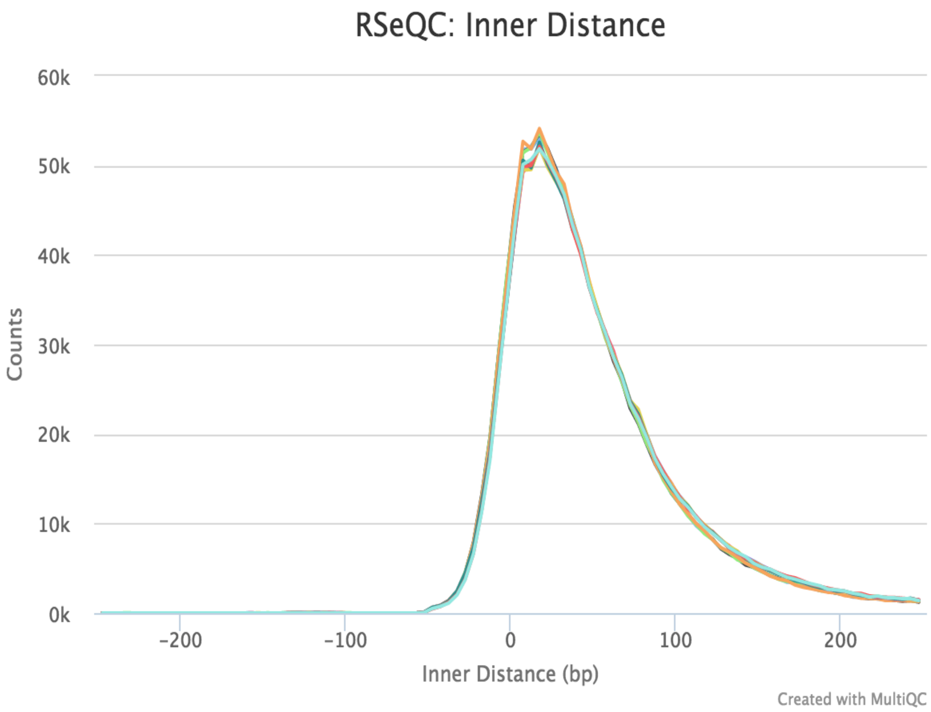
Output: `inner_distance/`

- `{sample_id}.inner_distance.txt`
    - First column is read ID
    - Second column is inner distance. Could be negative value if PE reads were overlapped or mapping error (e.g. Read1_start < Read2_start, while Read1_end >> Read2_end due to spliced mapping of read1).
    - Third column indicates how paired reads were mapped: PE_within_same_exon, PE_within_diff_exon,PE_reads_overlap.
- `data/`
    - `{sample_id}.inner_distance_freq.txt`
    - Inner distance starts
    - Inner distance ends
    - Number of read pairs
    - Note the first 2 columns are left side half open interval
- `plots/`
    - `{sample_id}.inner_distance.pdf`
    - Histogram representing the previous txt file's information.



> *Credit: modified from RSeQC documentation.*

Note that values can be negative if the reads overlap. A typical set of samples may look like this:



This plot will not be generated for single-end data. Very short inner distances are often seen in old or degraded samples (*eg.* FFPE).

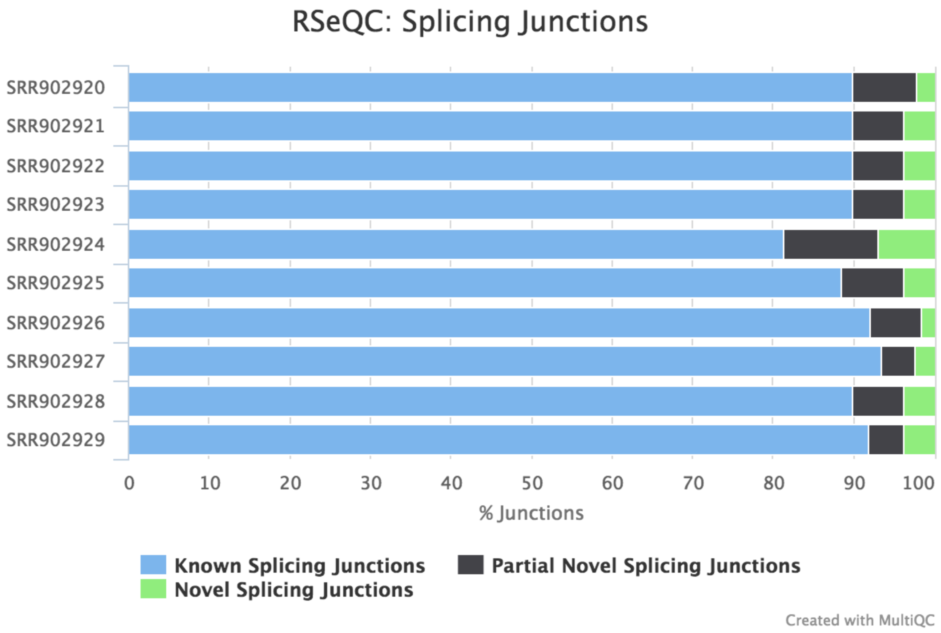
RSeQC documentation: inner_distance.py

Junction annotation

Junction annotation compares 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.

Output: `junction_annotation/`

- `logs/`
- `{sample_id}.junction_annotation_log.txt`
  - Log files of the analysis.
- `data/`
- `{sample_id}.junction.xls`
  - Data tables used to generate the PDF plots.
  - First column is chromosome ID.
  - Start position of junction (0 based)
  - End position of junction (1 based)
  - Number of splice events supporting this junction
  - 'Annotated', 'complete_novel' or 'partial_novel'
- `events/`
  - `{sample_id}.splice_events.pdf`
  - Shows the distribution of the splicing events found.
- `junctions/`
  - `{sample_id}.splice_junction.pdf`
  - Shows the distribution of the splicing junctions found.



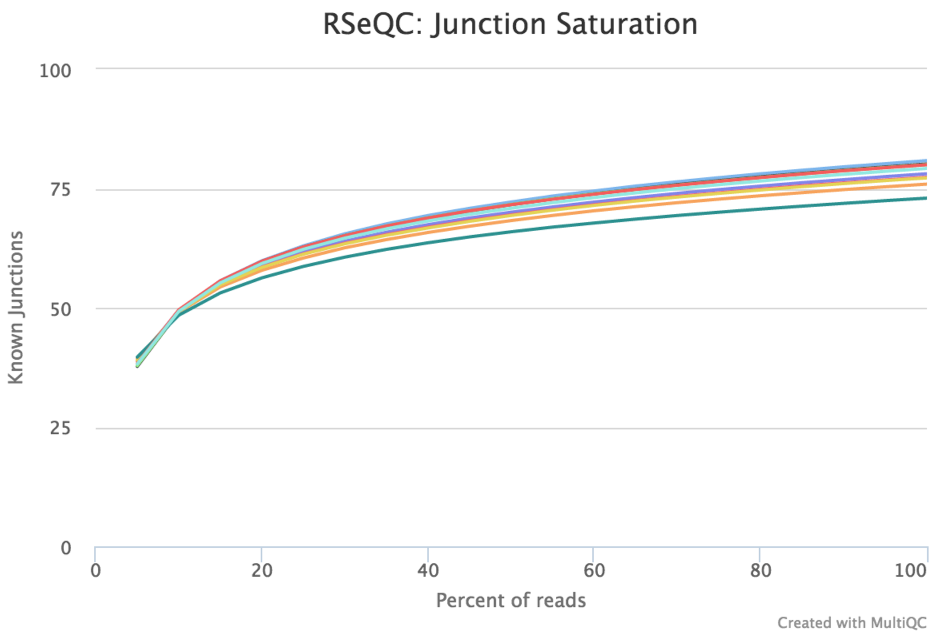
RSeQC documentation: junction_annotation.py

