

RNA-seq data analysis:

How to find differentially expressed genes?

Eija Korpelainen, Maria Lehtivaara
CSC – IT Center for Science, Finland
chipster@csc.fi



Understanding data analysis - why?

- **Bioinformaticians might not always be available when needed**
- **Biologists know their own experiments best**
 - Biology involved (e.g. genes, pathways, etc)
 - Potential batch effects etc
- **Allows you to design experiments better**
 - Enough replicates, reads etc → less money wasted
- **Allows you to discuss more easily with bioinformaticians**

What will I learn?

- **Introduction to RNA sequencing**
- **How to operate the Chipster software used in the exercises**
- **RNA-seq data analysis**
 - Central concepts
 - Analysis steps
 - File formats
- **Things to take into account when designing experiments**

Introduction to Chipster

Chipster

- **Provides an easy access to over 400 analysis tools**
 - Command line tools
 - R/Bioconductor packages
- **Free, open source software**
- **What can I do with Chipster?**
 - analyze and integrate high-throughput data
 - visualize data efficiently
 - share analysis sessions
 - save and share automatic workflows

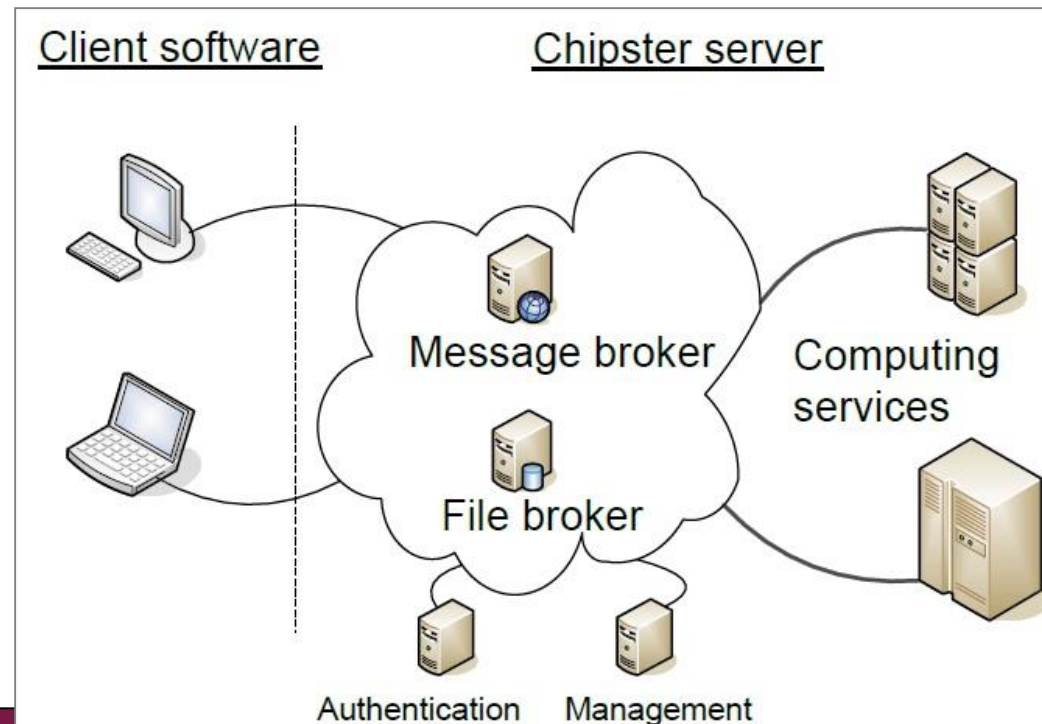
Technical aspects

➤ Client-server system

- Enough CPU and memory for large analysis jobs
- Centralized maintenance

➤ Easy to install

- Client uses Java Web Start
- Server available as a virtual machine





Chipster

Open source platform for data analysis



- Home
- Getting access
- Analysis tool content
- Screenshots
- Manual
- Tutorial videos
- Course material
- Cite
- FAQ
- Contact

- For developers:
 - Open source project
 - Tool editor

Welcome to Chipster

Chipster is a user-friendly software for analyzing high-throughput data such as NGS and microarrays. It contains over 400 analysis tools and a large collection of reference genomes. Users can save and share automatic analysis workflows, and visualize data interactively using for example the [built-in genome browser](#). Chipster's client software uses Java Web Start to install itself automatically, and it connects to computing servers for the actual analysis. Chipster is open source, and the server environment is available as a [virtual machine image](#) free of charge. If you would like to use Chipster running on CSC's server, you need a [user account](#).



Launch Chipster v3.14

...or launch with more memory: [3 GB](#) or [6 GB](#)

If you have trouble launching Chipster, read [this](#)

News and resources:

- 24.9.2018 [Video tutorials for single cell RNA-seq data analysis available!](#)
- 13.9.2018 [Version 3.14 released](#)
- 17.4.2018 [RNA-seq tutorial for differential expression analysis](#)
- 19.8.2014 [RNA-seq data analysis guidebook](#) with Chipster instructions
- [News archive](#)

Training:

- 14.-15.3.2019 [Single cell RNA-seq data analysis](#), CSC
- 11.12.2018 Community analysis of amplicon sequencing data, Evira
- 19.9.2018 [Single cell RNA-seq data analysis](#), CSC
- 4.-5.9.2018 RNA-seq data analysis, University of Oulu
- 8.8.2018 Community analysis of amplicon sequencing data, JyU
- 26.-28.6.2018 Expression data analysis, DKFZ
- 9.2.2018 [Single cell RNA-seq data analysis](#), CSC
- 16.1.2018 [Webinar: VirusDetect pipeline](#)

File Edit View Workflow Help

Datasets

two-sample.tsv

column-value-filter.tsv

hc.tre

kmeans.pdf

kmeans.tsv

extract.tsv

seqs.txt.wee

seqs.html

annotations.tsv

annotations.html

cpdb-pathways.html

cpdb-pathways.tsv

cpdb-genes.tsv

Analysis tools

Microarrays

NGS

Misc

Normalisation

Quality control

Preprocessing

Statistics

Clustering

Annotation

Pathways

Promoter analysis

Copy number aberrations

Visualisation

Utilities

One sample tests

Two groups tests

ROTS

SAM

Several groups tests

Linear modelling

Linear modelling using user-defined design matrix

Test proportions

Correlate with phenodata

Correlate miRNA with target expression

Time series

Association analysis

Tests for comparing the mean gene expression of two groups. LPE only works, if the whole normalized data is used, i.e., the data should not be filtered. Other than empiricalBayes might be slow, if run on unfiltered data.

Show parameters

Run

More help

Show tool sourcecode

Workflow

Fit

13

tsv

pdf

tsv

png

pdf

pdf

pdf

tsv

tsv

tsv

tsv

html

tsv

tsv

tre

pdf

tsv

tsv

html

wee

html

Visualisation

Maximise

Detach

Close

two-sample.tsv

472 kB, Wed Sep 03 06:56:07 EEST 2014

(Click here to add your notes)

Analysis history

Statistics / Two groups tests

Column

group

Pairing

EMPTY

Test

empirical Bayes

p-value adjustment method

BH

p-value threshold

0.01

Show NA

no

Spreadsheet

Heatmap

Expression profile

Volcano plot

Scatterplot

3D Scatterplot

Histogram

Open in external web browser

Connected to chipster.csc.fi

View jobs

0 jobs running

Used memory 118M / 870M

Mode of operation

Select: data → tool category → tool → run → visualize

The screenshot displays the Chipster 3.4.0 (build 1441) interface, which is divided into several panels:

- Datasets:** Lists input files such as `control_chr_1_sorted.bam`, `control_chr_1_sorted.bam.bai`, `treatment_chr_1_sorted.bam`, `treatment_chr_1_sorted.bam.bai`, `macs2-summits.bed`, `macs2-peaks.tsv`, `macs2_model.pdf`, `macs2-peaks.bed`, `macs2-log.txt`, and `macs2_narrowpeak.bed`.
- Analysis tools:** A list of tools categorized by Microarrays, NGS, and Misc. The **NGS** category is selected, showing tools like "Find peaks using MACS2", "Find peaks using MACS, treatment only", "Find peaks using MACS, treatment vs. control", "Find broad peaks using F-seq", "Find the nearest genes for regions", "Find unique and annotated genes", "GO enrichment for list of genes", "Find motifs with GADEM and match to JASPAR", "Dimont sequence extractor", and "Dimont sequence extractor using own genome". A red arrow points to the "Find peaks using MACS2" tool.
- Workflow:** A diagram showing the data flow. It starts with `bam` and `bai` files, which are converted to `bed` and `tsv` formats. A red circle highlights a `bed` file in the workflow. A red arrow points from this `bed` file to the "Find peaks using MACS2" tool in the Analysis tools panel.
- Visualisation:** A genome browser view showing the results of the MACS2 analysis. It displays a genomic track for chromosome 1, with a peak at position 144M 322k. The peak is labeled `RNF115-001`. Below the track, there are two tracks for `control_chr_1_sorted.bam` and `treatment_chr_1_sorted.bam`, showing read coverage. A red arrow points from the `bed` file in the workflow to the peak in the visualization.
- Settings / Selected / Legend:** A panel on the right showing settings for the selected tool. It includes options for **Genome** (Homo sapiens NCBB36.54 (hg18)), **Location** (Chromosome 1, Location (gene or position) 144322773, View size 4 kb), and **Options** (Reads, Highlight SNPs, Density graph, Low complexity regions, Mark multimapping reads, Coverage type strand-specific, Coverage scale 50). A red arrow points from the "Show parameters" button in the Analysis tools panel to this settings panel.
- View jobs:** A button at the bottom right, circled in red, which is used to view the status of running jobs.

At the bottom of the interface, it shows "Connected to chipster.csc.fi", "0 jobs running", and "Used memory 199M / 870M".

When running analysis tools, pay attention to parameters!

- **make sure the input files are correctly assigned** if there are multiple files (see below)
- **choose the right reference genome**
- **check especially the bolded parameters**

Analysis tools - Single cell RNA-seq - Merge aligned and unaligned BAM

Reference genome: Homo_sapiens.GR... ☒ Hide parameters

Input datasets

Unaligned BAM: drseq_read_1_unali...

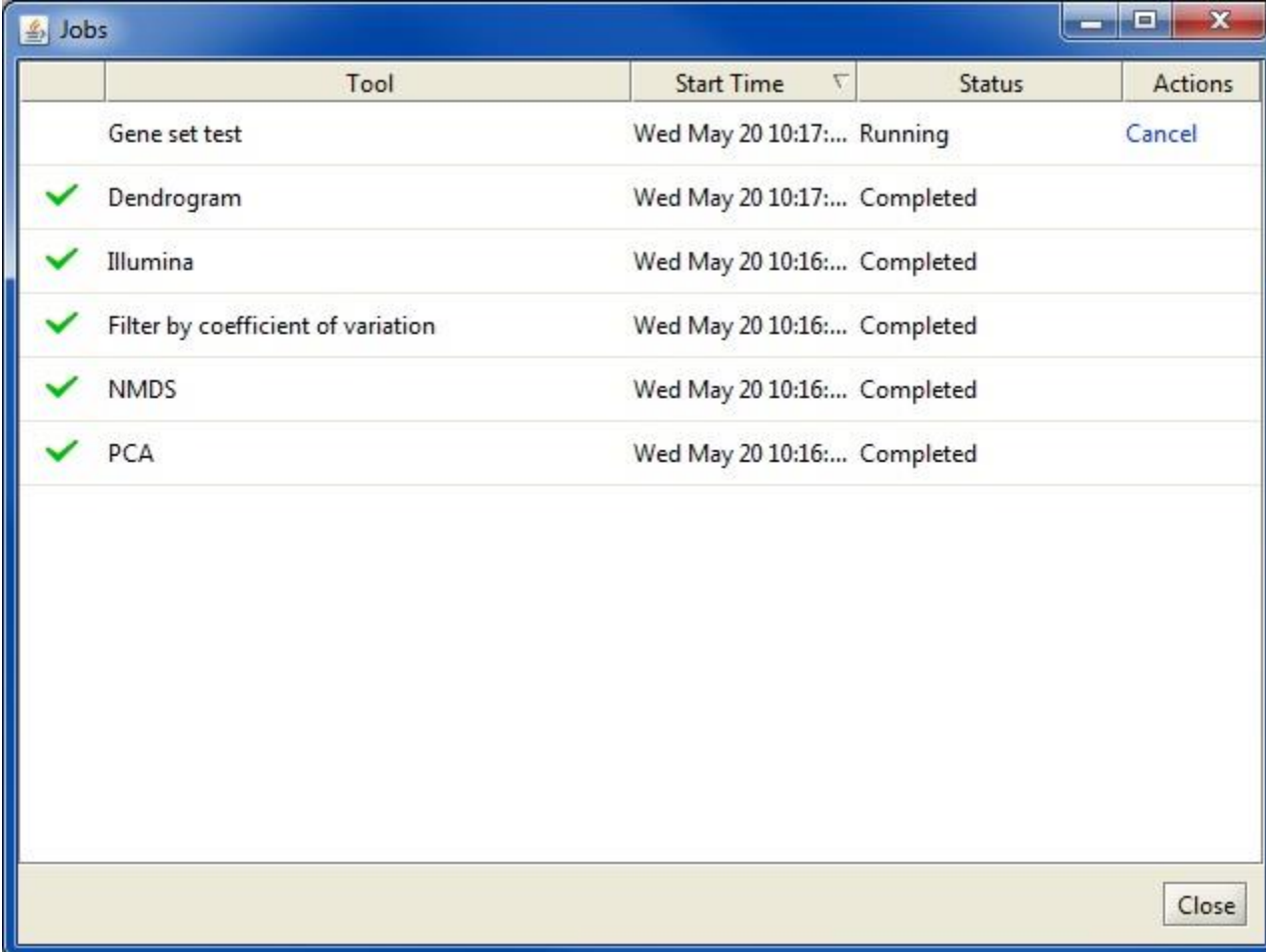
Aligned BAM: drseq_read_1.bam

drseq_read_1_unaligned.bam
drseq_read_1.bam

Merge sorted BAM alignment and unaligned, tagged BAM file. Make sure the input files are assigned correctly!

Job manager

- You can run many analysis jobs at the same time
- Use Job manager to
 - view status
 - cancel jobs
 - view time
 - view parameters



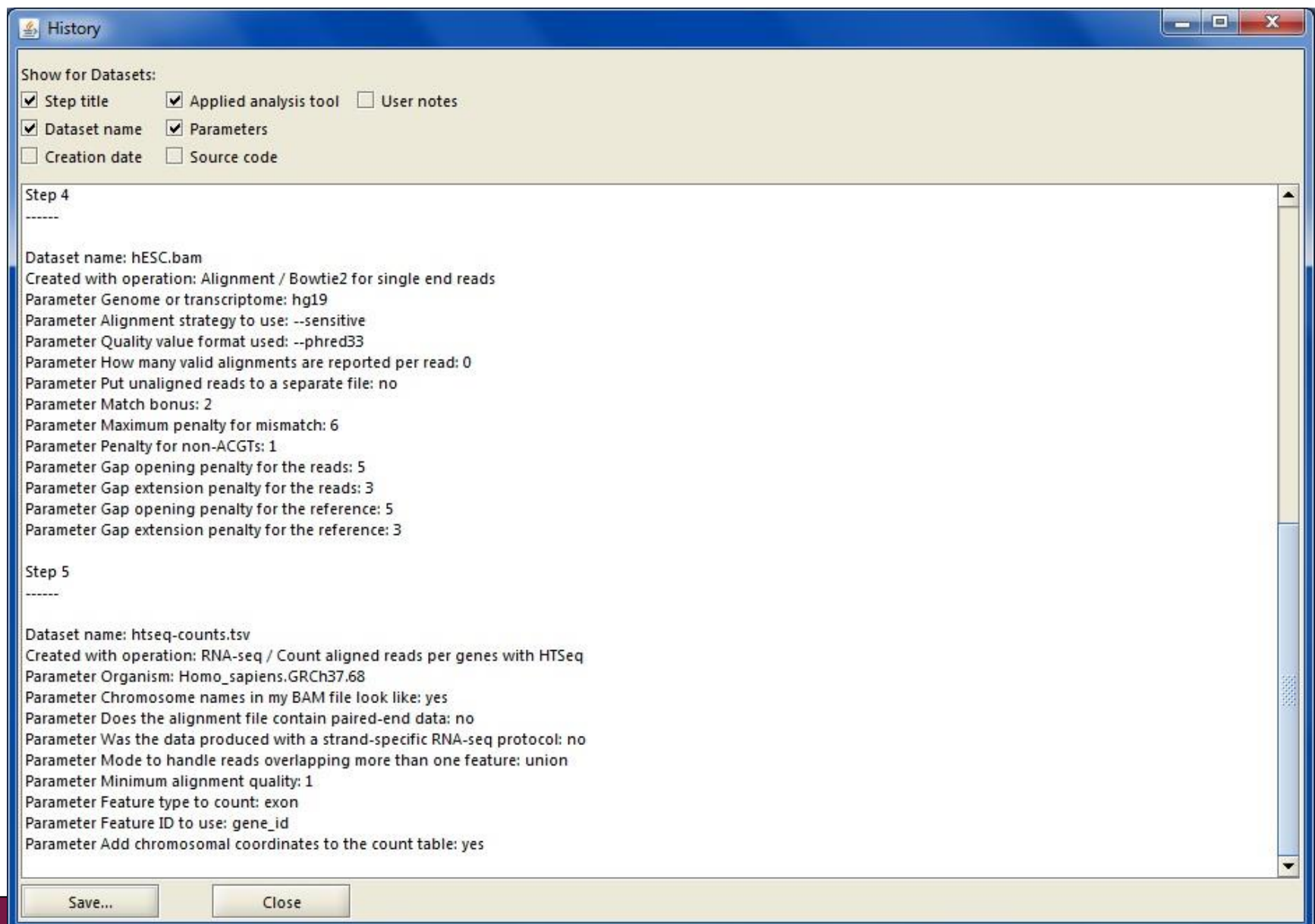
The screenshot shows a window titled 'Jobs' with a table of analysis jobs. The table has columns for Tool, Start Time, Status, and Actions. The first job, 'Gene set test', is 'Running' and has a 'Cancel' button. The other five jobs are 'Completed' and each has a green checkmark icon in the Actions column.