Junction saturation

Output: junction_saturation/

- `{sample_id}.junctionSaturation_plot.pdf`
  - Shows the number of splice sites detected at the data at various levels of subsampling. A sample that reaches a plateau before getting to 100% data indicates that all junctions in the library have been detected, and that further sequencing will not yield more observations. A good sample should approach such a plateau of *Known junctions*, very deep sequencing is typically requires to saturate all *Novel Junctions* in a sample.

None of the lines in this example have plateaued and thus these samples could reveal more alternative splicing information if they were sequenced deeper.



RSeQC documentation: junction_saturation.py

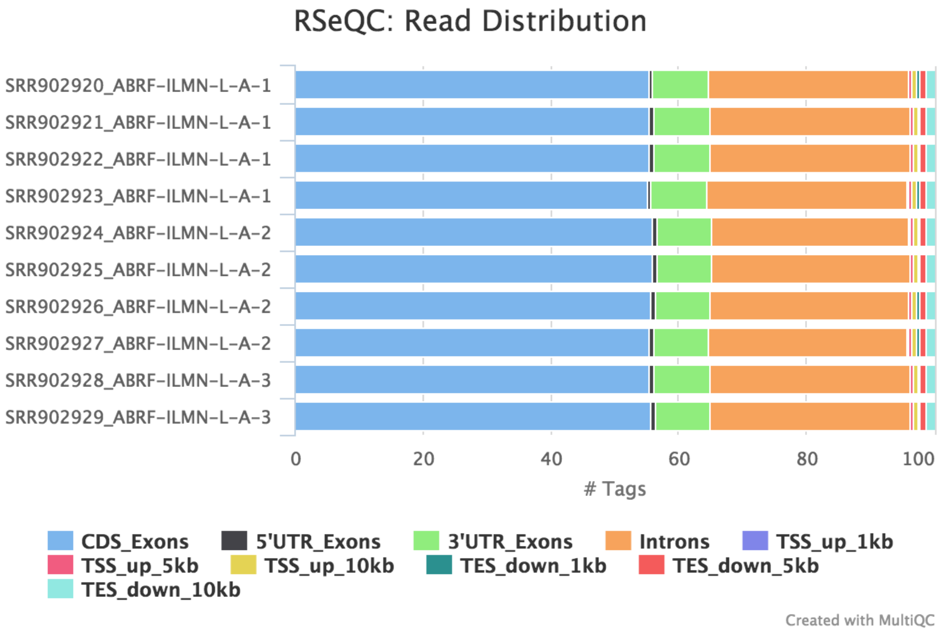
Read Distribution

Output:read_distribution/

This tool calculates how mapped reads are distributed over genomic features. A good result for a standard RNA seq experiments is generally to have as many exonic reads as possible (`CDS_Exons`). A large amount of intronic reads could be indicative of DNA contamination in your sample or some other problem.

- `{sample_id}.read_distribution.txt`: The ouput is a txt table with four columns:
  - Total_bases: This does NOT include those QC fail,duplicate and non-primary hit reads

- Tag_count: Number of tags that can be unambiguously assigned the 10 groups.
- Tags/Kb: Tags per kilobase

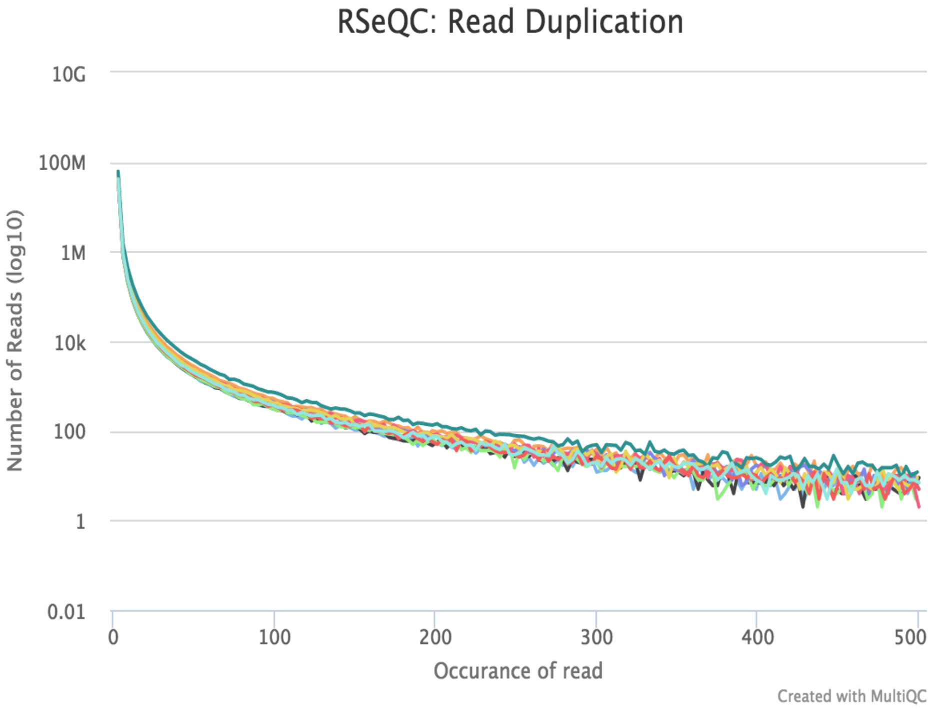


RSeQC documentation: read_distribution.py

Read duplication

Output:read_duplication/

- `{sample_id}.read_duplication.DupRate_plot.pdf`
    - This plot shows the number of reads (y-axis) with a given number of exact duplicates (x-axis). Most reads in an RNA-seq library should have a low number of exact duplicates. Samples which have many reads with many duplicates (a large area under the curve) may be suffering excessive technical duplication.
- `dup_pos/`
    - `dup_pos/{sample_id}.read_duplication.pos.DupRate.xls`
    - Table with the read duplication rate determined from mapping position of read. First column is “occurrence” or duplication times, second column is number of uniquely mapped reads.
- `dup_seq/`
    - `dup_seq/{sample_id}.read_duplication.seq.DupRate.xls`
    - Table with th read duplication rate determined from sequence of read. First column is “occurrence” or duplication times, second column is number of uniquely mapped reads.