	Tool	Start Time	Status	Actions
	Gene set test	Wed May 20 10:17:...	Running	Cancel
✓	Dendrogram	Wed May 20 10:17:...	Completed	
✓	Illumina	Wed May 20 10:16:...	Completed	
✓	Filter by coefficient of variation	Wed May 20 10:16:...	Completed	
✓	NMDS	Wed May 20 10:16:...	Completed	
✓	PCA	Wed May 20 10:16:...	Completed	

Close

Analysis history is saved automatically

-you can add tool source code to reports if needed

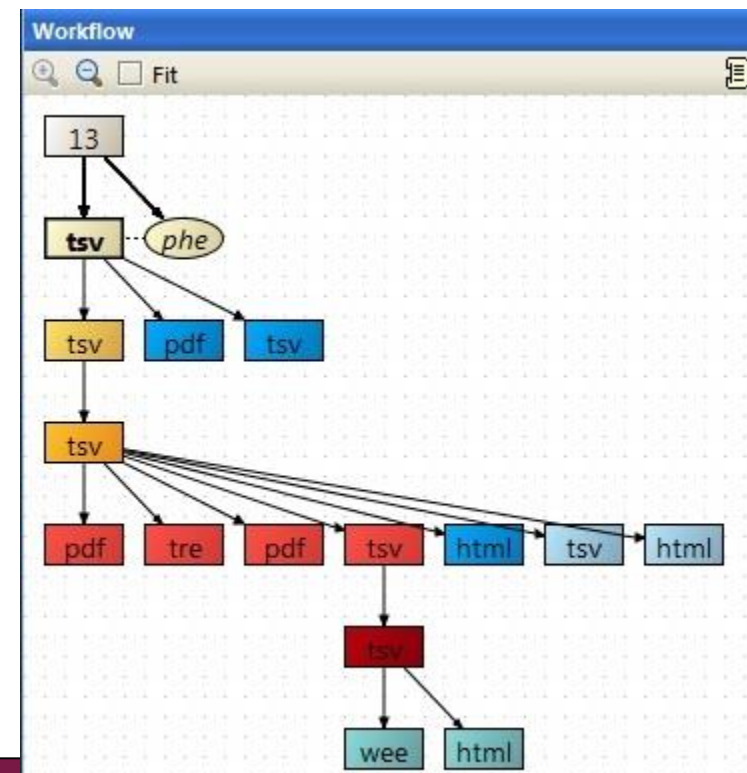


Analysis sessions

- **Remember to save the analysis session within 3 days**
 - Session includes all the files, their relationships and metadata (what tool and parameters were used to produce each file)
 - Session is a single .zip file
 - Note that you can save two sessions of the same data
 - one with raw data (FASTQ files) and one smaller, working version where the FASTQ files are deleted after alignment
- **You can save a session locally (= on your computer)**
- **and in the cloud**
 - but note that the cloud sessions are not stored forever!
 - **If your analysis job takes a long time, you don't need to keep Chipster open:**
 - Wait that the data transfer to the server has completed (job status = running)
 - Save the session in the cloud and close Chipster
 - Open Chipster within 3 days and save the session containing the results

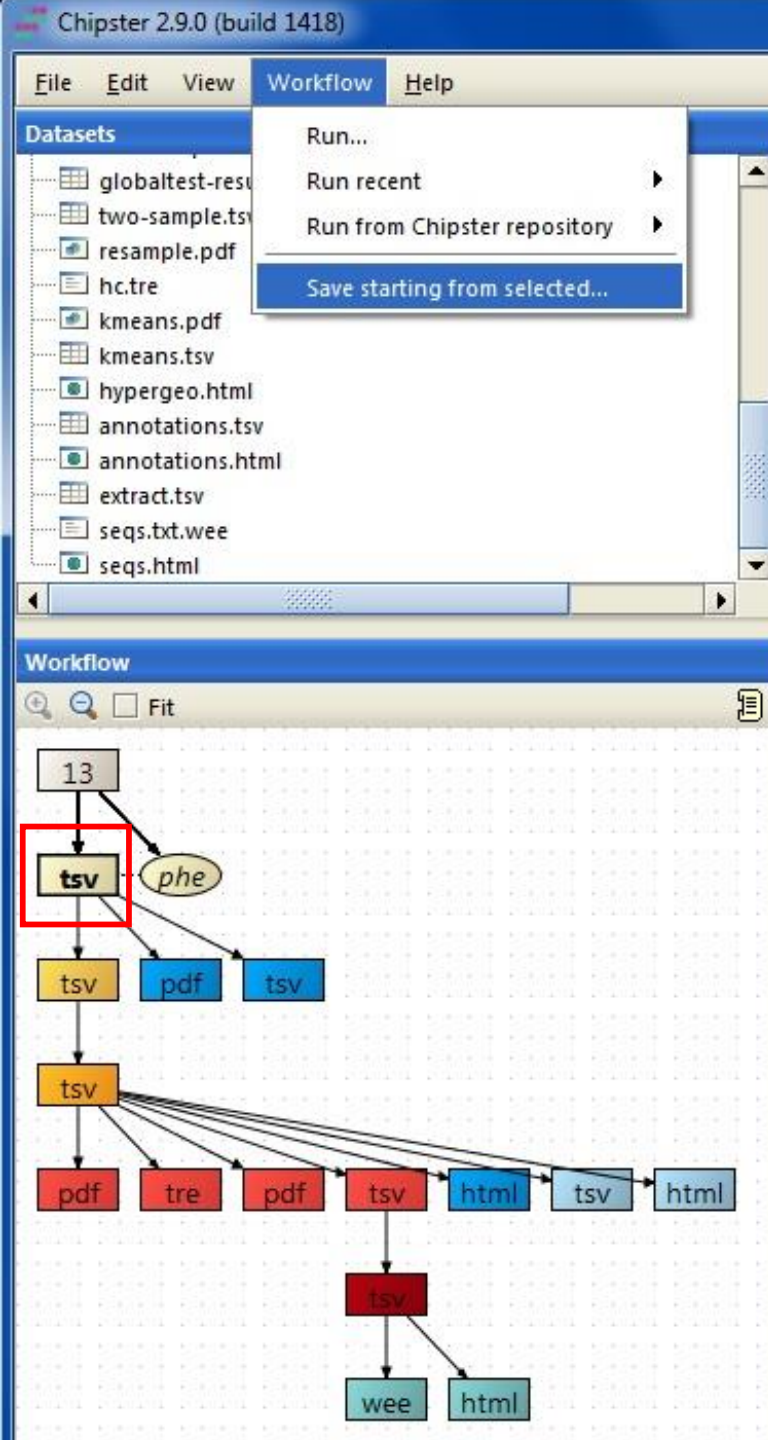
Workflow panel

- Shows the relationships of the files
- You can move the boxes around, and zoom in and out.
- Several files can be selected by keeping the Ctrl key down
- Right clicking on the data file allows you to
 - Save an individual result file ("Export")
 - Delete
 - Link to another data file
 - Save workflow



Workflow – reusing and sharing your analysis pipeline

- **You can save your analysis steps as a reusable automatic “macro”, which you can apply to another dataset**
- **When you save a workflow, all the analysis steps and their parameters are saved as a script file, which you can share with other users**



Saving and using workflows

- Select the starting point for your workflow
- Select "Workflow/ Save starting from selected"
- Save the workflow file on your computer with a meaningful name
 - Don't change the ending (.bsh)
- To run a workflow, select
 - Workflow->Open and run
 - Workflow->Run recent (if you saved the workflow recently).

Analysis tools for different types of data

➤ 200 NGS tools for

- RNA-seq
- single cell RNA-seq
- small RNA-seq
- microbiome analysis (16S)
- exome/genome-seq
- ChIP-seq
- FAIRE/DNase-seq

➤ 140 microarray tools for

- gene expression
- miRNA expression
- protein expression
- aCGH
- SNP
- integration of different data

➤ 60 tools for sequence analysis

- BLAST, EMBOSS, MAFFT
- Phylip

Visualizing the data

➤ Data visualization panel

- Maximize and redraw for better viewing
- Detach = open in a separate window, allows you to view several images at the same time

➤ Two types of visualizations

1. Interactive visualizations produced by the client program

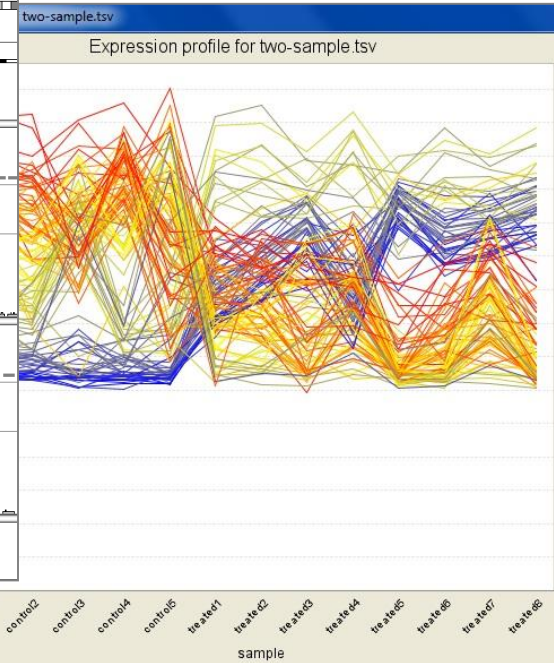
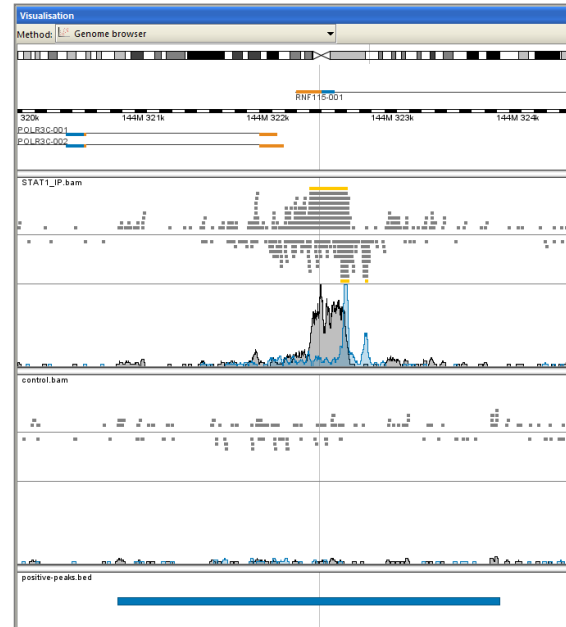
- Select the visualization method from the pulldown menu
- Save by right clicking on the image

2. Static images produced by analysis tools

- Select from Analysis tools/ Visualisation
- View by double clicking on the image file
- Save by right clicking on the file name and choosing "Export"

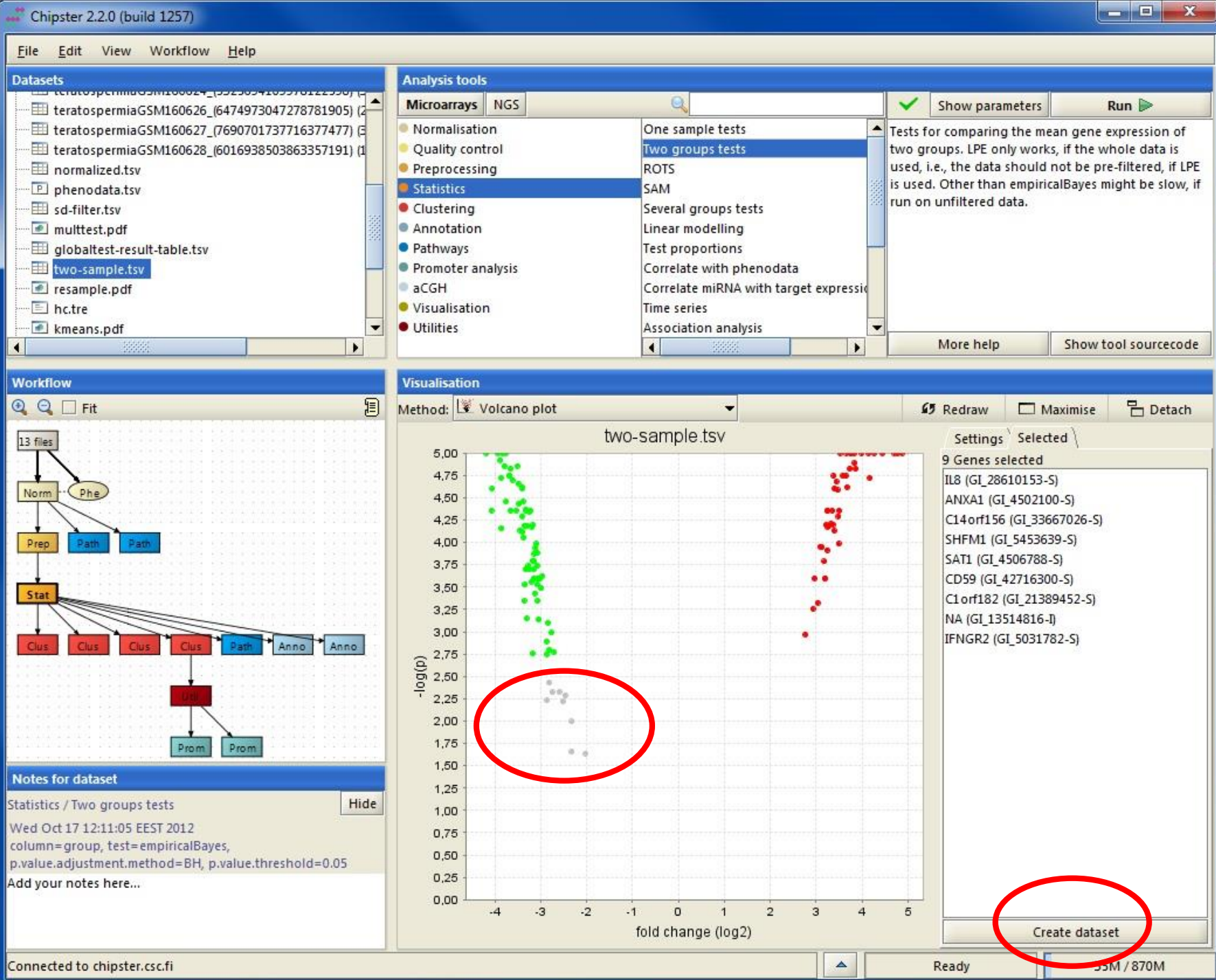
Interactive visualizations by the client

- **Genome browser**
- **Spreadsheet**
- **Histogram**
- **Venn diagram**
- **Scatterplot**
- **3D scatterplot**
- **Volcano plot**
- **Expression profiles**
- **Clustered profiles**
- **Hierarchical clustering**
- **SOM clustering**



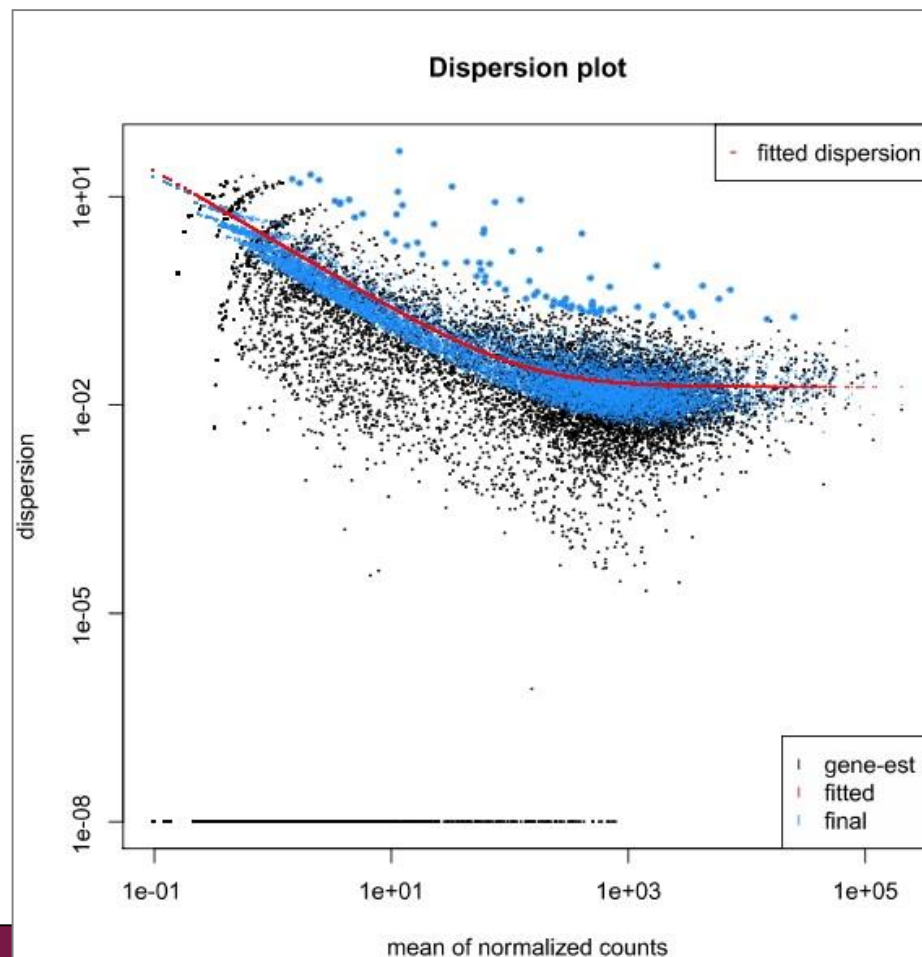
Available actions:

- **Select genes and create a gene list**
- **Change titles, colors etc**
- **Zoom in/out**



Static images produced by R/Bioconductor

- Dispersion plot
- Heatmap
- tSNE plot
- Violin plot
- PCA plot
- MA plot
- MDS plot
- Box plot
- Histogram
- Dendrogram
- K-means clustering
- etc...



Options for importing data to Chipster

- **Import files/ Import folder**
- **Import from URL**
 - Utilities / Download file from URL directly to server
- **Open an analysis session**
 - Files / Open session
- **Possibility to import from BaseSpace coming soon**
- **Import from SRA database**
 - Utilities / Retrieve FASTQ or BAM files from SRA
- **Import from Ensembl database**
 - Utilities / Retrieve data for a given organism in Ensembl
- **What kind of data files can I use in Chipster?**
 - Compressed files (.gz) are ok
 - FASTQ, BAM, read count files (.tsv), GTF

How to import a tar package containing many files and use only some of them?

- **Import the tar package**
 - File / Import from / URL directly to server
- **Check what files it contains**
 - Utilities / List contents of a tar file
- **Selectively extract the files you want**
 - Utilities / Extract .tar or .tar.gz file

Problems? Send us a support request

-request includes the error message and link to analysis session (optional)

```
Hi,  
I'm trying to normalise my Illumina microarray data (obtained with the Illumina HT-12 v4.0)  
For that purpose I have selected the Normalisation option "Illumina - lumi pipeline"  
However, the normalisation did not complete successfully.
```

```
Any advice to solve this problem ?
```

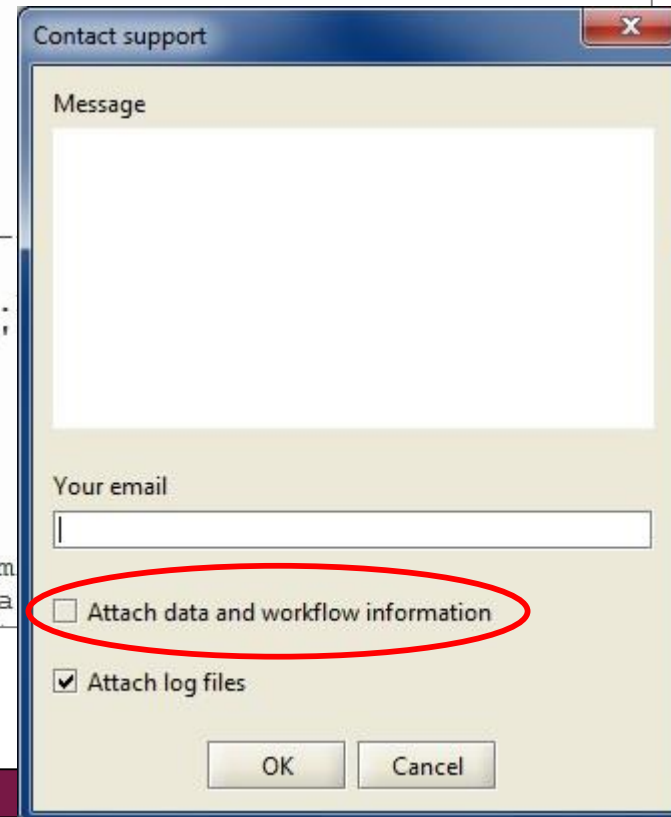
```
Thank you in advance for your precious help.
```

```
Best regards
```

```
Error message:
```

```
in library(chiptype, character.only = T) :  
  there is no package called 'Illumina.db'
```

```
-----  
> chipster.common.path = '/opt/chipster/comp/modules/common/R-2.  
> chipster.module.path = '/opt/chipster/comp/modules/microarray'  
> setwd("271661a6-946c-450f-bb21-5d5b5a2837aa")  
> probe.identifier <- "Probe_ID"  
> transformation <- "log2"  
> background.correction <- "none"  
> normalize.chips <- "quantile"  
> chiptype <- "empty"  
> # TOOL norm-illumina-lumi.R: "Illumina - lumi pipeline" (Illum  
BeadSummaryData files, and using lumi methodology. If you have a
```



Contact support

Message

Your email

☐ Attach data and workflow information

☒ Attach log files

OK Cancel

How to get a user account?