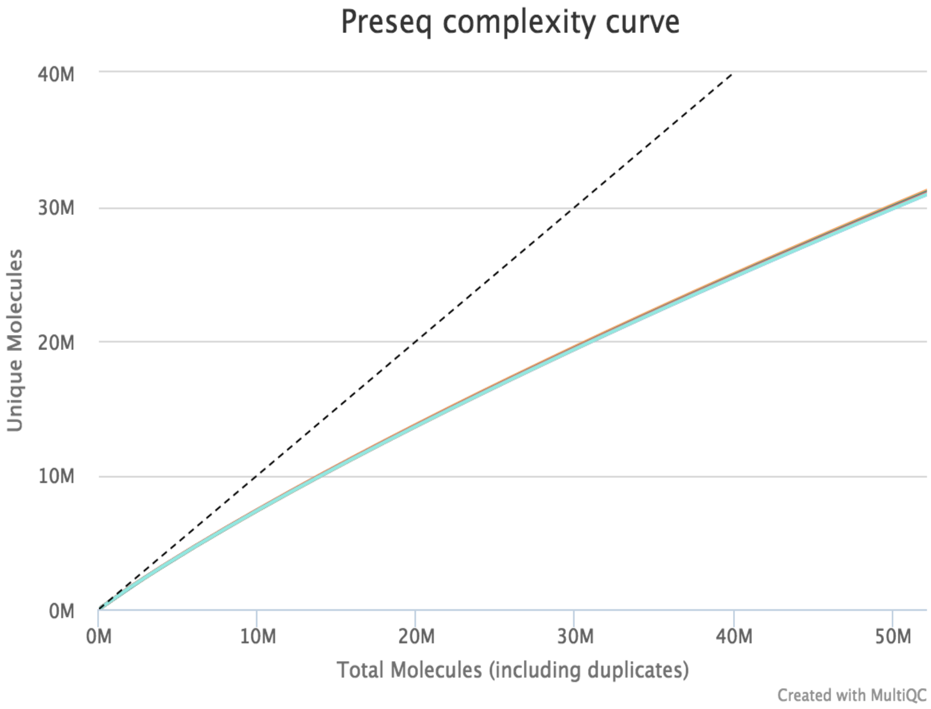
RSeQC documentation: read_duplication.py

Counts

Preseq

Preseq [5] estimates the complexity of a library, showing how many additional unique reads are sequenced for increasing the total read count. A shallow curve indicates that the library has reached complexity saturation and further sequencing would likely not add further unique reads. The dashed line shows a perfectly complex library where total reads = unique reads.

Note that these are predictive numbers only, not absolute. The MultiQC plot can sometimes give extreme sequencing depth on the X axis - click and drag from the left side of the plot to zoom in on more realistic numbers.



Output directory: `05-preseq/`

- `{sample_id}.ccurve.txt`
  - This file contains plot values for the complexity curve, plotted in the MultiQC report.

Picard

Picard [6] is a set of command line tools for manipulating high-throughput sequencing (HTS) data. In this case we used it to locate and tag duplicate reads in BAM files.

Output directory: `06-removeDuplicates/picard/`

- `metrics/`
  - `metrics/{sample_id}.markDups_metrics.txt`
  - Read duplication metrics.

Picard documentation: Picard docs

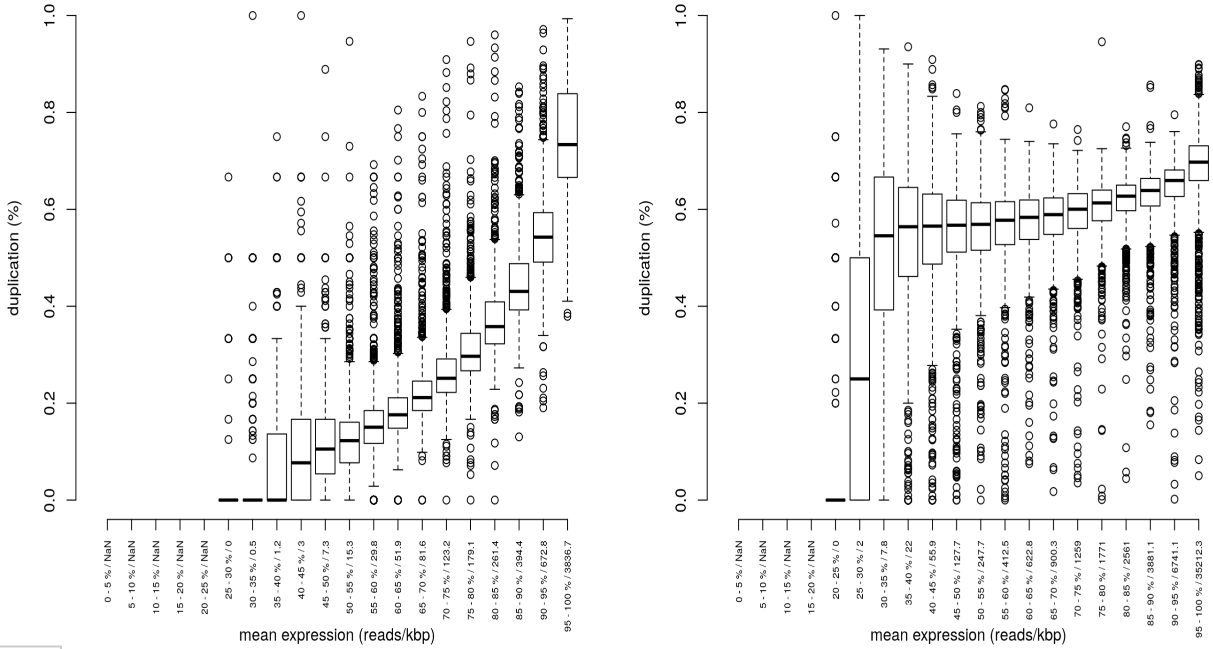
dupRadar

dupRadar [7] is a Bioconductor library for R. It plots the duplication rate against expression (RPKM) for every gene. A good sample with little technical duplication will only show high numbers of duplicates for highly expressed genes. Samples with technical duplication will have high duplication for all genes, irrespective of transcription level.

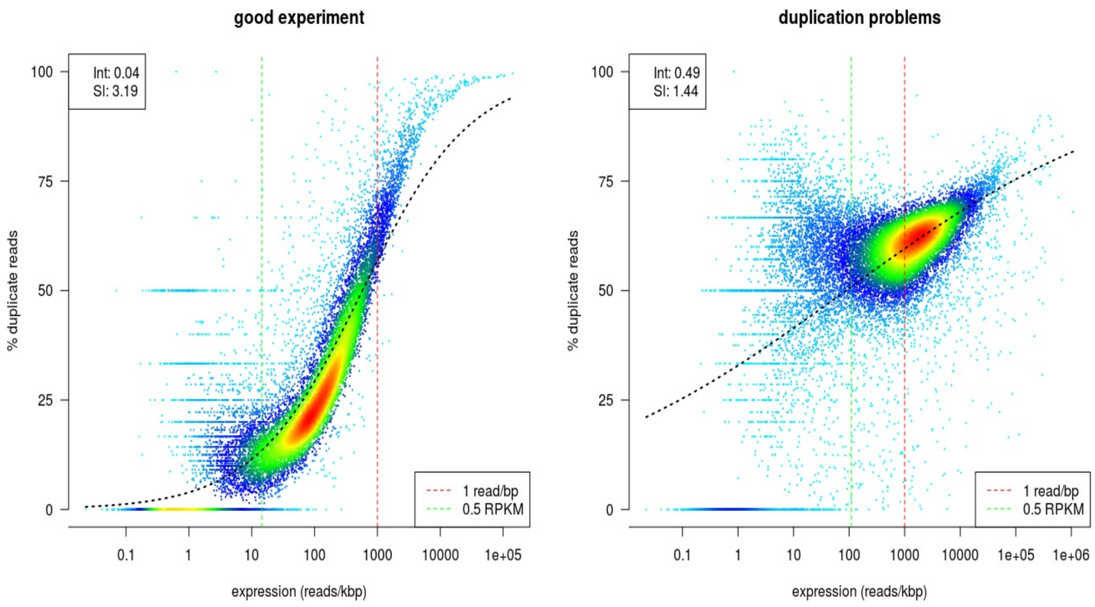
> *Credit: dupRadar documentation*

Output directory: `06-removeDuplicates/dupRadar/`

- `{sample_id}.markDups_dup_intercept_mqc.txt`
  - General stats file.
- `box_plots/`
  - `box_plots/{sample_id}.markDups_duprateExpBoxplot.pdf`
  - These box plots show the percentage of read duplication vs the mean expression. The boxplot in the left side represents a good experiment.



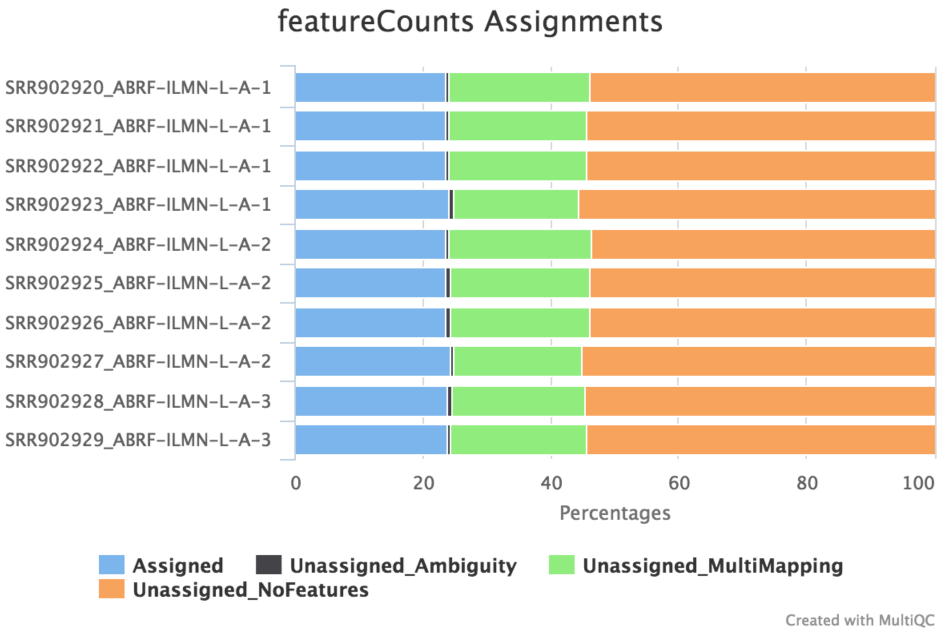
- `gene_data/`
  - `gene_data/{sample_id}.markDups_dupMatrix.txt`
  - The duplication matrix contains read counts in different scenarios and RPK and RPKM values for every gene.
- `histograms/`
  - `histograms/{sample_id}.markDups_expressionHist.pdf`
  - This histogram represents the distribution of RPK values per genes. This would help in identifying skewed distributions with unusual amount of lowly expressed genes, or to detect no consensus between replicates.
- `intercepts_slopes/`
  - `intercepts_slopes/{sample_id}.markDups_intercept_slope.txt`
- `scatter_curve_data/`
  - `scatter_curve_data/{sample_id}.markDups_duprateExpDensCurve_mqc.tx`
  - Provides duplication rate quality control for RNA-Seq datasets. Highly expressed genes can be expected to have a lot of duplicate reads, but high numbers of duplicates at low read counts can indicate low library complexity with technical duplication.
- `scatter_plots/`
  - `scatter_plots/{sample_id}.markDups.bam_duprateExpDens.pdf`
  - This plot relates the normalized number of counts per gene (RPK, as a quantification of the gene expression) and the fraction represented by duplicated reads. A good experiment is shown in the left pic.



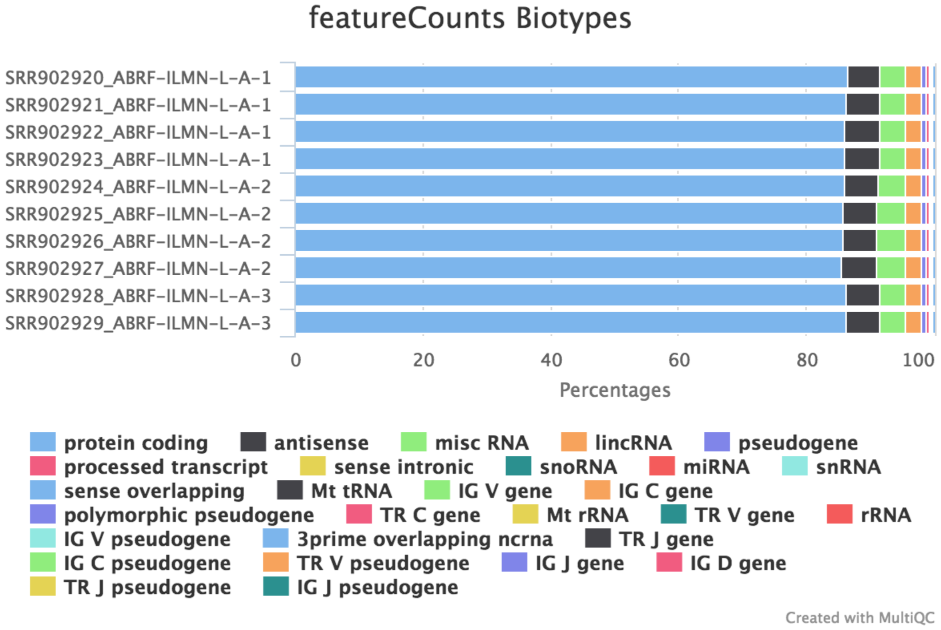
DupRadar documentation: dupRadar docs

featureCounts

featureCounts [8] from the subread package summarizes the read distribution over genomic features such as genes, exons, promotors, gene bodies, genomic bins and chromosomal locations. RNA reads should mostly overlap genes, so be assigned.