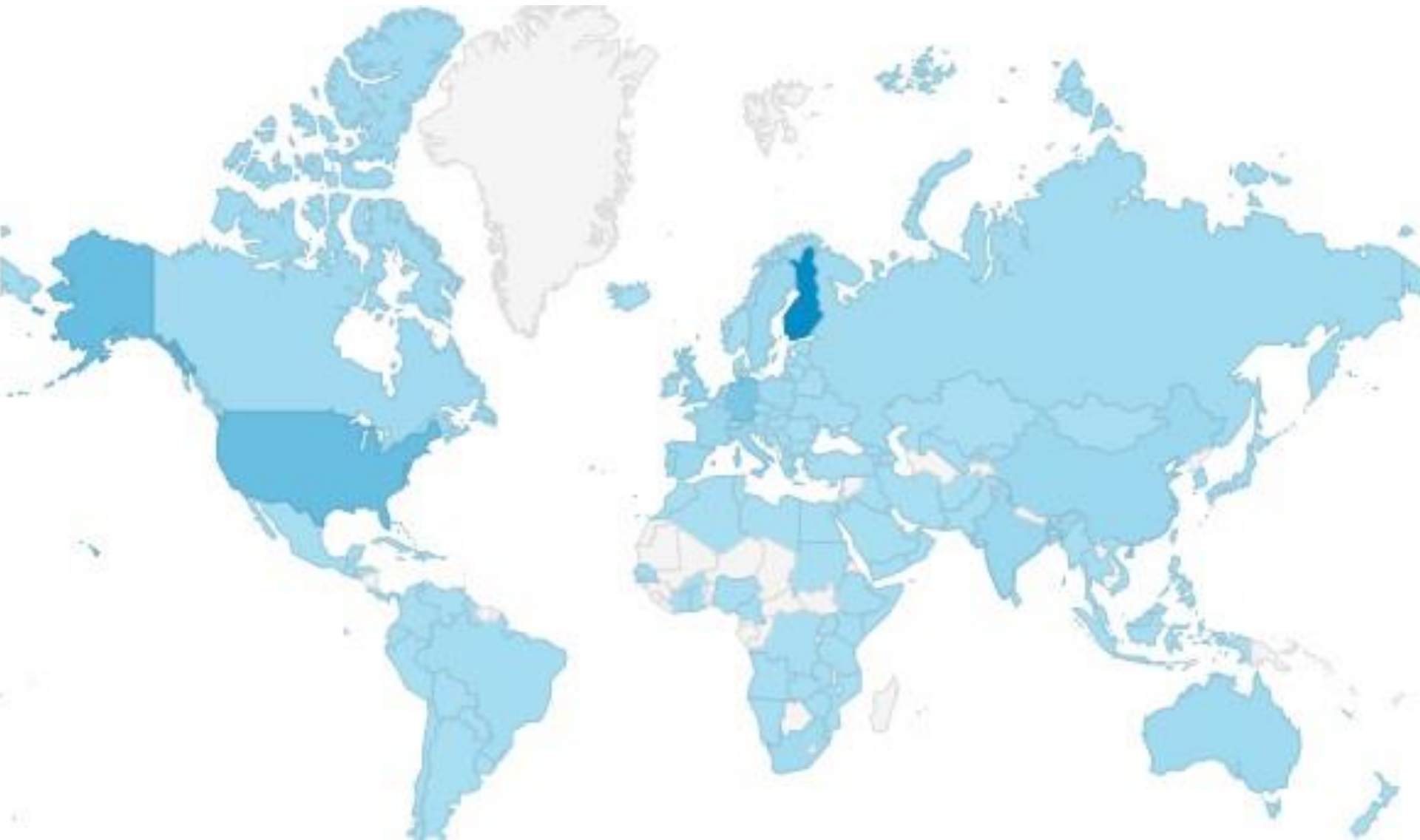
➤ **Getting a CSC user account is free of charge**

- use Scientist's User Interface service (<https://sui.csc.fi>) with HAKA credentials
 - click on the purple HAKA link
 - log in with your HAKA username and password
 - fill in the sign up form as shown in <https://research.csc.fi/csc-guide-getting-access-to-csc-services#1.2.1>
- If you don't have HAKA credentials, email servicedesk@csc.fi
- Users who already have a regular CSC username and password can use those for Chipster

➤ **See <https://chipster.csc.fi/access.shtml> for details**



Acknowledgements to Chipster users and contributors



More info

- chipster@csc.fi
- <http://chipster.csc.fi>
- Chipster tutorials in YouTube

GitHub

This repository Search

Explore Features

chipster / chipster

Chipster is a user-friendly analysis software for high-throughput data.

7,565 commits

18 branches

123 releases

14 contributors

BMC
Genomics

IMPACT
FACTOR
4.21

[home](#) | [journals A-Z](#) | [subject areas](#) | [advanced search](#) | [authors](#) | [reviewers](#) | [libraries](#) | [about](#) | [my BioMed Central](#)

Software

Highly accessed Open Access

Chipster: user-friendly analysis software for microarray and other high-throughput data

M Aleks Kallio ✉, Jarno T Tuimala ✉, Taavi Hupponen ✉, Petri Klemela ✉, Massimiliano Gentile ✉, Ilari Scheinin ✉, Mikko Koski ✉, Janne Kaki ✉ and Eija I Korpelainen ✉

BMC Genomics 2011, 12:507 doi:10.1186/1471-2164-12-507

RNA-seq Data Analysis

Korpelainen, Tuimala,
Somervuo, Huss, and Wong



Chapman & Hall/CRC
Mathematical and Computational Biology Series

RNA-seq Data Analysis

A Practical Approach



Eija Korpelainen, Jarno Tuimala,
Panu Somervuo, Mikael Huss, and Garry Wong

CRC Press
Taylor & Francis Group
A CHAPMAN & HALL BOOK

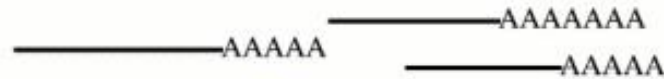
Introduction to RNA-seq

What can I investigate with RNA-seq?

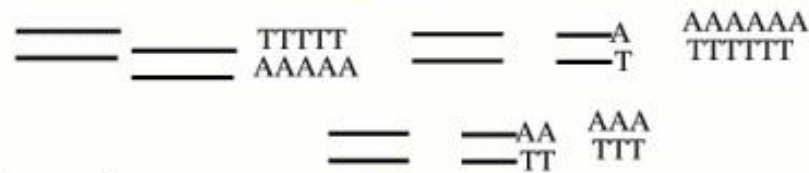
- **Differential expression**
- **Isoform switching**
- **New genes and transcripts**
- **New transcriptomes**
- **Variants**
- **Allele-specific expression**
- **Etc etc**

How was your data produced?

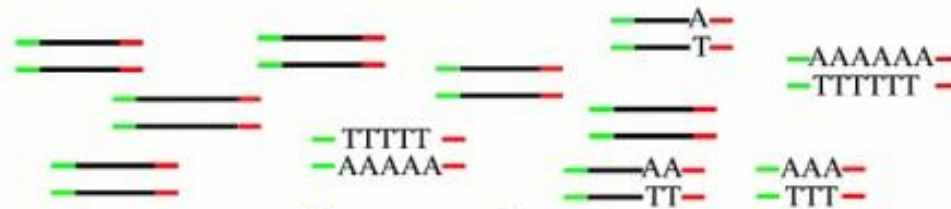
extraction of poly-A RNAs



conversion into ds-cDNA
and shearing

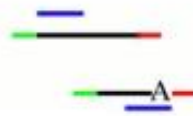


amplification and
adapter ligation

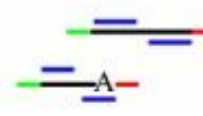


sequencing

single end (SET)



paired-end (PET)



PolyA purification

cDNA generation
& fragmentation

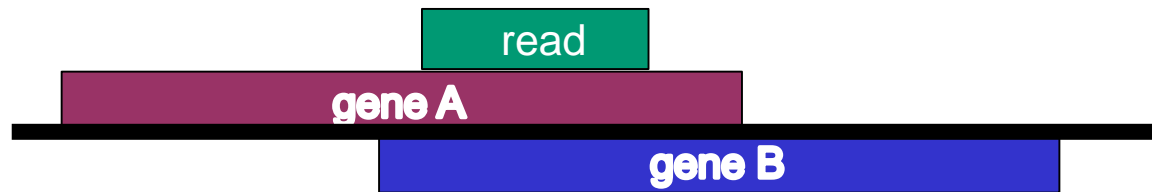
Library construction

Size selection

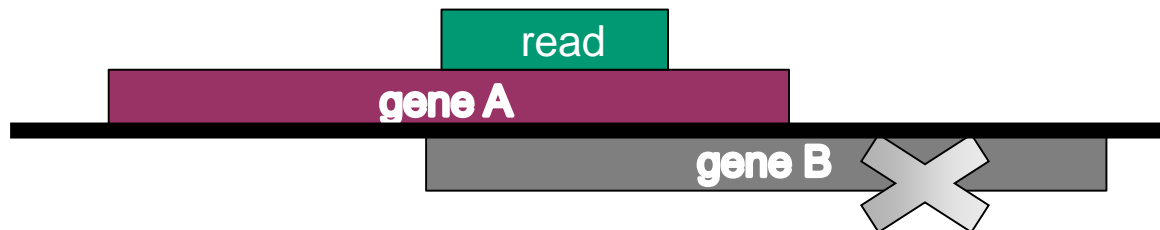


Stranded RNA-seq data

- **Tells if a read maps to the same strand where the parental gene is, or to the opposite strand**
 - Useful information when a read maps to a genomic location where there is a gene on both strands
- **Several lab methods, you need to know which one was used**
 - TruSeq stranded, NEB Ultra Directional, Agilent SureSelect Strand-Specific...



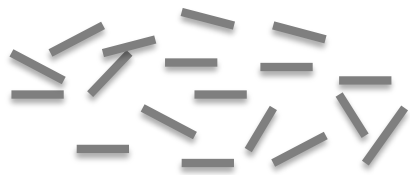
Unstranded data:
Does the read come
from geneA or
geneB?



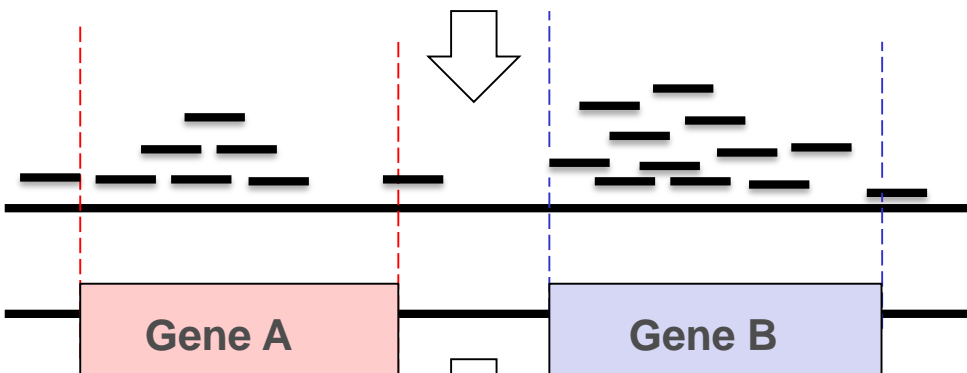
Stranded data
→ the read comes
from geneA



RNA-seq data analysis: typical steps



Raw data (reads)



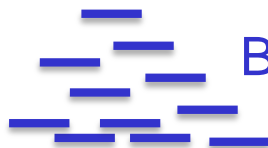
Align reads to
reference genome

Match alignment positions
with known gene positions

A = 6



B = 11



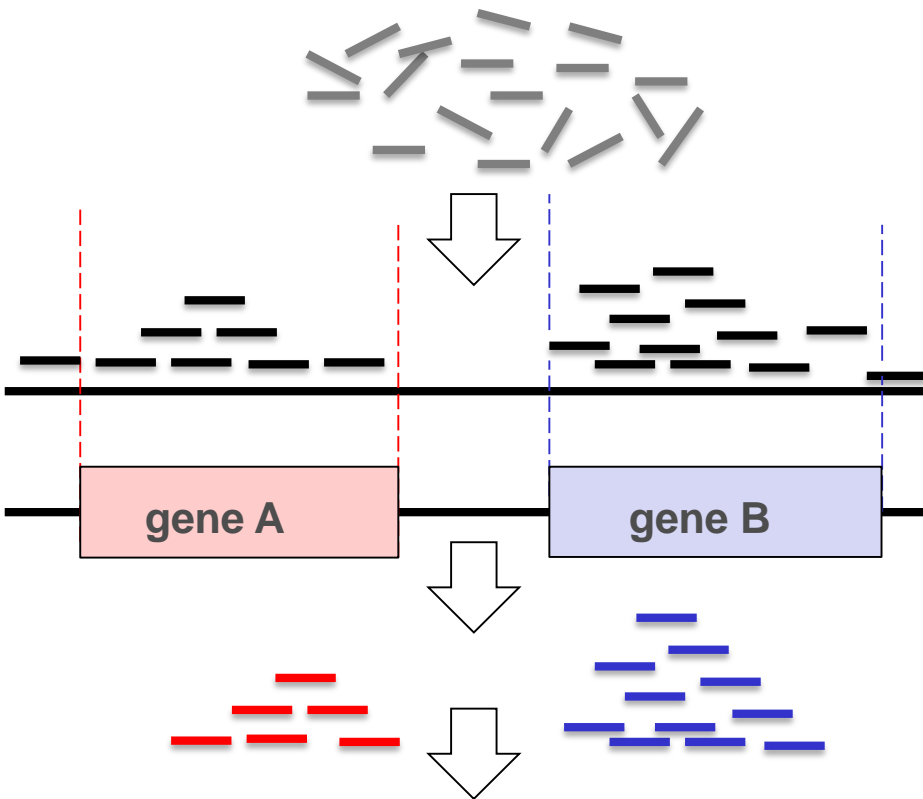
Count how many reads
each gene has

	Control 1	Control 2	Control 3	Sample 1	Sample 2	Sample 3
Gene A	6	5	7	170	100	110
Gene B	11	11	10	3	4	2
Gene C	200	150	355	50	1	3
Gene D	0	1	0	2	0	1

Compare sample groups:
differential expression
analysis



RNA-seq data analysis: steps, tools and files

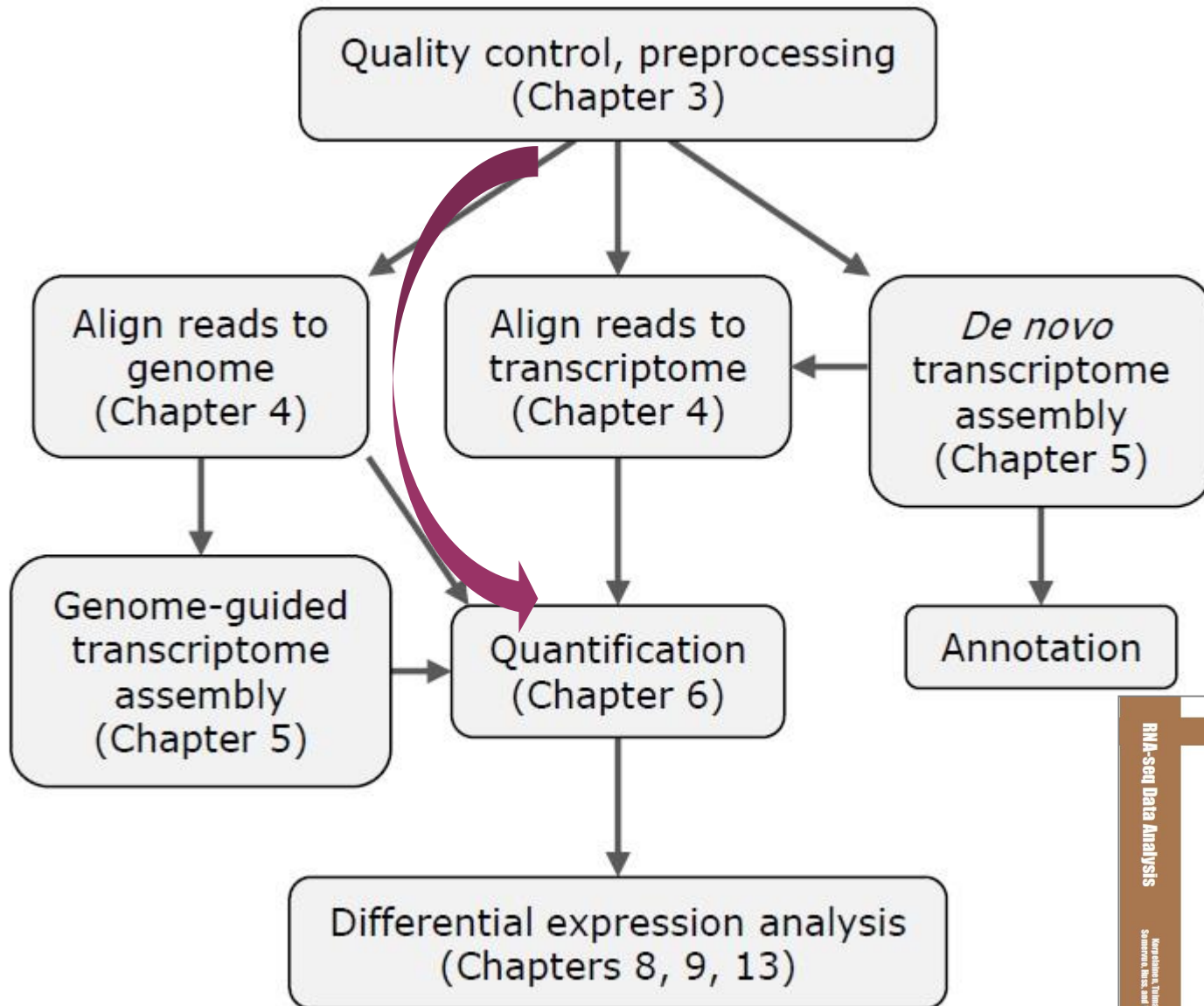


	Control 1	Control 2	Control 3	Sample 1	Sample 2	Sample 3
Gene A	6	5	7	170	100	110
Gene B	11	11	10	3	4	2
Gene C	200	150	355	50	1	3
Gene D	0	1	0	2	0	1

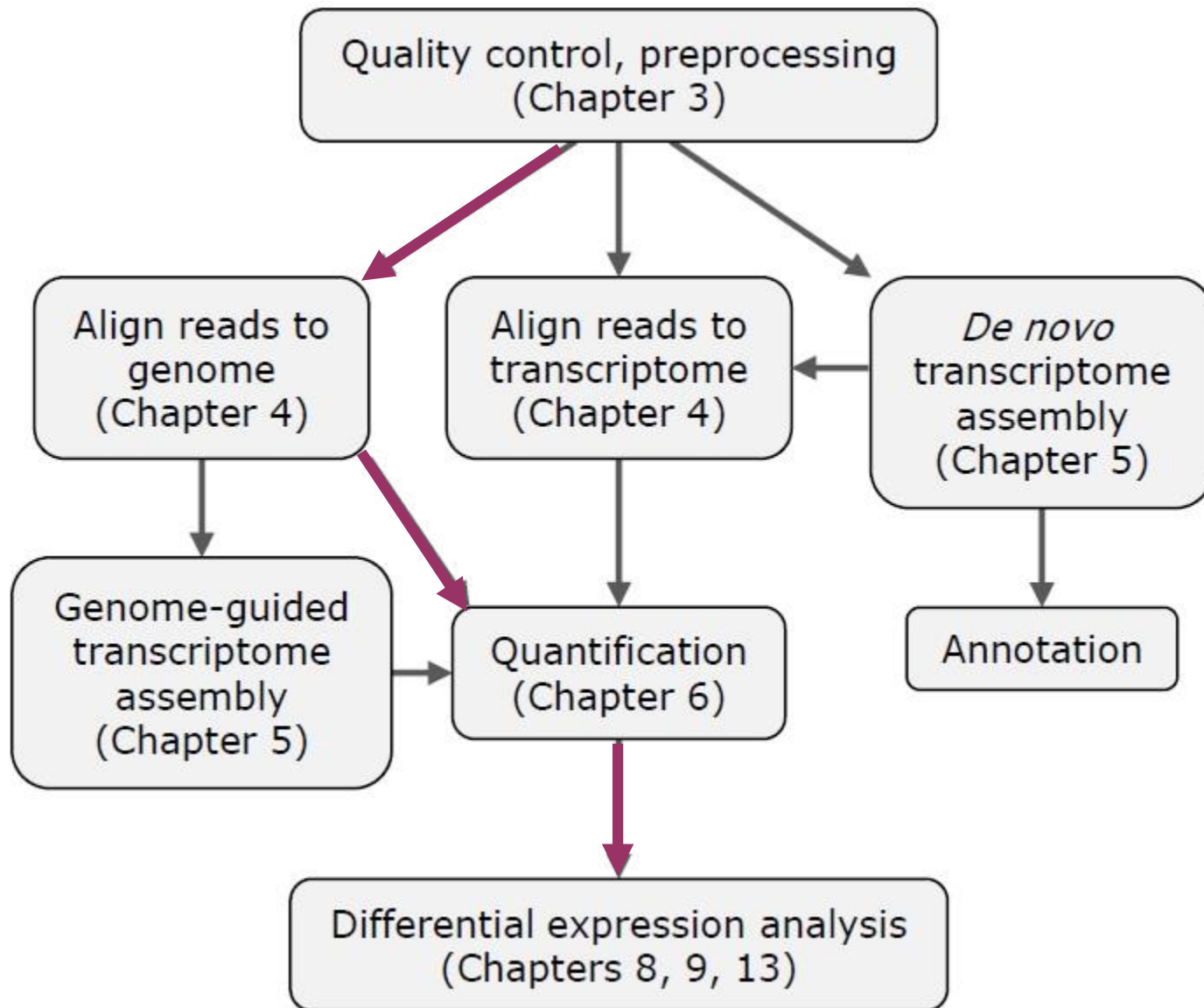
STEP	TOOL	FILE
Quality control	FastQC	FASTQ
Pre-processing	Trimmo-matic	FASTQ
Alignment	HISAT2	BAM
Quality control	RSeQC	
Quantitation	HTSeq	Read count file (TSV)
Combine count files to table	Define NGS experiment	Read count table (TSV)
Quality control	PCA, clustering	
Differential expression analysis	DESeq2, edgeR	Gene lists (TSV)

CSC

RNA-seq data analysis workflow



The steps we practise during the course



RNA-seq data analysis workflow

- **Quality control of raw reads**
- **Preprocessing if needed**
- **Alignment (=mapping) to reference genome**
- **Alignment level quality control**
- **Quantitation**
- **Experiment level quality control**
- **Differential expression analysis**
- **Visualization of reads and results in genomic context**

RNA-seq data analysis workflow

- **Quality control of raw reads**
- Preprocessing if needed
- Alignment (=mapping) to reference genome
- Alignment level quality control
- Quantitation
- Experiment level quality control
- Differential expression analysis
- Visualization of reads and results in genomic context

What and why?