We also use featureCounts to count overlaps with different classes of features. This gives a good idea of where aligned reads are ending up and can show potential problems such as rRNA contamination.



Output directory: `07-featureCounts/`

- `biotype_counts/{sample_id}_biotype_counts_mqc.txt`
  - Read counts for the different gene biotypes that featureCounts distinguishes.
- `gene_counts/{sample_id}_gene.featureCounts.txt`
  - Read the counts for each gene provided in the reference `gtf` file
- `gene_counts_summaries/{sample_id}_gene.featureCounts.txt.summary`
  - Summary file, containing statistics about the counts
- `merged_gene_counts.txt`
  - File with gene counts merged between all samples.

StringTie

StringTie [9] assembles RNA-Seq alignments into potential transcripts. It assembles and quantitates full-length transcripts representing multiple splice variants for each gene locus.

StringTie outputs FPKM metrics for genes and transcripts as well as the transcript features that it generates.

Output directory: `08-stringtieFPKM/`

- `{sample_id}.gene_abund.txt`
  - Gene abundances, FPKM values.
- `ballgown/`

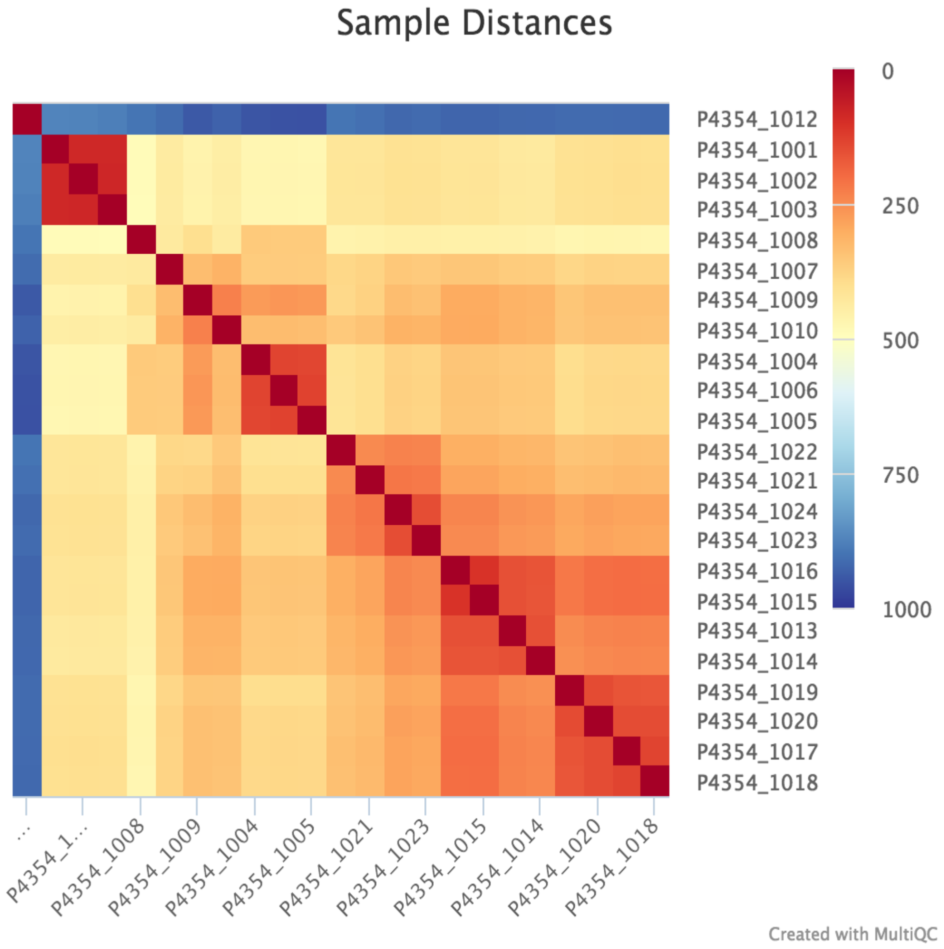
  - Folder with ballgown data for each sample. For more information about these files and its usage read this page
- `transcipts/`
  - `transcripts/{sample_id}_transcripts.gtf`
  - This `.gtf` file contains all of the assembled transcipts from StringTie
- `cov_refs/`
  - `{sample_id}.cov_refs.gtf`
  - This `.gtf` file contains the transcripts that are fully covered by reads.

Sample correlation and Differential Expression (DE)

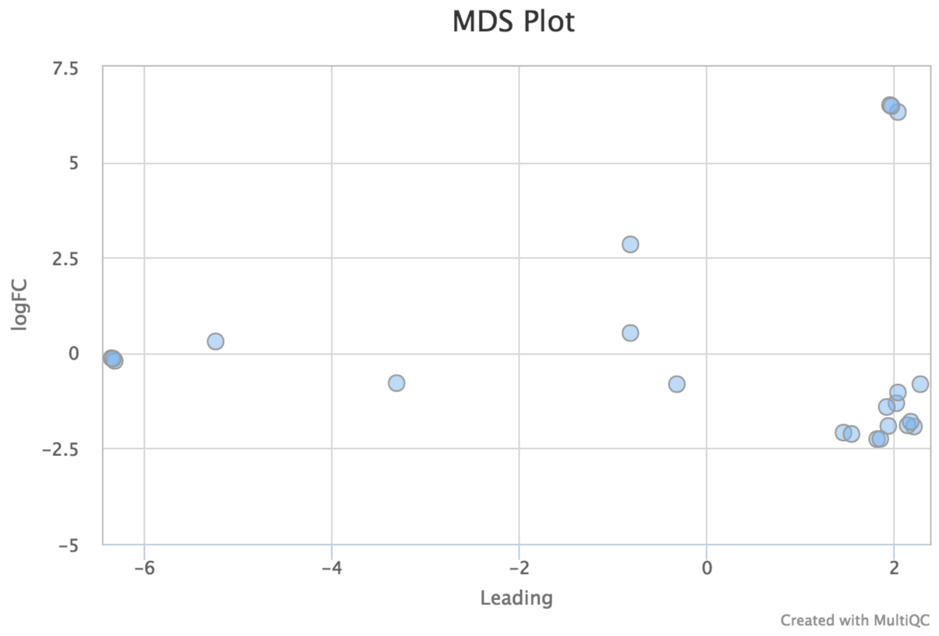
edgeR

Sample correlation with edgeR. edgeR [10] is a Bioconductor package for R used for RNA-seq data analysis. The script included in the pipeline uses edgeR to normalize read counts and create a heatmap / dendrogram showing pairwise euclidean distance (sample similarity). It also creates a 2D MDS scatter plot showing sample grouping. These help to show sample similarity and can reveal batch effects and sample groupings.