➤ **Potential problems**

- low confidence bases, Ns
- sequence specific bias, GC bias
- adapters
- sequence contamination
- ...

Knowing about potential problems in your data allows you to

- **correct for them before you spend a lot of time on analysis**
- **take them into account when interpreting results**

Software packages for quality control

- **FastQC**
- **MultiQC**
- **PRINSEQ**
- **FastX**
- **TagCleaner**
- **...**

Raw reads: FASTQ file format

➤ Four lines per read:

@read name

GATTGGGGTTCAAAGCAGTATCGATCAAATAGTAAATCCATTTGTTCAACTCACAGTTT

+ read name

!'"((((**+))%%%++)(%%%%).1***-+*"))**55CCF>>>>>CCCCCCC65

➤ http://en.wikipedia.org/wiki/FASTQ_format

➤ Attention: Do not unzip FASTQ files

- Chipster's analysis tools can cope with zipped files (.gz)



Base qualities

- **If the quality of a base is 20, the probability that it is wrong is 0.01.**
 - Phred quality score $Q = -10 * \log_{10}(\text{probability that the base is wrong})$

T	C	A	G	T	A	C	T	C	G
40	40	40	40	40	40	40	40	37	35
- **"Sanger" encoding: numbers are shown as ASCII characters so that 33 is added to the Phred score**
 - E.g. 39 is encoded as "H", the 72nd ASCII character (39+33 = 72)
 - Note that older Illumina data uses different encoding
 - Illumina1.3: add 64 to Phred
 - Illumina 1.5-1.7: add 64 to Phred, ASCII 66 "B" means that the whole read segment has low quality

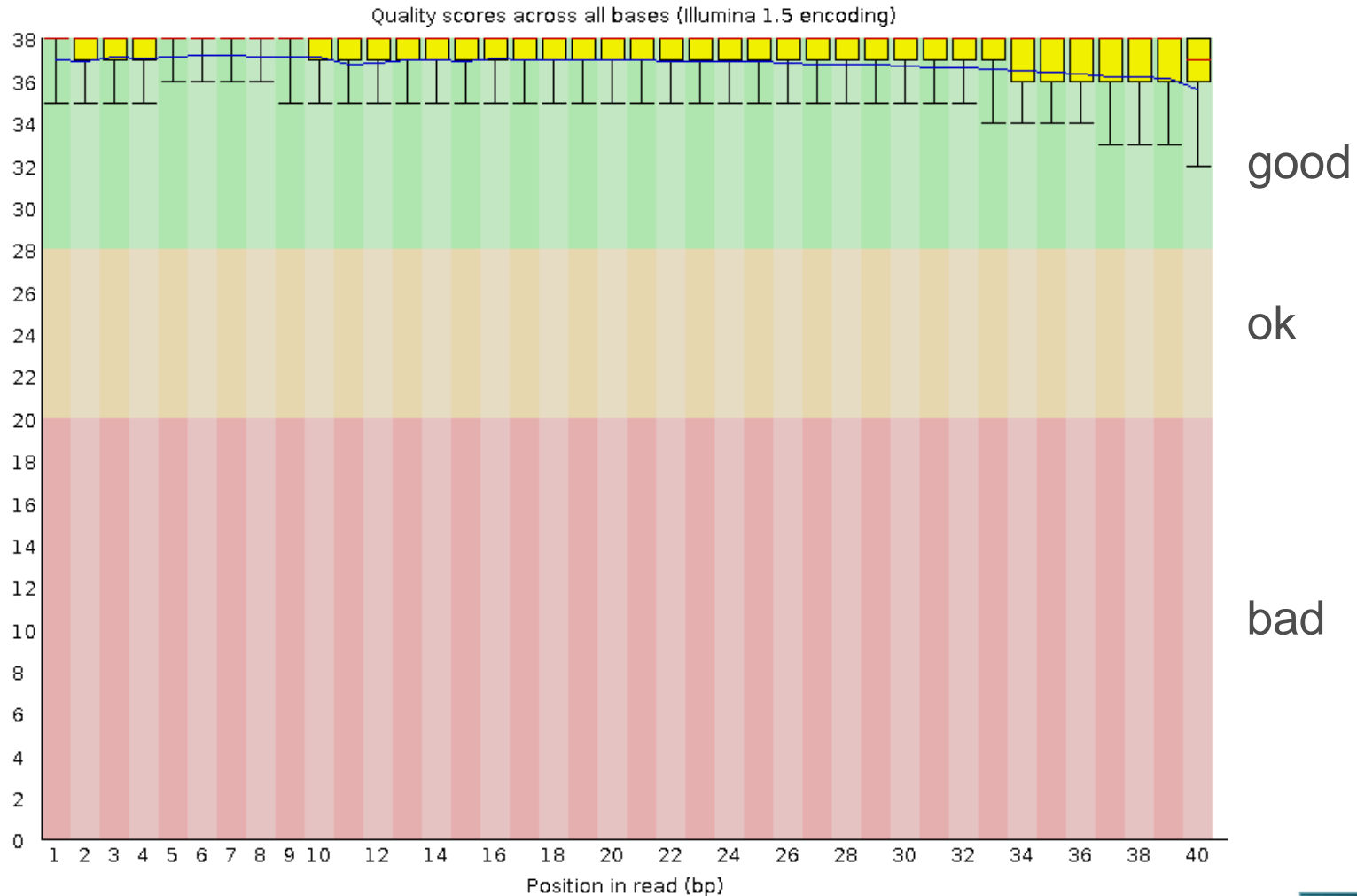


Base quality encoding systems

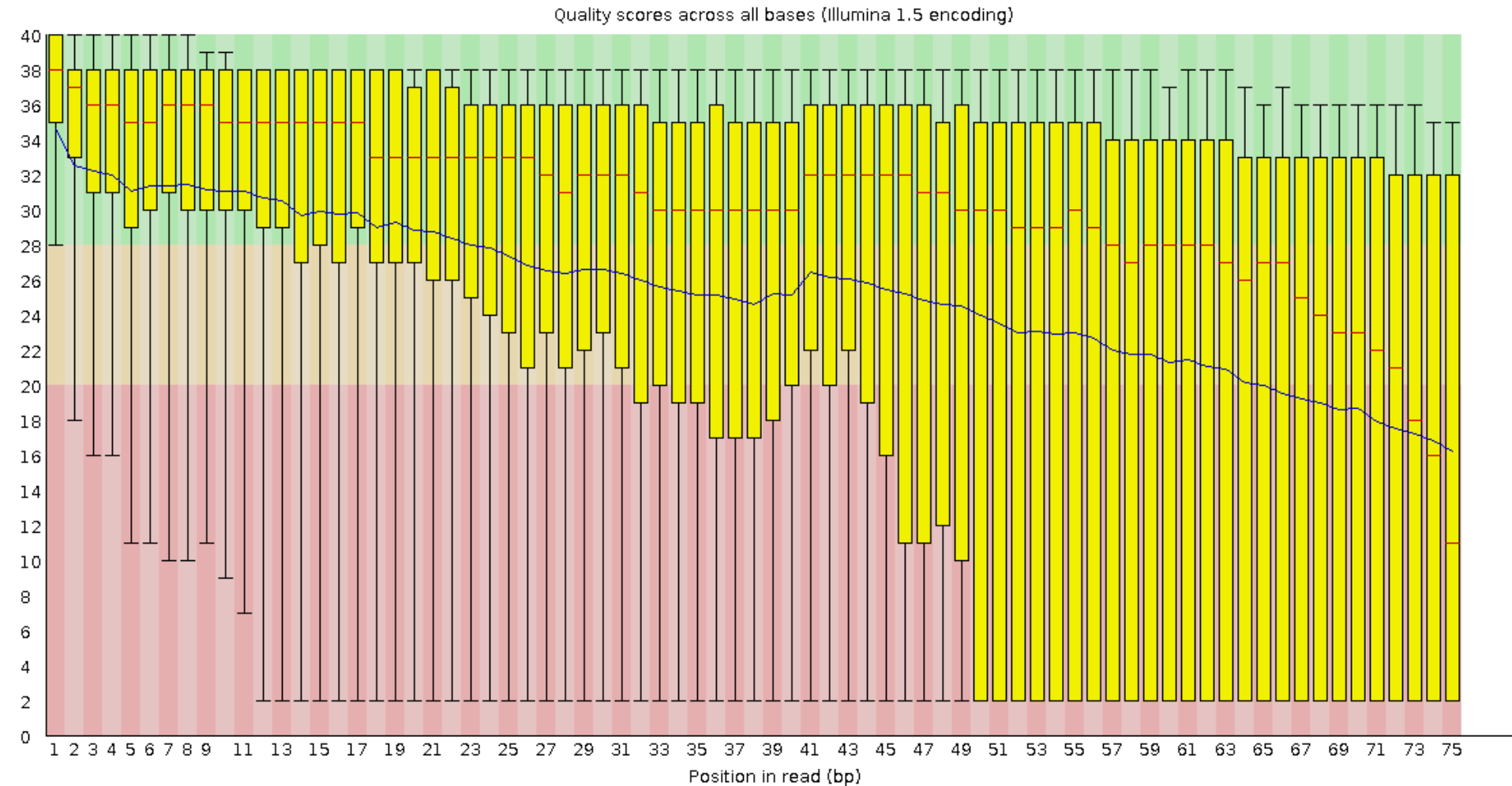
```
SSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSS.....
...XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX..
...IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII..
...JJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJ..
...LLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLL..
!"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefghijklmnopqrstuvwxyz
|                                     |   |   |                                     |
33                               59    64       73                                104
0.....26....31.....40
          -5....0.....9.....40
              0.....9.....40
                  3.....9.....40
0.2.....26....31.....41
```

```
S - Sanger          Phred+33,  raw reads typically (0, 40)
X - Solexa          Solexa+64,  raw reads typically (-5, 40)
I - Illumina 1.3+   Phred+64,  raw reads typically (0, 40)
J - Illumina 1.5+   Phred+64,  raw reads typically (3, 40)
      with 0=unused, 1=unused, 2=Read Segment Quality Control Indicator (bold)
      (Note: See discussion above).
L - Illumina 1.8+   Phred+33,  raw reads typically (0, 41)
```

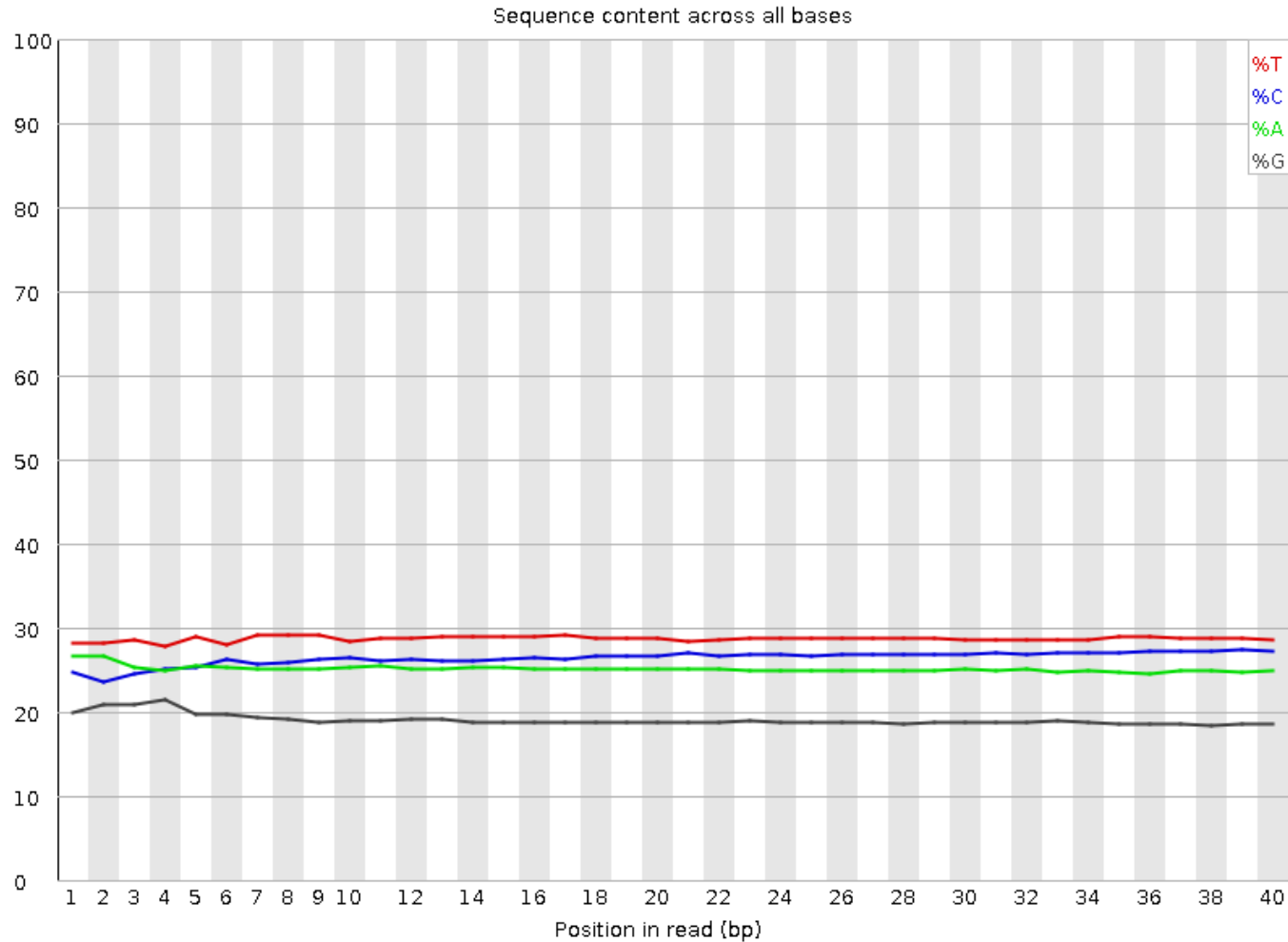
Per position base quality (FastQC)



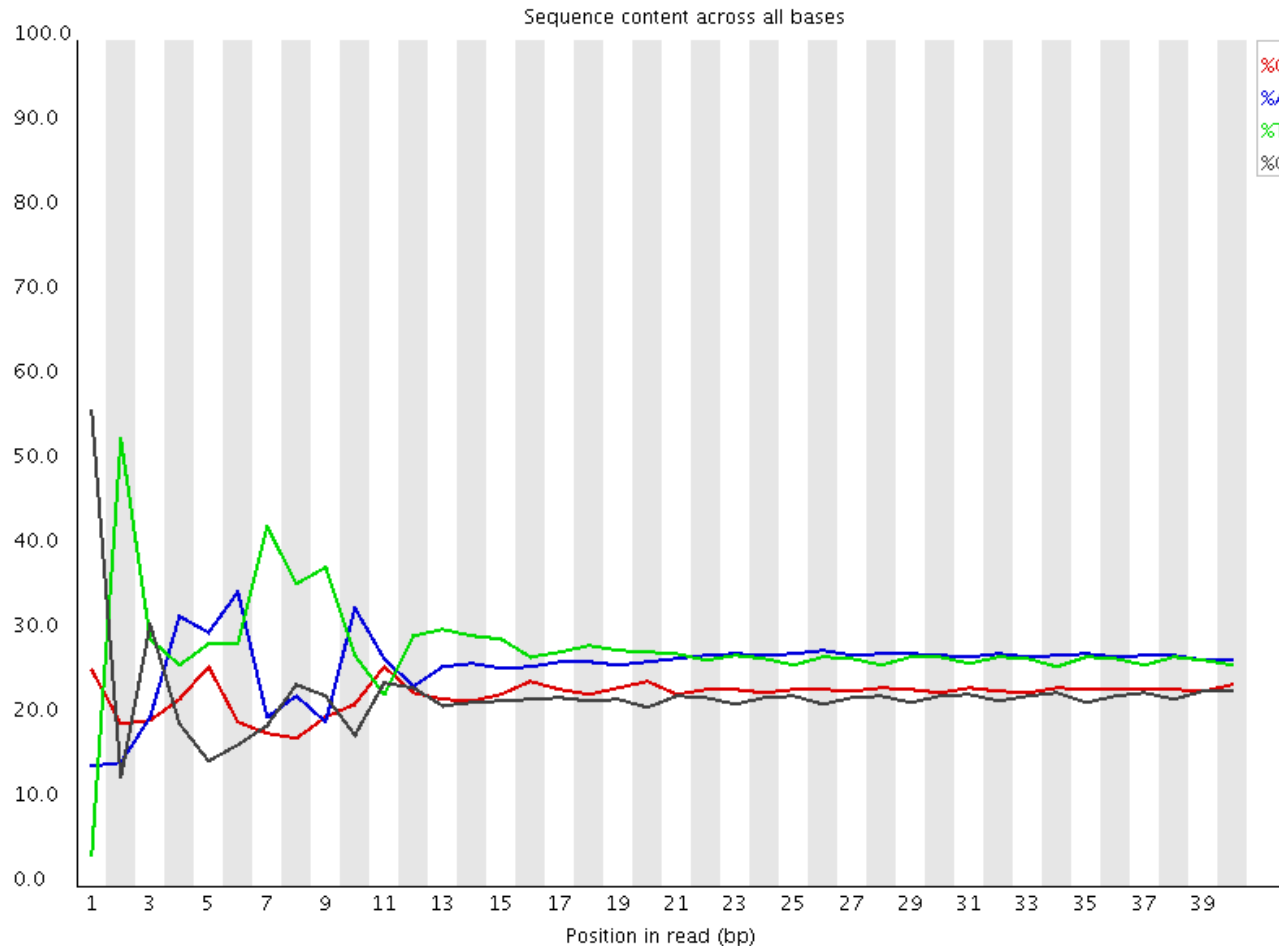
Per position base quality (FastQC)



Per position sequence content (FastQC)



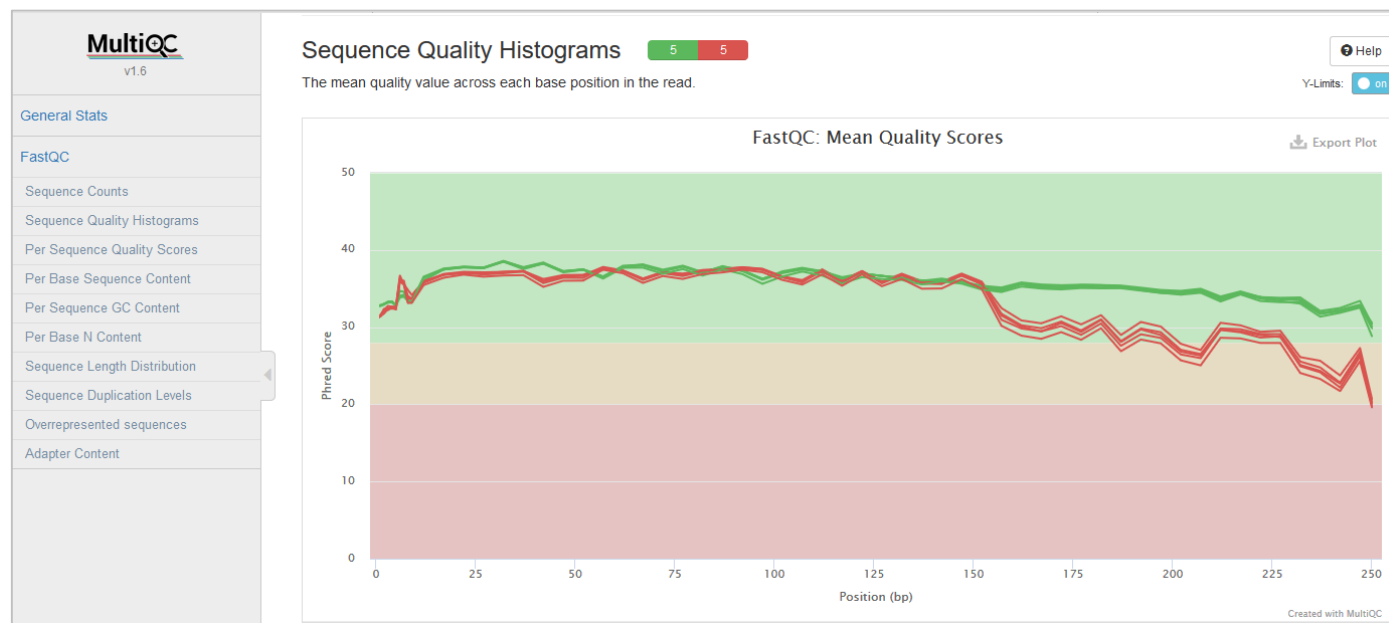
Per position sequence content (FastQC)



- **Enrichment of k-mers at the 5' end due to use of random hexamers or transposases in the library preparation**
- **Typical for RNA-seq data**
- **Can't be corrected, doesn't usually effect the analysis**

I have many FASTQ files – how can I quickly check them all?

- Make a tar package of all the FASTQ files using the tool **Utilities / Make a tar package**
- Select the tar package and run the tool **Quality control / Read quality with MultiQC for many FASTQ files**



Was your data made with stranded protocol?

- **You need to indicate it when**
 - aligning reads to genome (e.g. HISAT2)
 - counting reads per genes (e.g. HTSeq)
- **If you don't know if a stranded sequencing protocol was used, you can check it**
 - Select your FASTQ file and run the tool Quality control / RNA-seq strandedness inference and inner distance estimation using RseQC
 - aligns a subset of the reads to genome and compares the locations to reference annotation
- **For more info please see the manual**
 - <http://chipster.csc.fi/manual/library-type-summary.html>

RseQC strandedness report

Visualisation

View text ▼

This is SingleEnd Data

Fraction of reads failed to determine: 0.0433

Fraction of reads explained by "++,--": 0.9498

Fraction of reads explained by "+-,+-": 0.0069

It seems the data is stranded. Read is always on the same strand as the gene.

Corresponding parameters are:

TopHat, HISAT2, Cufflinks and Cuffdiff: library-type fr-secondstrand

HTSeq: stranded -- yes

RSeQC: ++,--

Input files were assigned as follows:

Read 1 file: hESC.fastq

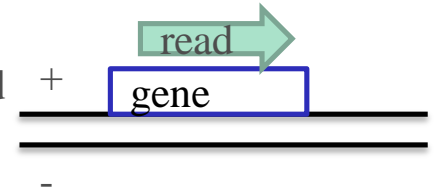
What does this ++, - - mean?

Single end:

++,--

read mapped to '+' strand indicates parental gene on '+' strand

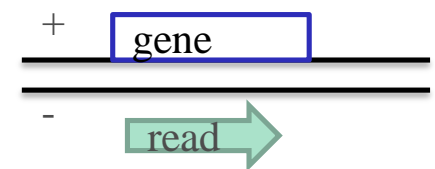
read mapped to '-' strand indicates parental gene on '-' strand



+-,-+

read mapped to '+' strand indicates parental gene on '-' strand

read mapped to '-' strand indicates parental gene on '+' strand



Paired end:

1++,1-,2+-,2-+

read1 mapped to '+' strand indicates parental gene on '+' strand

read1 mapped to '-' strand indicates parental gene on '-' strand

read2 mapped to '+' strand indicates parental gene on '-' strand

read2 mapped to '-' strand indicates parental gene on '+' strand

1+-,1-+,2++,2--

read1 mapped to '+' strand indicates parental gene on '-' strand

read1 mapped to '-' strand indicates parental gene on '+' strand

read2 mapped to '+' strand indicates parental gene on '+' strand

read2 mapped to '-' strand indicates parental gene on '-' strand



RNA-seq data analysis workflow

- Quality control of raw reads
- **Preprocessing (trimming / filtering) if needed**
- Alignment (=mapping) to reference genome
- Alignment level quality control
- Quantitation
- Experiment level quality control
- Differential expression analysis
- Visualization of reads and results in genomic context

Filtering vs trimming

- **Filtering removes the entire read**
- **Trimming removes only the bad quality bases**
 - It can remove the entire read, if all bases are bad
- **Trimming makes reads shorter**
 - This might not be optimal for some applications
- **Paired end data: the matching order of the reads in the two files has to be preserved**
 - If a read is removed, its pair has to be removed as well

What base quality threshold should be used?

- No consensus
- Trade-off between having good quality reads and having enough sequence
- Start with gentle trimming and check with FastQC

An Extensive Evaluation of Read Trimming Effects on Illumina NGS Data Analysis

Cristian Del Fabbro¹, Simone Scalabrin², Michele Morgante¹, Federico M. Giorgi^{1,3*}

¹ Institute of Applied Genomics, Udine, Italy, ² IGA Technology Services, Udine, Italy, ³ Center for Computational Biology and Bioinformatics, Columbia University, New York, New York, United States of America

frontiers in
GENETICS

ORIGINAL RESEARCH ARTICLE

published: 31 January 2014
doi: 10.3389/fgene.2014.00013

On the optimal trimming of high-throughput mRNA sequence data

Matthew D. MacManes^{1,2*}

¹ Department of Molecular, Cellular and Biomedical Sciences, University of New Hampshire, Durham, NH, USA

² Hubbard Center for Genome Studies, Durham, NH, USA

Software packages for preprocessing

- **Trimmomatic**
- **FastX**
- **PRINSEQ**
- **TagCleaner**
- **...**

Trimmomatic options in Chipster

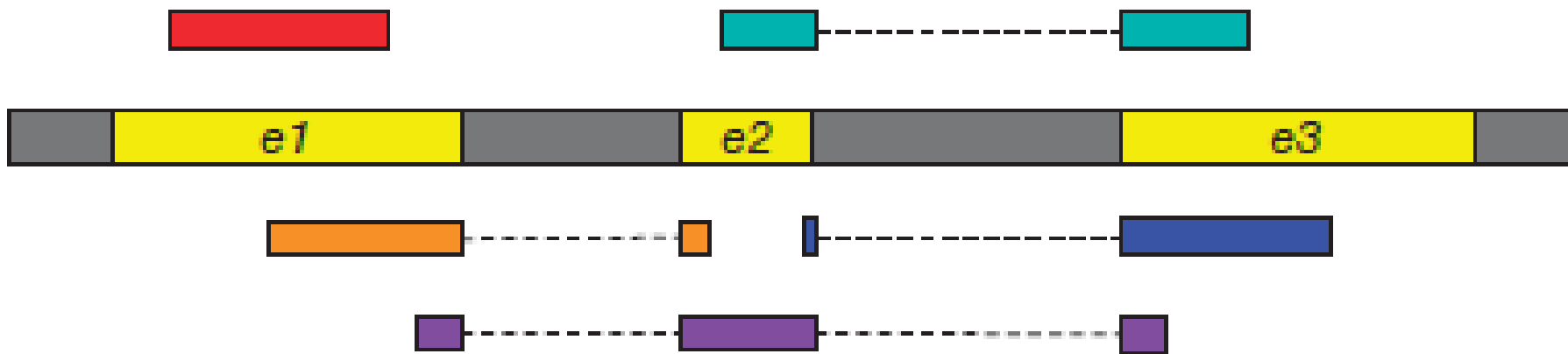
- **Adapters**
- **Minimum quality**
 - Per base, one base at a time or in a sliding window, from 3' or 5' end
 - Per base adaptive quality trimming (balance length and errors)
 - Minimum (mean) base quality
- **Trim x bases from left/ right**
- **Minimum read length after trimming**
- **Copes with paired end data**

RNA-seq data analysis workflow

- Quality control of raw reads
- Preprocessing (trimming / filtering) if needed
- **Alignment (=mapping) to reference genome**
- Alignment level quality control
- Quantitation
- Experiment level quality control
- Visualization of reads and results in genomic context
- Differential expression analysis

Aligning reads to reference genome

- **The goal is to find the location where a read originated from**
- **Challenges**
 - Reads contain genomic variants and sequencing errors
 - Genomes contain non-unique sequence and introns
- **RNA-seq aligner needs to be able to map splice junction spanning reads to genome non-contiguously**
 - Spliced alignments are difficult because sequence signals at splice sites are limited, and introns can be thousands of bases long



Alignment programs

- **Many aligners have been developed over the years**
 - Convert genome fasta file to a data structure which is faster to search (e.g. BWT index or suffix array)
 - Differ in speed, memory requirements, accuracy and ability to deal with spliced alignments
- **Use splice-aware aligner for mapping RNA-seq reads**
 - Examples:
 - STAR (fast and accurate, needs a lot of memory)
 - HISAT2 (fast and accurate, creating the genomic index needs a LOT of memory)
 - TopHat2 (slower, needs less memory)

Splice-aware aligners in Chipster

➤ **STAR**

- Human genome available

➤ **HISAT2**

- Human and mouse genome available
- You can also supply own genome if it is small

➤ **TopHat2**

- Many genomes available
- You can also supply own genome

➤ **Output files**

- BAM = contains the alignments
- bai = index file for BAM, required by genome browsers etc
- log = useful information about the alignment run

HISAT2

- **HISAT = Hierarchical Indexing for Spliced Alignment of Transcripts**
- **Fast spliced aligner with low memory requirement**
- **Reference genome is (BWT FM) indexed for fast searching**
 - Currently Chipster offers human and mouse reference genome
 - Let us know if you need others!
 - You can provide own (small) reference genome in fasta format
- **Uses two types of indexes**
 - A global index: used to anchor a read in genome (28 bp is enough)
 - Thousands of small local indexes, each covering a genomic region of 56 Kbp: used for rapid extension of alignments (good for spliced reads with short anchors)
- **Uses splice site information found during the alignment of earlier reads in the same run**

HISAT2 parameters

Analysis tools - Alignment - HISAT2 for paired end reads	
Genome	Homo_sapiens.G... ▼
Library type	fr-unstranded ▼
How many hits to report per read	5 ▲▼
Base quality encoding used	Sanger - Phred+... ▼
Minimum intron length	20 ▲▼
Maximum intron length	500000 ▲▼
Disallow soft-clipping	Use soft-clipping ▼
Require long anchor lengths for subsequent assembly	Don't require ▼

- Remember to set the strandedness (library type) correctly!
- Note that there can be alignments that are better than the 5 reported ones
- Require long anchors (> 16 bp) if you are going to do transcript assembly
- Soft-clipping = read ends don't need to align to the genome, if this maximizes the alignment score

STAR

- **STAR = Spliced Transcripts Alignment to a Reference**
- **Reference genome fasta is converted to a suffix array for fast searching**
- **2-pass mapping process**
 - splice junctions found during the 1st pass are inserted into the genome index, and all reads are re-mapped in the 2nd mapping pass
 - this doesn't increase the number of detected novel junctions, but it allows more spliced reads mapping to novel junctions.
- **Maximum alignments per read -parameter sets the maximum number of loci the read is allowed to map to**
 - Alignments (all of them) will be output only if the read maps to no more loci than this. Otherwise no alignments will be output.
- **Chipster offers an Ensembl GTF file to detect annotated splice junctions**
 - you can also give your own, e.g. GENCODE GTF



What if my sample has several FASTQ files?

- **Align all of them together**
- **Single end data: Select all the FASTQ files for the sample**
- **Paired end data: Make filename list files first**
 - Select all the read1 files and run the tool "Utilities / Make a list of file names"
 - Repeat with all the read2 files
 - Select all the FASTQ files and both filename list files and run HISAT2/STAR (check that the files have been assigned correctly)

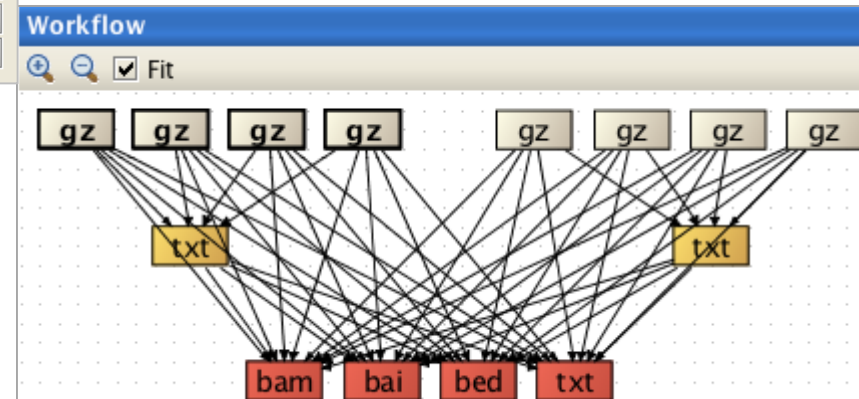
Input datasets

Reads

sample1_S1_L001_R1_001.fastq.gz
sample1_S1_L001_R2_001.fastq.gz
sample1_S1_L002_R1_001.fastq.gz
sample1_S1_L002_R2_001.fastq.gz
sample1_S1_L003_R1_001.fastq.gz
sample1_S1_L003_R2_001.fastq.gz
sample1_S1_L004_R1_001.fastq.gz
sample1_S1_L004_R2_001.fastq.gz
R1files.txt
R2files.txt

List of read 1 files
List of read 2 files
Optional GTF file

R1files.txt
R2files.txt



File format for mapped reads: BAM/SAM

Visualisation

BAM viewer

Maximise Detach

```
@HD      VN:1.5      SO:coordinate
@SQ      SN:1       LN:248956422
@SQ      SN:2       LN:242193529
@SQ      SN:3       LN:198295559
@SQ      SN:4       LN:190214555
@SQ      SN:5       LN:181538259
@SQ      SN:6       LN:170805979
@SQ      SN:7       LN:159345973
@SQ      SN:8       LN:145138636
@SQ      SN:9       LN:138394717
@SQ      SN:10      LN:133797422
@SQ      SN:11      LN:135086622
@SQ      SN:12      LN:133275309
@SQ      SN:13      LN:114364328
@SQ      SN:14      LN:107043718
@SQ      SN:15      LN:101991189
@SQ      SN:16      LN:90338345
@SQ      SN:17      LN:83257441
@SQ      SN:18      LN:80373285
@SQ      SN:19      LN:58617616
@SQ      SN:20      LN:64444167
@SQ      SN:21      LN:46709983
@SQ      SN:22      LN:50818468
@SQ      SN:X       LN:156040895
@SQ      SN:Y       LN:57227415
@SQ      SN:MT      LN:16569
@PG      ID:hisat2  PN:hisat2  VN:2.1.0  CL:"/opt/chipster/tools/hisat2/hisat2-align-s --wrapper basic-0 --phred33
--min-intronlen 20 --max-intronlen 500000 -x Homo_sapiens.GRCh38.92 -k 5 -p 16 --passthrough -l lung3e_1.fastq.gz -2
lung3e_2.fastq.gz"
ERR315346.13741151 355 1 11591 1 101M = 11641 151
GTTCTGTATCCCACCAGCAATGTCTAGGAATGCCTGCTTCTCCACAAAGTGTCTTACTTTTGGATTTTGGCCAGTCTAACAGGTAAAGCCCTGGAGATTCTT
BBBFFFFFFFFFFFFFFFIIIIIFIIIIIBFFIIIIIIIIIIIF'BFBBFFIIIIIIIBBFFFFIIFIIIIIIIFBFF<BFBFFFFFFFFFBBBFFFFFBB<B<BBBBF MD:Z:36T46G17
XG:i:0 NH:i:4 NM:i:2 XM:i:2 XN:i:0 XO:i:0 AS:i:-7 YS:i:-5 ZS:i:-7 YT:Z:CP
```

← BAM header

alignment information: one line per read alignment, containing 11 mandatory fields, followed by optional tags

Fields in BAM/SAM files

- **read name** HWI-EAS229_1:2:40:1280:283
- **flag** 272
- **reference name** 1
- **position** 18506
- **mapping quality** 0
- **CIGAR** 49M6183N26M
- **mate name** *
- **mate position** 0
- **insert size** 0
- **sequence**
AGGGCCGATCTTGGTGCCATCCAGGGGGGCCTCTACAAGGAT
AATCTGACCTGCTGAAGATGTCTCCAGAGACCTT
- **base qualities**
ECC@EEF@EB:EECFEECCCBEEEE;>5;2FBB@FBFEEFCF@F
FFFCEFFFFEE>FFEFC=@A;@>1@6.+5/5
- **tags** MD:Z:75 NH:i:7 AS:i:-8 XS:A:-



@HD VN:1.5 SO:coordinate

@SQ SN:ref LN:45

```
r001 99 ref 7 30 8M2I4M1D3M = 37 39 TTAGATAAAGGATACTG *
r002 0 ref 9 30 3S6M1P1I4M * 0 0 AAAAGATAAGGATA *
r003 0 ref 9 30 5S6M * 0 0 GCCTAAGCTAA * SA:Z:ref,29,-,6H5M,17,0;
r004 0 ref 16 30 6M14N5M * 0 0 ATAGCTTCAGC *
r003 2064 ref 29 17 6H5M * 0 0 TAGGC * SA:Z:ref,9,+,5S6M,30,1;
r001 147 ref 37 30 9M = 7 -39 CAGCGGCAT * NM:i:1
```

Header
section

Alignment
section

Optional fields in the format of TAG:TYPE:VALUE

QUAL: read quality; * meaning such information is not available

SEQ: read sequence

TLEN: the number of bases covered by the reads from the same fragment. Plus/minus means the current read is the leftmost/rightmost read. E.g. compare first and last lines.

PNEXT: Position of the primary alignment of the NEXT read in the template. Set as 0 when the information is unavailable. It corresponds to POS column.

RNEXT: reference sequence name of the primary alignment of the NEXT read. For paired-end sequencing, NEXT read is the paired read, corresponding to the RNAME column.