Heatmap:



MDS plot:



Output directory: `09-sample_correlation/`

- `edgeR_MDS_plot.pdf`
    - MDS scatter plot, showing sample similarity
- `edgeR_MDS_distance_matrix.txt`
    - Distance matrix containing raw data from MDS analysis
- `edgeR_MDS_plot_coordinates_mqc.txt`
    - Scatter plot coordinates from MDS plot, used for MultiQC report
- `log2CPM_sample_distances_dendrogram.pdf`
    - Dendrogram plot showing the euclidian distance between your samples
- `log2CPM_sample_distances_heatmap.pdf`
    - Heatmap plot showing the euclidian distance between your samples
- `log2CPM_sample_distances_mqc.csv`
    - Raw data used for heatmap and dendrogram plots.

Final reports

MultiQC

MultiQC [11] is a visualization tool that generates a single HTML report summarizing all samples in your project. Most of the pipeline QC results are visualized in the report and further statistics are available in within the report data directory.

The pipeline has special steps which allow the software versions used to be reported in the MultiQC output for future traceability.

Output directory: `99-stats/MultiQC/`

- `multiqc_report.html`
    - A standalone HTML file that can be viewed in your web browser
- `multiqc_data/`
    - Directory containing parsed statistics from the different tools used in the pipeline

For more information about how to use MultiQC reports, see http://multiqc.info

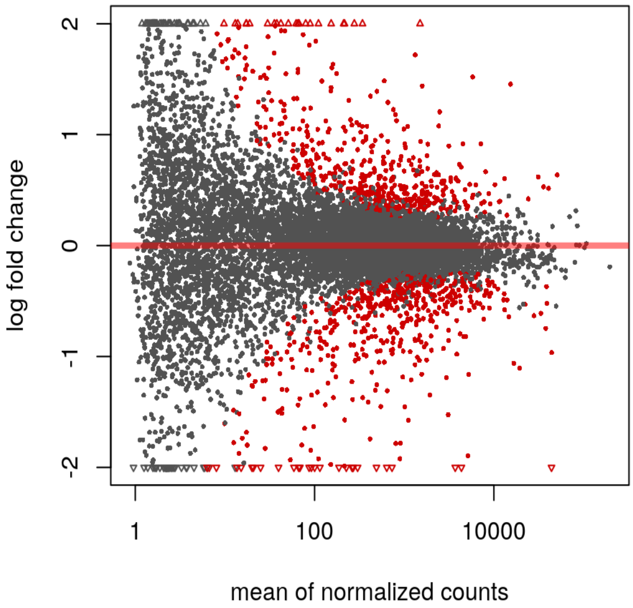
Custom analysis

This part contains the customized analyses realized for the researcher specifically for this analysis.

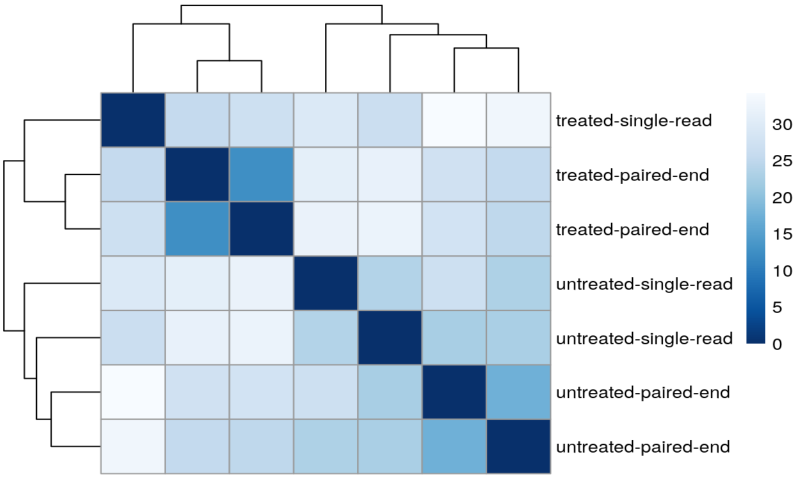
DESeq2

Differential expression analysis with DESeq2. DESeq2 [12] is a Bioconductor package for R used for RNA-seq data analysis. The script included in the pipeline uses DESeq2 to normalize read counts and create a heatmap / dendrogram showing pairwise euclidean distance (sample similarity). It also creates other plots to evaluate the sample dispersion. It also provides PCA plots to evaluate sample grouping.