CIGAR: summary of alignment, e.g. insertion, deletion

MAPQ: mapping quality

POS: 1-based position

RNAME: reference sequence name, e.g. chromosome/transcript id

FLAG: indicates alignment information about the read, e.g. paired, aligned, etc.

QNAME: query template name, aka. read ID

➤ Really nice pages for SAM/BAM interpretation:
<http://www.samformat.info>



Mapping quality

- **Confidence in read's point of origin**
- **Depends on many things, including**
 - uniqueness of the aligned region in the genome
 - length of alignment
 - number of mismatches and gaps
- **Expressed in Phred scores, like base qualities**
 - $Q = -10 * \log_{10} (\text{probability that mapping location is wrong})$
- **Values differ in different aligners. E. g. unique mapping is**
 - 60 in HISAT2
 - 255 in STAR
 - 50 in TopHat
 - <https://sequencing.qcfail.com/articles/mapq-values-are-really-useful-but-their-implementation-is-a-mess/>



CIGAR string

- M = match or mismatch
- I = insertion
- D = deletion
- N = intron (in RNA-seq read alignments)
- S = soft clip (ignore these bases)
- H = hard clip (ignore and remove these bases)

- Example:

@HD VN:1.3 SO:coordinate

@SQ SN:ref LN:45

r001 163 ref 7 30 8M2I4M1D3M = 37 39 TTAGATAAAGGATACTG *

- The corresponding alignment

Ref	AGCATG	TTAGATAA**GATAGCTG	TGCTAGTAGGCAGTCAGCGCCAT
r001		TTAGATAAAGGATA*CTG	

Flag field in BAM

➤ Read's flag number is a sum of values

- E.g. 4 = unmapped, 1024 = duplicate
- Explained in detail at <http://samtools.github.io/hts-specs/SAMv1.pdf>
- You can interpret them at <http://broadinstitute.github.io/picard/explain-flags.html>

This utility explains SAM flags in plain English.
It also allows switching easily from a read to its mate.

Flag:

Explanation:

- ☒ read paired
- ☒ read mapped in proper pair
- ☐ read unmapped
- ☐ mate unmapped
- ☒ read reverse strand
- ☐ mate reverse strand
- ☐ first in pair
- ☒ second in pair
- ☒ not primary alignment
- ☐ read fails platform/vendor quality checks
- ☐ read is PCR or optical duplicate
- ☐ supplementary alignment



How did the alignment go? Check the log file

- **How many reads mapped to the reference?**
 - How many of them mapped uniquely?
- **How many pairs mapped?**
 - How many pairs mapped concordantly?
- **What was the overall alignment rate?**

```
Visualisation
View text
25354832 reads; of these:
  25354832 (100.00%) were paired; of these:
    6098272 (24.05%) aligned concordantly 0 times
    18567284 (73.23%) aligned concordantly exactly 1 time
    689276 (2.72%) aligned concordantly >1 times
    ----
    6098272 pairs aligned concordantly 0 times; of these:
      724806 (11.89%) aligned discordantly 1 time
    ----
    5373466 pairs aligned 0 times concordantly or discordantly; of these:
      10746932 mates make up the pairs; of these:
        8812069 (82.00%) aligned 0 times
        1800817 (16.76%) aligned exactly 1 time
        134046 (1.25%) aligned >1 times
82.62% overall alignment rate
```



Log file by STAR

Visualisation

View text

```
Started job on | Feb 17 12:38:11
Started mapping on | Feb 17 12:47:47
Finished on | Feb 17 12:52:32
Mapping speed, Million of reads per hour | 320.27
```

```
Number of input reads | 25354832
Average input read length | 202
UNIQUE READS:
Uniquely mapped reads number | 20409554
Uniquely mapped reads % | 80.50%
Average mapped length | 197.39
Number of splices: Total | 12378576
Number of splices: Annotated (sjdb) | 12378175
Number of splices: GT/AG | 12272618
Number of splices: GC/AG | 89423
Number of splices: AT/AC | 9589
Number of splices: Non-canonical | 6946
Mismatch rate per base, % | 0.39%
Deletion rate per base | 0.01%
Deletion average length | 1.75
Insertion rate per base | 0.01%
Insertion average length | 1.36
```

```
MULTI-MAPPING READS:
Number of reads mapped to multiple loci | 970016
% of reads mapped to multiple loci | 3.83%
Number of reads mapped to too many loci | 11610
% of reads mapped to too many loci | 0.05%
```

```
UNMAPPED READS:
% of reads unmapped: too many mismatches | 0.00%
% of reads unmapped: too short | 15.55%
% of reads unmapped: other | 0.08%
```

```
CHIMERIC READS:
Number of chimeric reads | 0
% of chimeric reads | 0.00%
```



Other tools for checking BAM files

- **Count alignments in BAM**
 - How many alignments does the BAM contain.
 - Includes an optional mapping quality filter.
- **Count alignments per chromosome in BAM**
- **Count alignment statistics for BAM**
- **Collect multiple metrics for BAM**

Visualisation	
View text	▼
<pre>45141520 + 0 in total (QC-passed reads + QC-failed reads) 0 + 0 duplicates 45141520 + 0 mapped (100.00%:-nan%) 45141520 + 0 paired in sequencing 22772818 + 0 read1 22368702 + 0 read2 41537534 + 0 properly paired (92.02%:-nan%) 43544007 + 0 with itself and mate mapped 1597513 + 0 singletons (3.54%:-nan%) 266664 + 0 with mate mapped to a different chr 186766 + 0 with mate mapped to a different chr (mapQ>=5)</pre>	



Tools for manipulating BAM files

➤ **Make a subset of BAM**

- Retrieve alignments for a given chromosome/region, e.g. chr1:100-1000
- Can filter based on mapping quality

➤ **Index BAM**

➤ **Convert SAM to BAM, sort and index BAM**

- "Preprocessing" when importing SAM/BAM, runs on your computer.
- The tool available in the "Utilities" category runs on the server

RNA-seq data analysis workflow

- Quality control of raw reads
- Preprocessing (trimming / filtering) if needed
- Alignment (=mapping) to reference genome
- **Alignment level quality control**
- Quantitation
- Experiment level quality control
- Differential expression analysis
- Visualization of reads and results in genomic context

Annotation-based quality metrics

➤ **Saturation of sequencing depth**

- Would more sequencing detect more genes and splice junctions?

➤ **Read distribution between different genomic features**

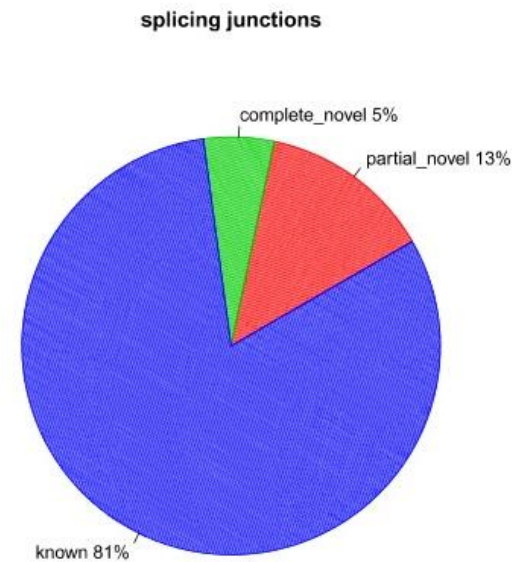
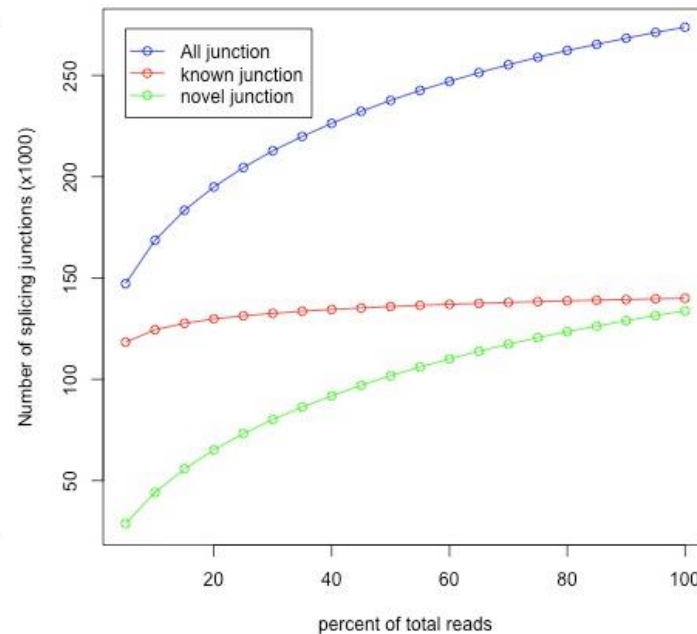
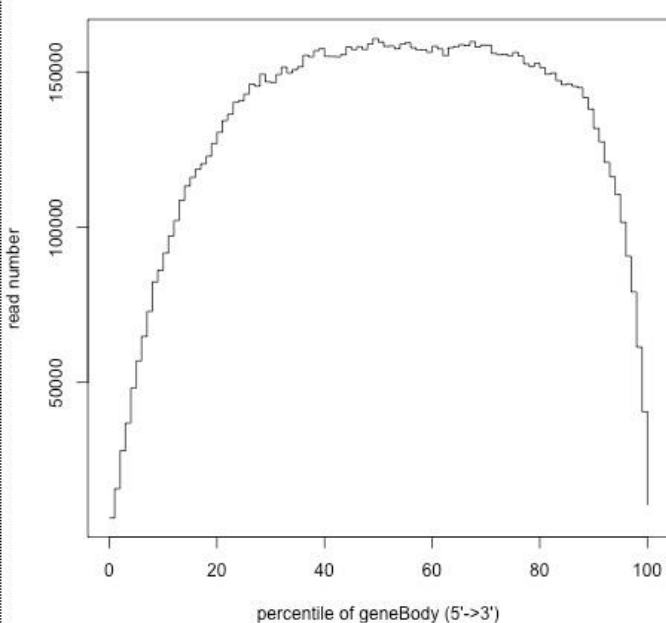
- Exonic, intronic, intergenic regions
- Coding, 3' and 5' UTR exons
- Protein coding genes, pseudogenes, rRNA, miRNA, etc

➤ **Is read coverage uniform along transcripts?**

- Biases introduced in library construction and sequencing
 - polyA capture and polyT priming can cause 3' bias
 - random primers can cause sequence-specific bias
 - GC-rich and GC-poor regions can be under-sampled
- Genomic regions have different mappabilities (uniqueness)

Quality assessment with RseQC

- Checks coverage uniformity, saturation of sequencing depth, novelty of splice junctions, read distribution between different genomic regions, etc.
- Takes a BAM file and a BED file
 - Chipster has BED files available for several organisms
 - You can also use your own BED if you prefer



BED file format

- **BED (Browser extensible data) file format is used for reporting location of features (e.g. genes and exons) in a genome**
- **5 obligatory columns: chr, start, end, name, score**
- **0-based, like BAM**

column0	column1	column2	column3	column4
chr22	21022480	21024796	JUNC00000001	1
chr19	201609	201783	JUNC00000002	5
chr19	281478	282180	JUNC00000003	3
chr19	282242	282811	JUNC00000004	21
chr19	282751	287541	JUNC00000005	37
chr19	287705	288084	JUNC00000006	6
chr19	288105	291354	JUNC00000007	18
chr19	307484	308600	JUNC00000008	1
chr19	308603	308858	JUNC00000009	2
chr19	308868	311907	JUNC00000010	13
chr19	311872	312256	JUNC00000011	26
chr19	312205	313558	JUNC00000012	22
chr19	313575	325706	JUNC00000013	68

Own BED? Check chromosome names

- **RseQC needs the same chromosome naming in BAM and BED**
- **Chromosome names in BED files can have the prefix “chr”**
 - e.g. chr1
- **Chipster BAM files are Ensembl-based and don't have the prefix**
 - If you use your own BED (e.g. from UCSC Table browser) you need to remove the prefix (chr1 → 1)
- **Use the tool **Utilities / Modify text** with the following parameters:**
 - Operation = Replace text
 - Search string = chr
 - Input file format = BED

QC tables by RseQC

```

=====
#All numbers are READ count (alignment, actually...)
=====

Total records:                103284

QC failed:                    0
Optical/PCR duplicate:        0
Non primary hits              18476
Unmapped reads:               0
mapq < mapq_cut (non-unique): 4208
                               Default=30
mapq >= mapq_cut (unique):    80600
Read-1:                       0
Read-2:                       0
Reads map to '+':             48292
Reads map to '-':             32308
Non-splice reads:             50919
Splice reads:                 29681
Reads mapped in proper pairs:  0
Proper-paired reads map to different chrom:0
    
```

read_distribution:

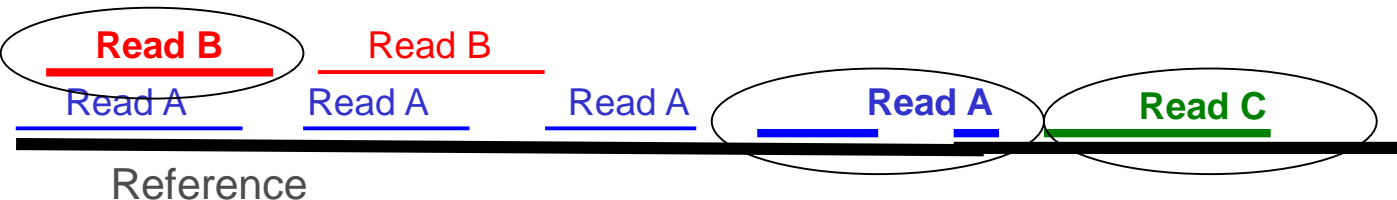
```

Total Reads                84808
Total Tags                 116738
Total Assigned Tags          111352
    
```

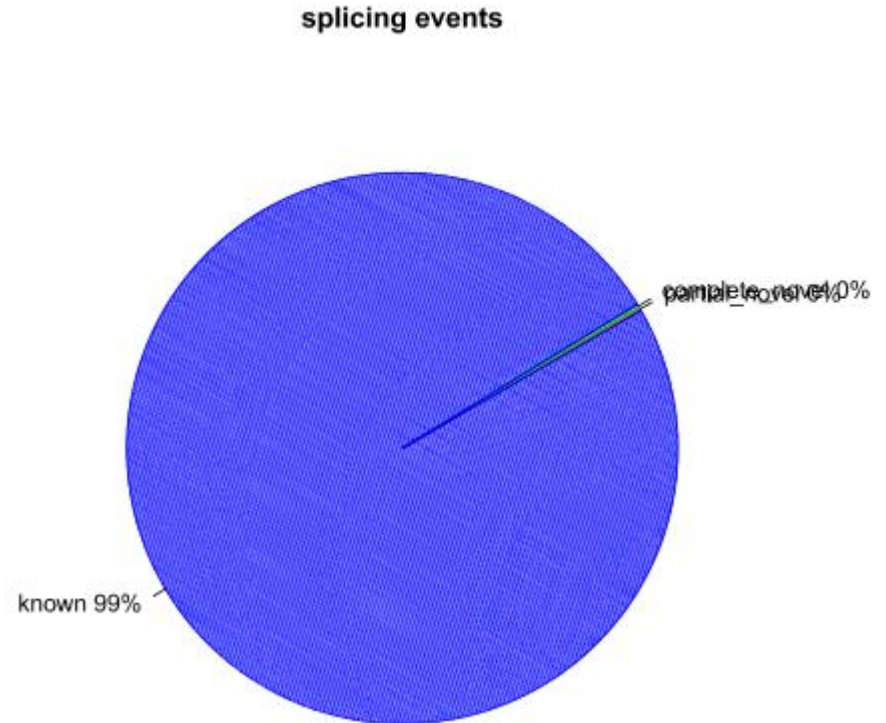
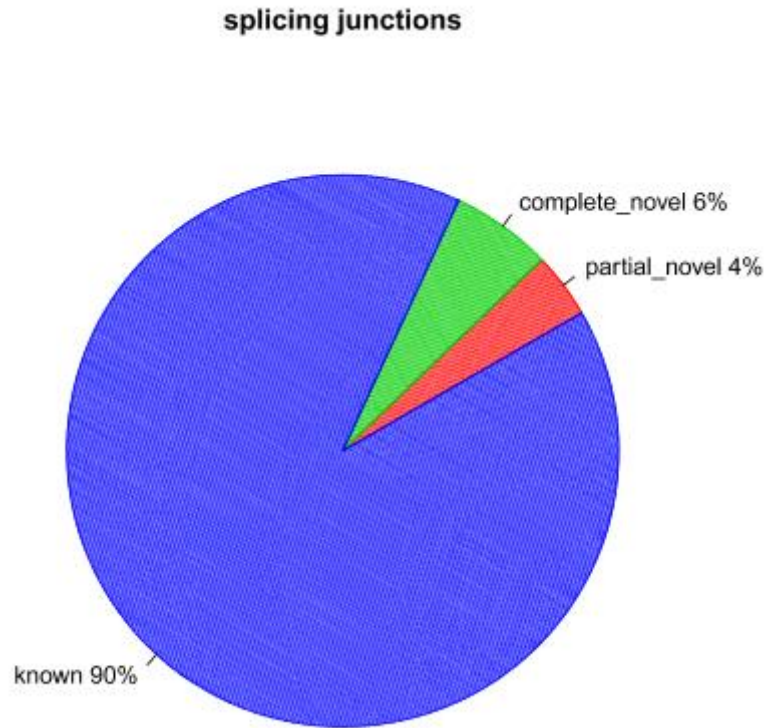
Group	Total_bases	Tag_count	Tags/Kb
CDS_Exons	2211343	90961	41.13
5'UTR_Exons	529860	1662	3.14
3'UTR_Exons	1415234	12423	8.78
Introns	25801210	5349	0.21
TSS_up_1kb	1295771	31	0.02
TSS_up_5kb	5332522	321	0.06
TSS_up_10kb	8804879	584	0.07
TES_down_1kb	1292506	217	0.17
TES_down_5kb	5108821	344	0.07
TES_down_10kb	8282641	373	0.05

```

Total records:      7
Non primary hits:   4
Total reads:        3
Total tags:         8
    
```



Splicing graphs by RseQC



- **Splicing junction = exon-exon junction covered by one or more reads**
- **Splicing event = a read is split across a splice junction**

Did I accidentally sequence ribosomal RNA?

- **The majority of RNA in cells is rRNA**
- **Typically we want to sequence protein coding genes, so we try to avoid rRNA**
 - polyA capture
 - Ribominus kit (may not work consistently between samples)
- **How to check if we managed to avoid rRNA?**
 - RseQC might not be able to tell, if the rRNA genes are not in the BED file (e.g. in human the rRNA gene repeating unit has not been assigned to any chromosome yet)
 - You can map the reads to human ribosomal DNA repeating unit sequence (instead of the genome) with the Bowtie aligner, and check the alignment percentage

RNA-seq data analysis workflow

- Quality control of raw reads
- Preprocessing (trimming / filtering) if needed
- Alignment (=mapping) to reference genome
- Alignment level quality control
- **Quantitation**
- Describing the experiment with phenodata
- Experiment level quality control
- Differential expression analysis
- Visualization of reads and results in genomic context

Software for counting reads per genes or transcripts

- **HTSeq**
- **Cufflinks**
- **StringTie**
- **Kallisto**
- **Salmon**

Counting reads per genes with HTSeq

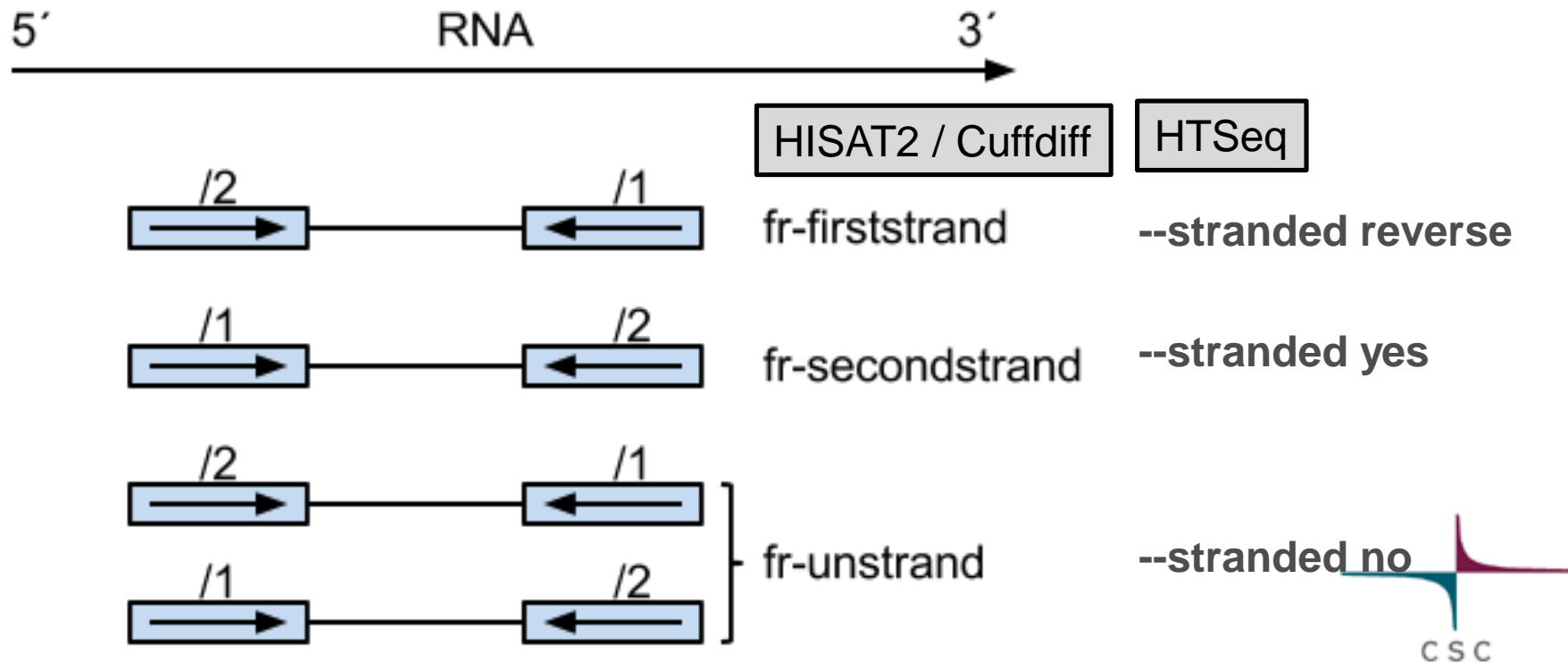
- **Given a BAM file and a list of gene locations, counts how many reads map to each gene.**
 - A gene is considered as the union of all its exons.
 - Reads can be counted also per exons.
- **Locations need to be supplied in GTF file**
 - Note that GTF and BAM must use the same chromosome naming
- **Multimapping reads and ambiguous reads are not counted**
- **3 modes to handle reads which overlap several genes**
 - Union (default), Intersection-strict, Intersection-nonempty
- **Attention: was your data made with stranded protocol?**
 - You need to select the right counting mode!

Stranded / directional RNA-seq data

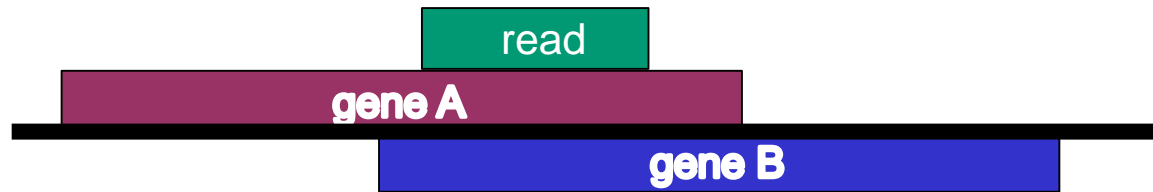
➤ Several protocols available

- TruSeq stranded, NEB Ultra Directional, Agilent SureSelect Strand-Specific...

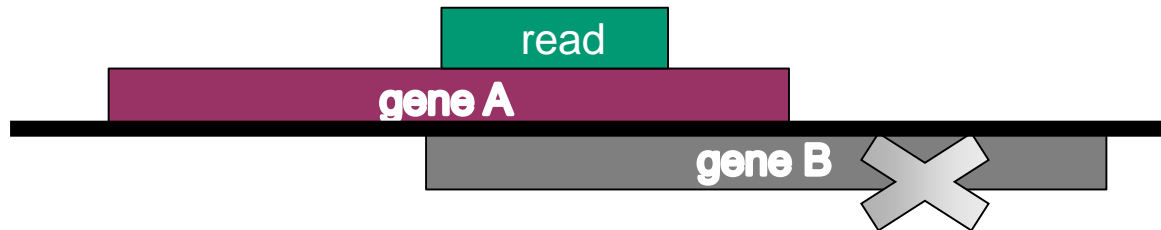
➤ Make sure that you set the strandedness parameter correctly



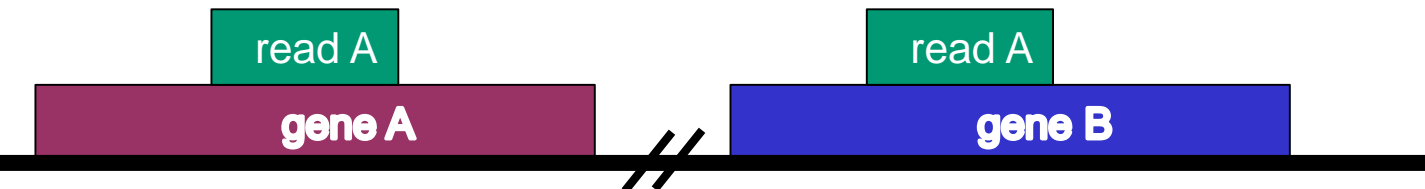
Not unique or ambiguous?



Ambiguous



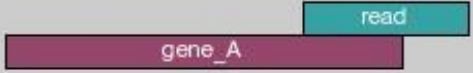

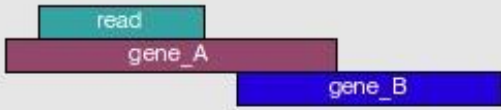
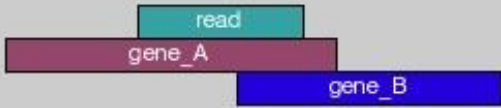
Stranded data
→ Not ambiguous



Multimapping
(not unique)



HTSeq count modes

	union	intersection_strict	intersection_nonempty
	gene_A	gene_A	gene_A
	gene_A	no_feature	gene_A
	gene_A	no_feature	gene_A
	gene_A	gene_A	gene_A
	gene_A	gene_A	gene_A
	ambiguous	gene_A	gene_A
	ambiguous	ambiguous	ambiguous

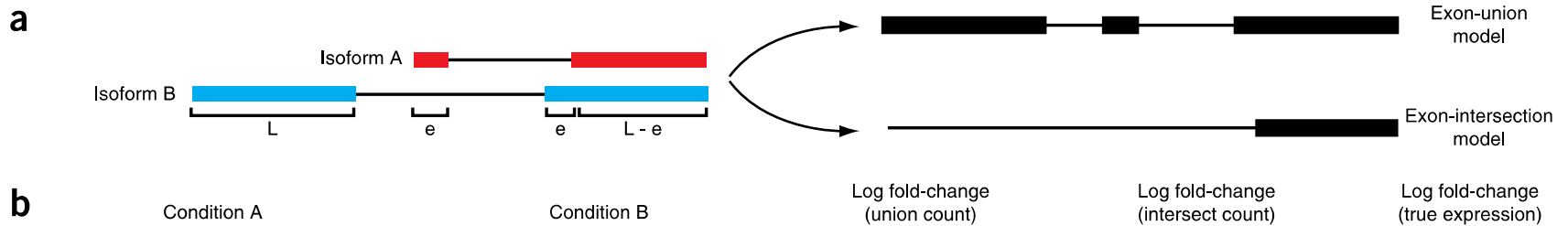
GTF file format

- **9 obligatory columns: chr, source, name, start, end, score, strand, frame, attribute**
- **1-based**
- **For HTSeq to work, all exons of a gene must have the same gene_id**
 - Use GTFs from Ensembl, avoid UCSC

chr1	unknown	exon	14362	14829	.	-	.	gene_id "WASH7P"; gene_name "WASH7P"; transcript_id "NR_024540"; tss_id "TSS7245";
chr1	unknown	exon	14970	15038	.	-	.	gene_id "WASH7P"; gene_name "WASH7P"; transcript_id "NR_024540"; tss_id "TSS7245";
chr1	unknown	exon	15796	15947	.	-	.	gene_id "WASH7P"; gene_name "WASH7P"; transcript_id "NR_024540"; tss_id "TSS7245";
chr1	unknown	exon	16607	16765	.	-	.	gene_id "WASH7P"; gene_name "WASH7P"; transcript_id "NR_024540"; tss_id "TSS7245";
chr1	unknown	exon	16858	17055	.	-	.	gene_id "WASH7P"; gene_name "WASH7P"; transcript_id "NR_024540"; tss_id "TSS7245";
chr1	unknown	exon	17233	17368	.	-	.	gene_id "WASH7P"; gene_name "WASH7P"; transcript_id "NR_024540"; tss_id "TSS7245";
chr1	unknown	exon	17606	17742	.	-	.	gene_id "WASH7P"; gene_name "WASH7P"; transcript_id "NR_024540"; tss_id "TSS7245";
chr1	unknown	exon	17915	18061	.	-	.	gene_id "WASH7P"; gene_name "WASH7P"; transcript_id "NR_024540"; tss_id "TSS7245";
chr1	unknown	exon	18268	18366	.	-	.	gene_id "WASH7P"; gene_name "WASH7P"; transcript_id "NR_024540"; tss_id "TSS7245";
chr1	unknown	exon	24738	24891	.	-	.	gene_id "WASH7P"; gene_name "WASH7P"; transcript_id "NR_024540"; tss_id "TSS7245";
chr1	unknown	exon	29321	29370	.	-	.	gene_id "WASH7P"; gene_name "WASH7P"; transcript_id "NR_024540"; tss_id "TSS7245";

Estimating gene expression at gene level

- the isoform switching problem



$$\log_2\left(\frac{5}{10}\right) = -1$$

$$\log\left(\frac{4}{5}\right) = -0.1$$

$$\log_2\left(\frac{\frac{5}{L}}{\frac{10}{2L}}\right) = 0$$

Trapnell et al. Nature Biotechnology 2013



RNA-seq data analysis workflow

- Quality control of raw reads
- Preprocessing (trimming / filtering) if needed
- Alignment (=mapping) to reference genome
- Alignment level quality control
- Quantitation
- **Describing the experiment with phenodata**
- Experiment level quality control
- Differential expression analysis
- Visualization of reads and results in genomic context

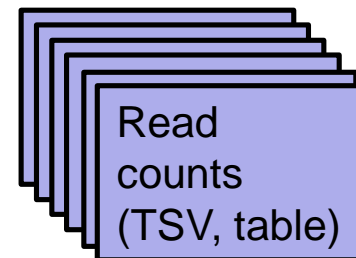
Combine individual count files into a count table

- Select all the count files and run “Utilities / Define NGS experiment”
- This creates a table of counts and a phenodata file, where you can describe experimental groups

						Control 1
Gene	Gene	Gene	Gene	Gene A	Gene A	6
Gene	Gene	Gene	Gene	Gene B	Gene B	11
Gene	Gene	Gene	Gene	Gene C	Gene C	200
Gene	Gene	Gene	Gene	Gene D	Gene D	0



	Control 1	Control 2	Control 3	Sample 1	Sample 2	Sample 3
Gene A	6	5	7	17	10	11
Gene B	11	11	10	3	4	2
Gene C	200	150	355	50	1	3
Gene D	0	1	0	2	0	1



Read table
(TSV, table)

phenodata

Phenodata file: describe the experiment

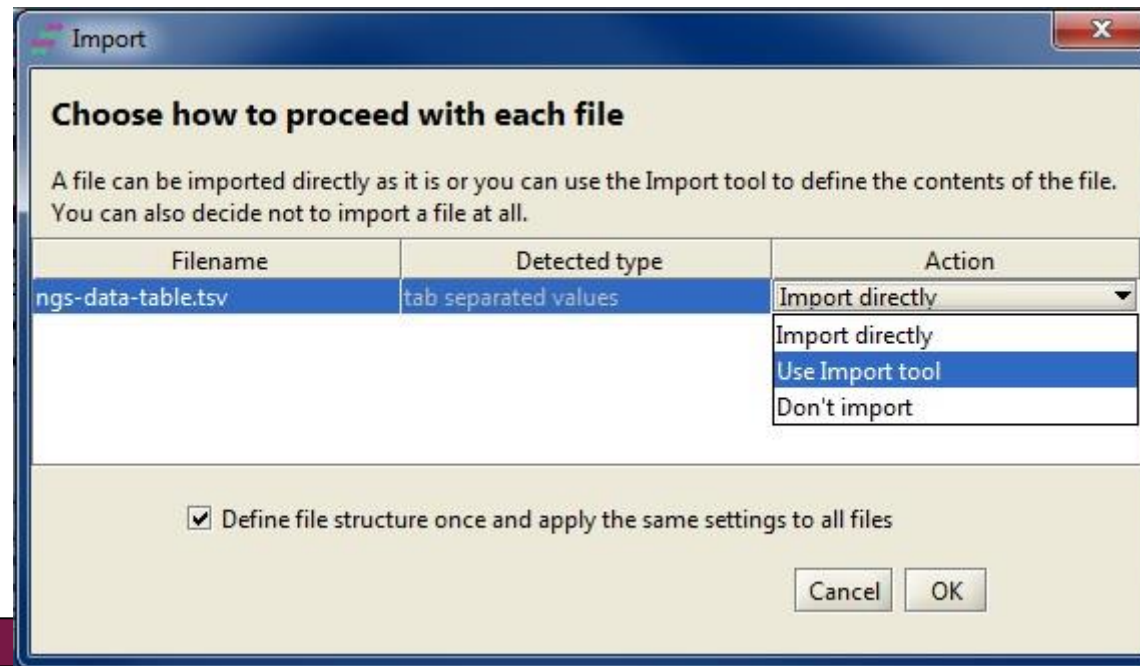
- **Describe experimental groups, time, pairing etc with numbers**
 - e.g. 1 = control, 2 = cancer
- **Define sample names for visualizations in the Description column**



sample	original_name	description	patient	group	treatment	time	hours
ngs001.tsv	SRR479052	1_C_24	1	1	Control	1	24h
ngs002.tsv	SRR479053	1_C_48	1	1	Control	2	48h
ngs003.tsv	SRR479054	1_DP_24	1	2	DPN	1	24h
ngs004.tsv	SRR479055	1_DP_48	1	2	DPN	2	48h
ngs007.tsv	SRR479058	2_C_24	2	1	Control	1	24h
ngs008.tsv	SRR479059	2_C_48	2	1	Control	2	48h
ngs009.tsv	SRR479060	2_DP_24	2	2	DPN	1	24h
ngs011.tsv	SRR479062	2_DP_48	2	2	DPN	2	48h
ngs015.tsv	SRR479066	3_C_24	3	1	Control	1	24h
ngs016.tsv	SRR479067	3_C_48	3	1	Control	2	48h
ngs017.tsv	SRR479068	3_DP_24	3	2	DPN	1	24h
ngs018.tsv	SRR479069	3_DP_48	3	2	DPN	2	48h

What if somebody gives you a count table?

- **Make sure that the filename ending is tsv**
- **When importing the file to Chipster select “Use Import tool”**
- **In Import tool**
 - Mark the title row
 - Mark the identifier column and the count columns
- **Select the imported files and run the tool “Utilities / Preprocess count table”**
 - This creates a count table and a phenodata file for it



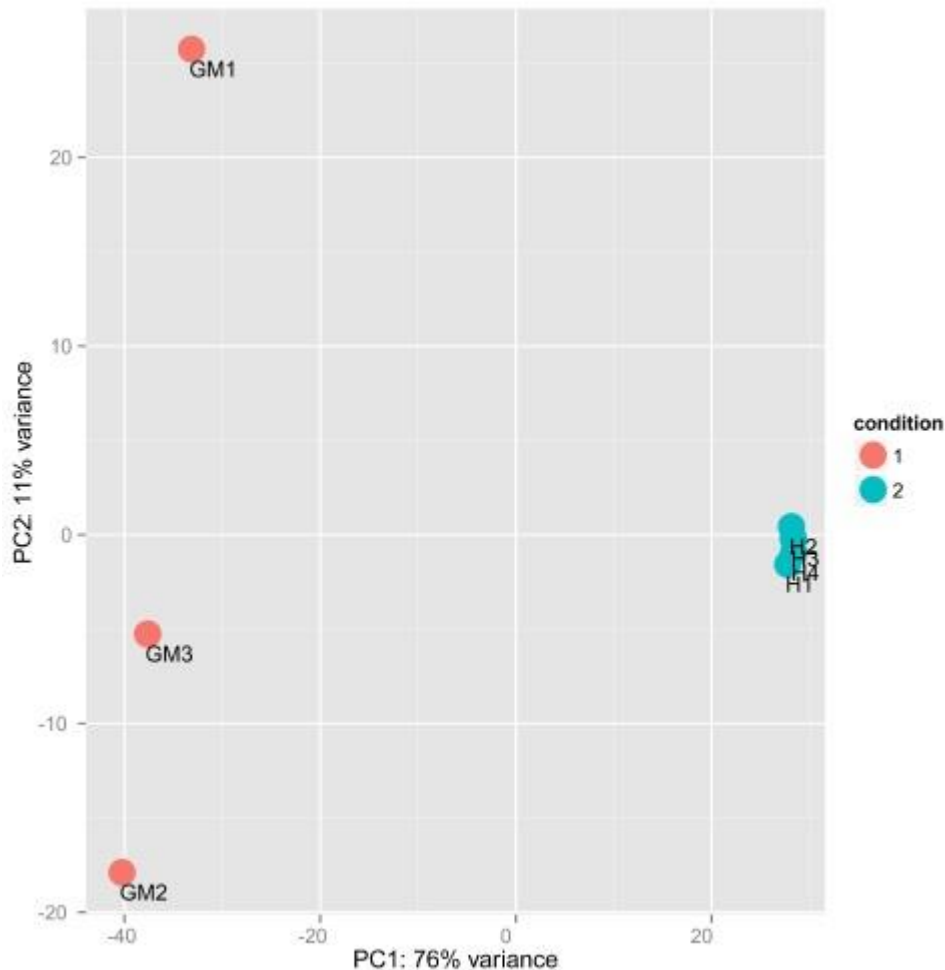
RNA-seq data analysis workflow

- Quality control of raw reads
- Preprocessing (trimming / filtering) if needed
- Alignment (=mapping) to reference genome
- Alignment level quality control
- Quantitation
- **Experiment level quality control**
- Differential expression analysis
- Visualization of reads and results in genomic context

Experiment level quality control

- **Getting an overview of similarities and dissimilarities between samples allows you to check**
 - Do the experimental groups separate from each other?
 - Is there a confounding factor (e.g. batch effect) that should be taken into account in the statistical analysis?
 - Are there sample outliers that should be removed?
- **Several methods available**
 - MDS (multidimensional scaling)
 - PCA (principal component analysis)
 - Clustering