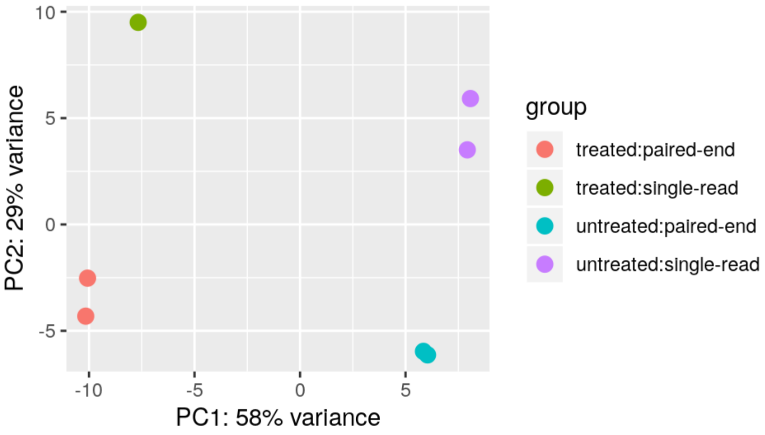
MA plot



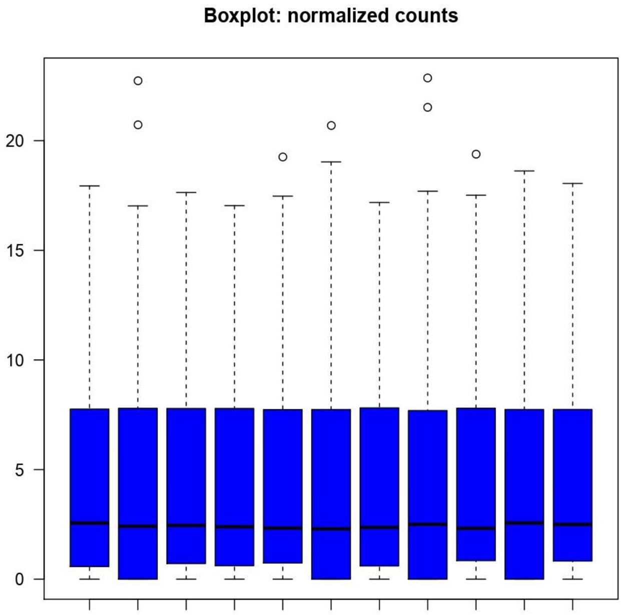
Sample to sample heatmap



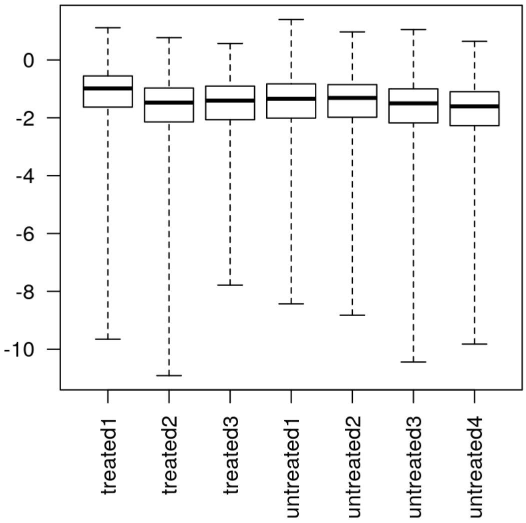
PCA plot



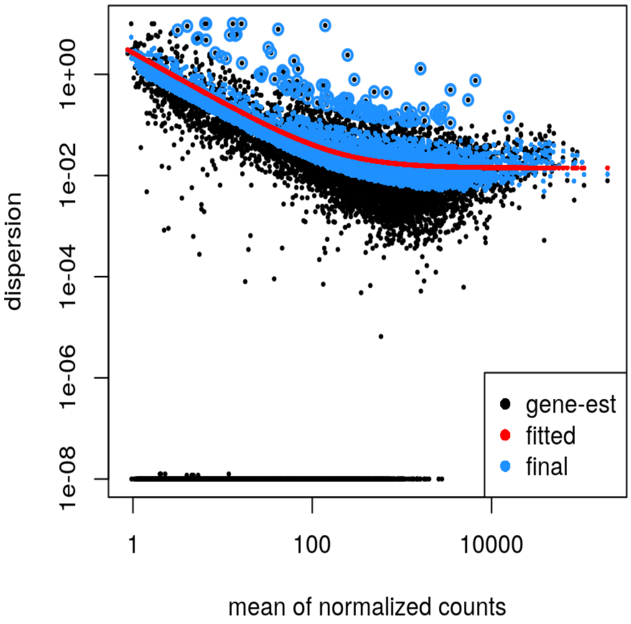
Normalized Boxplot



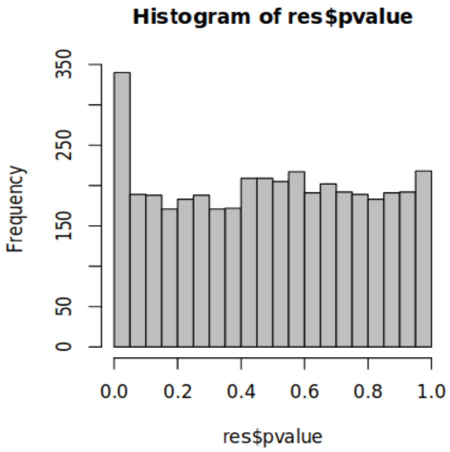
Cook Boxplot



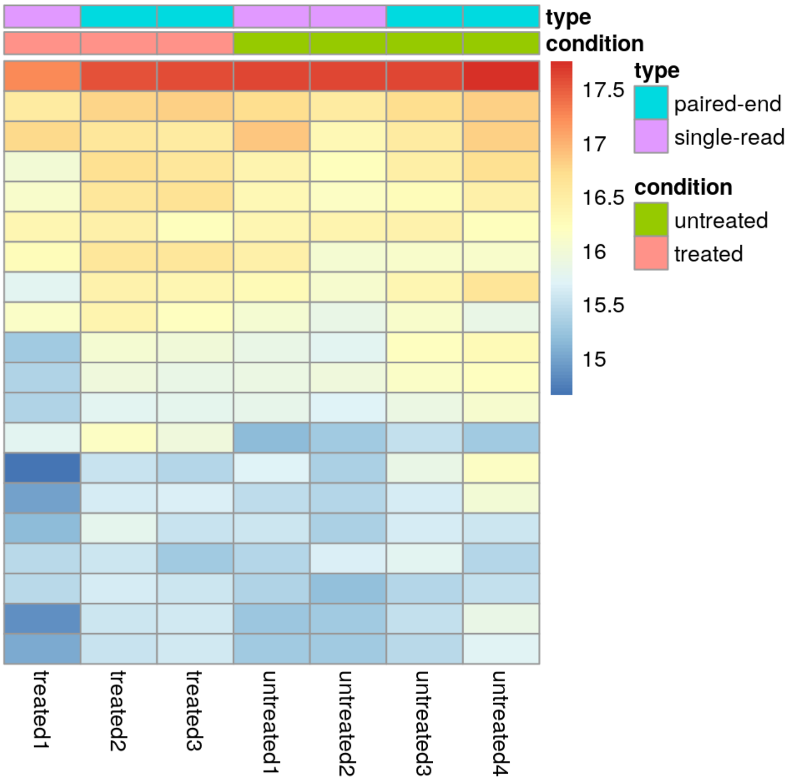
Dispersion Estimate



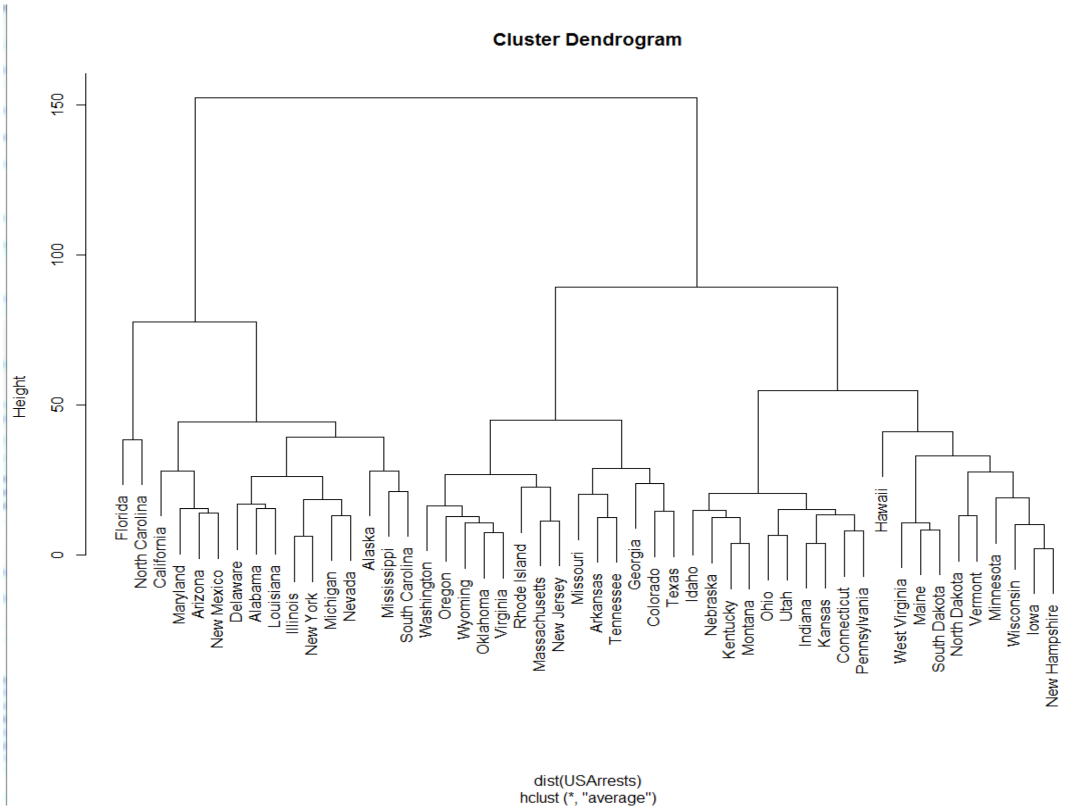
Pvalue test histogram



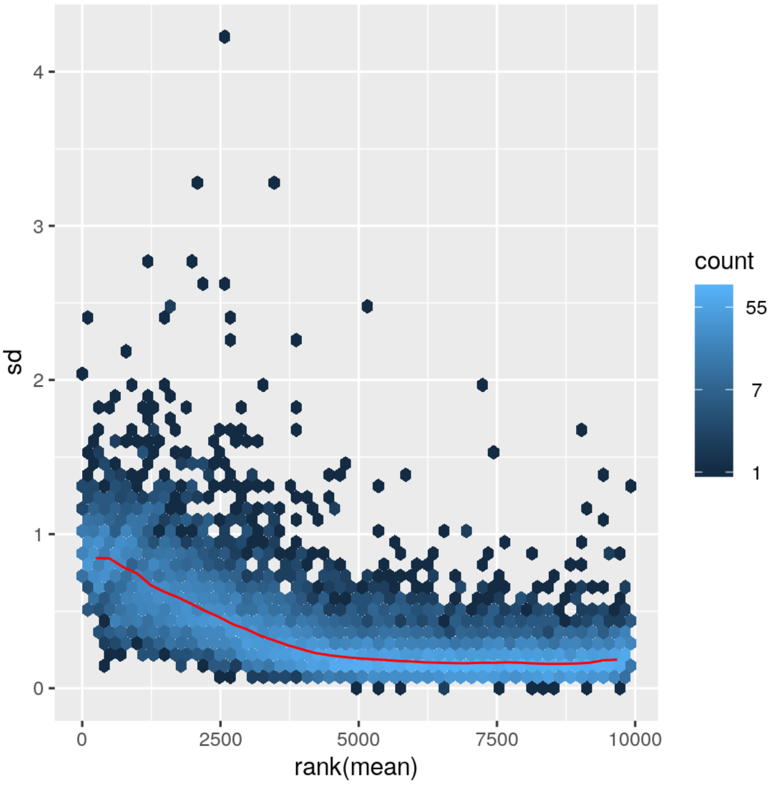
Top20 genes heatmap



Hierarchical clustering



Standard desviation curve



Output directory: `11-DESeq2/{comaprison_folder}/`

- `DE_matrix.txt`
  - Comparative table with the differential expression of two conditions.
- `DE_matrix_genes.txt`
  - Comparative table with the differential expression of two conditions with the second column being the gene_name.
- `clinical_data.txt`
  - Table with the metadata used to compare the samples.
- `maPlot_all.pdf`
  - MA plot of the DESeq analysis results for all the samples
- `heatmap_sample_to_sample.pdf`
  - Heatmap with the euclidean distance between samples.
- `plotPCA.pdf`
  - PCA plot of the samples for the rlog and the vsd.

    - rlog refers to the regularized log transformation, which transforms the count data to the log2 scale in a way which minimizes differences between samples for rows with small counts, and which normalizes with respect to library size.
    - vsd refers to variance stabilizing transformation (VST), which calculates a variance stabilizing transformation (VST) from the fitted dispersion-mean relation(s) and then transforms the count data (normalized by division by the size factors or normalization factors), yielding a matrix of values which are now approximately homoskedastic (having constant variance along the range of mean values). The transformation also normalizes with respect to library size.
- `boxplot.pdf`
  - PDF file with the box_plots
    - Box plot of the normalized Counts
    - Box plot of the counts cook distances to see if one sample is consistently higher than others.
- `plotDispersions.pdf`
  - PDF file with plots to analyze the dispersion of the samples
    - Dispersion calc is the per-gene dispersion estimate together with the fitted mean-dispersion relationship.
    - Histogram with the test of the differential expression pvalues
- `cluster_dendrogram.pdf`
  - PDF file with the hierarchical clustering of the samples. The input data comes from the normalization of the counts. For the normalization DESeq uses the normalization of the ratios where the counts are divided by sample-specific size factors determined by median ratio of gene counts relative ro geometric mean per gene.
- `heatmapCount_top20.pdf`
  - Heat map of the top 20 genes with the higher normalized mean count. The normalization is the same that the one of the hierarchical clustering.
- `heatmapCount_all_genes.pdf`
  - Heatmap of the normalized counts of all the genes.
- `plotSD.pdf`
  - Standard deviation of the transformed data, across samples, against the mean, using the shifted logarithm transformation.--->

References

1. web FASTQC P. Babraham Bioinformatics - FastQC. A Quality Control tool for High Throughput Sequence Data. 2012 trimming_reference
2. Bolger AM, Lohse M, Usadel B. Trimmomatic: a flexible trimmer for Illumina sequence data. Bioinformatics. 2014 Apr 28;30(15):2114–20.
3. Dobin A, Davis CA, Schlesinger F, Drenkow J, Zaleski C, Jha S, Batut P, Chaisson M, Gingeras TR. STAR: ultrafast universal RNA-seq aligner. Bioinformatics. 2013 Jan 1;29(1):15-21.
4. Wang, L., Wang, S., Li, W. RSeQC: quality control of RNA-seq experiments. Bioinformatics. 2012 Aug 15;28(16):2184-5.
5. web Preseq The Smith Lab. Computational Genomics Research.
6. web Picard toolkit Broad Institute.
7. Sayols S, Scherzinger D, Klein H. dupRadar: a Bioconductor package for theassessment of PCR artifacts in RNA-Seqdata BMC Bioinformatics (2016) 17:428.
8. Liao Y, Smyth GK, Shi W. featureCounts: an efficient general-purpose program for assigning sequence reads to genomic features. Bioinformatics, 30(7):923-30, 2014
9. Pertea M, Pertea GM, Antonescu CM, Chang TC, Mendell JT, Salzberg SL. StringTie enables improved reconstruction of a transcriptome from RNA-seq reads. Nat Biotechnol. 2015 Mar;33(3):290-5.
10. Robinson MD, McCarthy DJ, Smyth GK. edgeR: a Bioconductor package for differential expression analysis of digital gene expression data. Bioinformatics. 2010 Jan 1;26(1):139-40.
11. Ewels P, Magnusson M, Lundin S, Käller M. MultiQC: summarize analysis results for multiple tools and samples in a single report. Bioinformatics. 2016 Oct 1;32(19):3047-8.
12. Love MI, Huber W, Anders S. Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. Genome Biol. 2014;15(12):550.