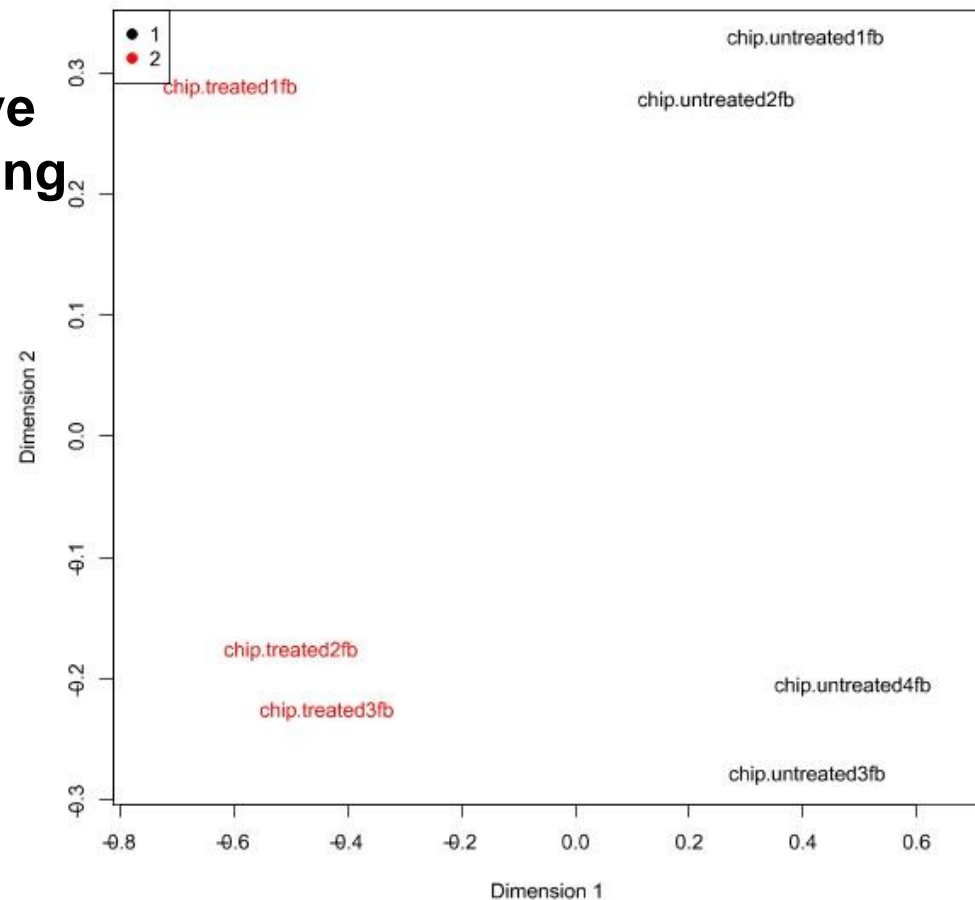
PCA plot by DESeq2



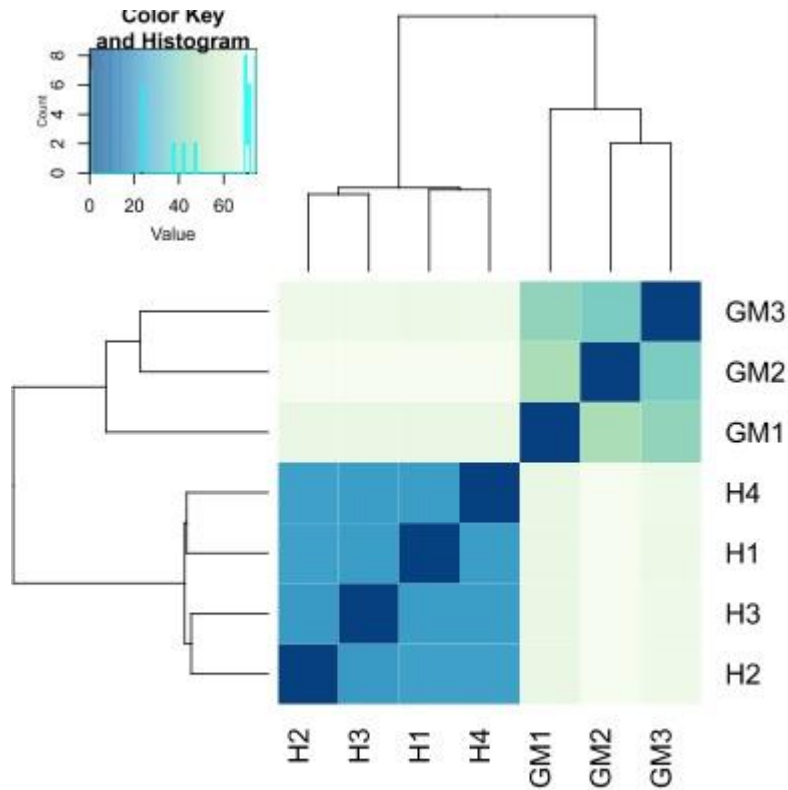
- The first two principal components, calculated after variance stabilizing transformation
- Indicates the proportion of variance explained by each component
 - If PC2 explains only a small percentage of variance, it can be ignored

MDS plot by edgeR

- Distances correspond to the logFC or biological coefficient of variation (BCV) between each pair of samples
- Calculated using 500 most heterogenous genes (that have largest dispersion when treating all samples as one group)



Sample heatmap by DESeq2



- **Euclidean distances between the samples, calculated after variance stabilizing transformation**

RNA-seq data analysis workflow

- Quality control of raw reads
- Preprocessing (trimming / filtering) if needed
- Alignment (=mapping) to reference genome
- Alignment level quality control
- Quantitation
- Experiment level quality control
- **Differential expression analysis**
- Visualization of reads and results in genomic context

Software packages for DE analysis

- **edgeR**
- **DESeq2**
- **sleuth**
- **DEXSeq**
- **Cuffdiff, Ballgown**
- **Limma + voom, limma + vst**
- **...**

Comparison of software packages for detecting differential expression in RNA-seq studies

Fatemeh Seyednasrollah, Asta Laiho and Laura L. Elo

Comprehensive evaluation of differential expression analysis methods for RNA-seq data

Franck Rapaport¹, Raya Khanin¹, Yupu Liang¹, Azra Krek¹, Paul Zumbo^{2,4},
Christopher E. Mason^{2,4}, Nicholas D. Socci¹, Doron Betel^{3,4}

¹Bioinformatics Core, Memorial Sloan-Kettering Cancer Center, New York

How many biological replicates are needed in an RNA-seq experiment and which differential expression tool should you use?

Medical College, New York

Cornell Medical College, New York

Medical College, New York

NICHOLAS J. SCHURCH,^{1,6} PIETÀ SCHOFIELD,^{1,2,6} MAREK GIERLIŃSKI,^{1,2,6} CHRISTIAN COLE,^{1,6}
ALEXANDER SHERSTNEV,^{1,6} VIJENDER SINGH,² NICOLA WROBEL,³ KARIM GHARBI,³
GORDON G. SIMPSON,⁴ TOM OWEN-HUGHES,² MARK BLAXTER,³ and GEOFFREY J. BARTON^{1,2,5}

A comparison of methods for differential expression analysis of RNA-seq data

BMC Bioinformatics 2013, **14**:91 doi:10.1186/1471-2105-14-91

Charlotte Soneson (Charlotte.Soneson@isb-sib.ch)
Mauro Delorenzi (Mauro.Delorenzi@unil.ch)

Differential gene expression analysis

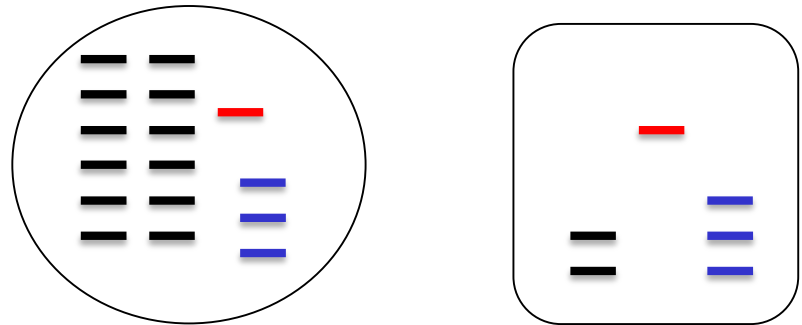
- **Normalization**
- **Dispersion estimation**
- **Log fold change estimation**
- **Statistical testing**
- **Filtering**
- **Multiple testing correction**

Differential expression analysis: Normalization

Normalization

➤ For comparing gene expression between (groups of) samples, normalize for

- Library size (number of reads obtained)
- RNA composition effect



➤ The number of reads for a gene is also affected by transcript length and GC content

- When studying differential expression you assume that they stay the same

A comprehensive evaluation of normalization methods for Illumina high-throughput RNA sequencing data analysis

Marie-Agnès Dillies*, Andrea Rau*, Julie Aubert*, Christelle Hennequet-Antier*, Marine Jeanmougin*, Nicolas Servant*, Céline Keime*, Guillemette Marot, David Castel, Jordi Estelle, Gregory Guernec, Bernd Jagla, Luc Jouneau, Denis Laloë, Caroline Le Gall, Brigitte Schaëffer, Stéphane Le Crom*, Mickaël Guedj*, Florence Jaffrézic* and on behalf of The French StatOmique Consortium

- “FPKM and TC are ineffective and should be definitely abandoned in the context of differential analysis”
- “In the presence of high count genes, only DESeq and TMM (edgeR) are able to maintain a reasonable false positive rate without any loss of power”



Normalization by edgeR and DESeq

- **Aim to make normalized counts for non-differentially expressed genes similar between samples**
 - Do not aim to adjust count distributions between samples
- **Assume that**
 - Most genes are not differentially expressed
 - Differentially expressed genes are divided equally between up- and down-regulation
- **Do not transform data, but use normalization factors within statistical testing**

Normalization by edgeR and DESeq – how?

➤ DESeq(2)

- Take geometric mean of gene's counts across all samples
- Divide gene's counts in a sample by the geometric mean
- Take median of these ratios → sample's normalization factor (applied to read counts)

➤ edgeR

- Select as reference the sample whose upper quartile is closest to the mean upper quartile
- Log ratio of gene's counts in sample vs reference → M value
- Take weighted trimmed mean of M-values (TMM) → normalization factor (applied to library sizes)
 - Trim: Exclude genes with high counts or large differences in expression
 - Weights are from the delta method on binomial data

Do not use RPKM/FPKM for differential expression analysis with edgeR and DESeq2!

- Reads (or fragments) per kilobase per million mapped reads.
- **Normalizes for gene length and library size:**
 - 20 kb transcript has 400 counts, library size is 20 million reads
→ $RPKM = (400/20) / 20 = 1$
 - 0.5 kb transcript has 10 counts, library size is 20 million reads
→ $RPKM = (10/0.5) / 20 = 1$
- **RPKM/FPKM can be used only for reporting expression values, not for testing differential expression**
 - In DE analysis raw counts are needed to assess the measurement precision correctly

Differential expression analysis: Dispersion estimation

Dispersion

- **When comparing gene's expression levels between groups, it is important to know also its within-group variability**
- **Dispersion = $(BCV)^2$**
 - BCV = gene's biological coefficient of variation
 - E.g. if gene's expression typically differs from replicate to replicate by 20% (so $BCV = 0.2$), then this gene's dispersion is $0.2^2 = 0.04$
- **Note that the variability seen in counts is a sum of 2 things:**
 - Sample-to-sample variation (dispersion)
 - Uncertainty in measuring expression by counting reads

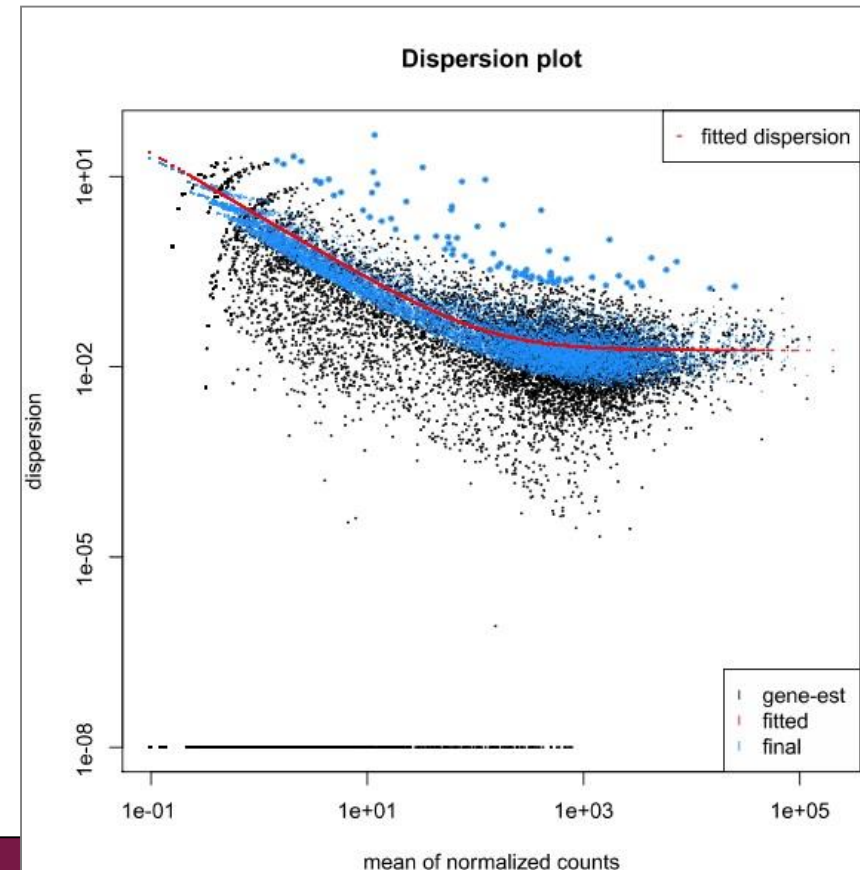


How to estimate dispersion reliably?

- **RNA-seq experiments typically have only few replicates**
→ it is difficult to estimate within-group variability
- **Solution: pool information across genes which are expressed at similar level**
 - assumes that genes of similar average expression strength have similar dispersion
- **Different approaches**
 - edgeR
 - DESeq2

Dispersion estimation by DESeq2

- Estimates genewise dispersions using maximum likelihood
- Fits a **curve** to capture the dependence of these estimates on the average expression strength
- Shrinks **genewise values towards the curve** using an empirical Bayes approach
 - The amount of shrinkage depends on several things including sample size
 - Genes with high gene-wise dispersion estimates are dispersion outliers (blue circles above the cloud) and they are not shrunk



Differential expression analysis: Statistical testing

Generalized linear models

➤ **Model the expression of each gene as a linear combination of explanatory factors (eg. group, time, patient)**

- $y = a + (b \cdot \text{group}) + (c \cdot \text{time}) + (d \cdot \text{patient}) + e$

y = gene's expression

a , b , c and d = parameters estimated from the data

a = intercept (expression when factors are at reference level)

e = error term

➤ **Generalized linear model (GLM) allows the expression value distribution to be different from normal distribution**

- Negative binomial distribution used for count data

Statistical testing

➤ edgeR

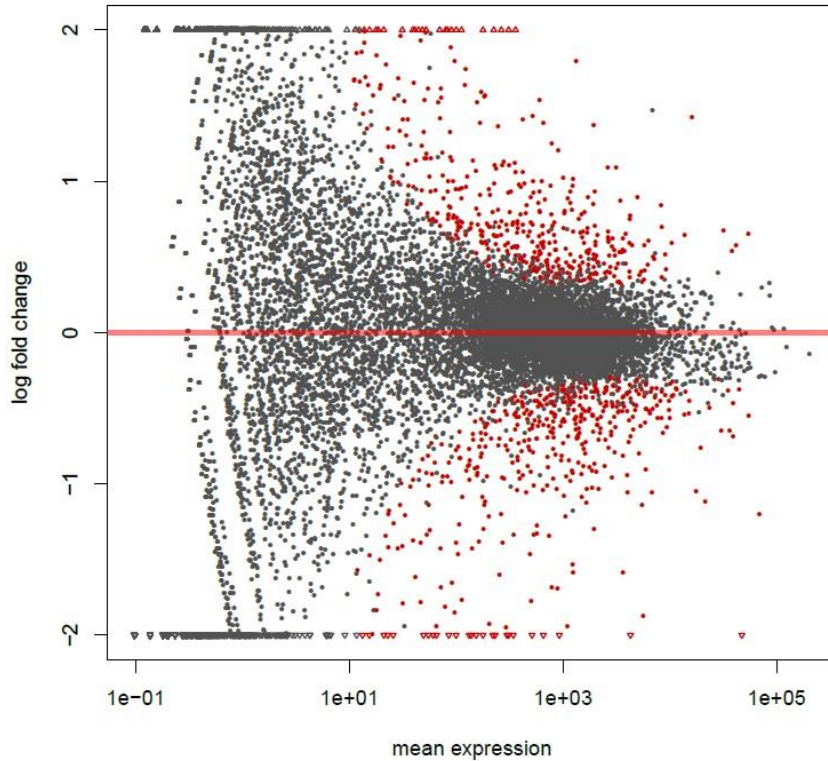
- Two group comparisons
 - Exact test for negative binomial distribution.
- Multifactor experiments
 - Generalized linear model, likelihood ratio test.

➤ DESeq2

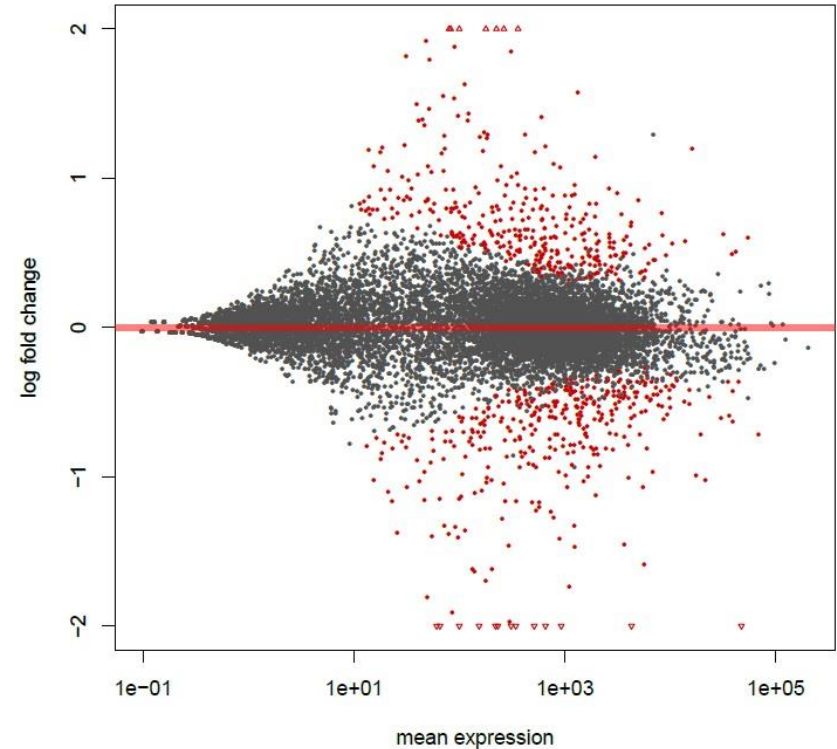
- Shrinks log fold change estimates toward zero using an empirical Bayes method
 - Shrinkage is stronger when counts are low, dispersion is high, or there are only a few samples
- Generalized linear model, Wald test for significance
 - Shrunk estimate of log fold change is divided by its standard error and the resulting z statistic is compared to a standard normal distribution

Fold change shrinkage by DESeq2

MA-plot, no FC shrinkage, FDR = 0.05



MA-plot, FDR = 0.05



Multiple testing correction

- We tests thousands of genes, so it is possible that some genes get good p-values just by chance
- To control this problem of false positives, p-values need to be corrected for multiple testing
- Several methods are available, the most popular one is the **Benjamini-Hochberg correction (BH)**
 - largest p-value is not corrected
 - second largest $p = (p * n) / (n-1)$
 - third largest $p = (p * n) / (n-2)$
 - ...
 - smallest $p = (p * n) / (n - n + 1) = p * n$
- The adjusted p-value is FDR (false discovery rate)

Filtering

- **Reduces the severity of multiple testing correction by removing some genes (makes n smaller)**
- **Filter out genes which have little chance of showing evidence for significant differential expression**
 - genes which are not expressed
 - genes which are expressed at very low level (low counts are unreliable)
- **Should be independent**
 - do not use information on what group the sample belongs to
- **DESeq2 selects filtering threshold automatically**

edgeR result table

- **logFC = log2 fold change**
- logCPM = average log2 counts per million
- Pvalue = raw p-value
- **FDR = false discovery rate (Benjamini-Hochberg adjusted p-value)**

	logFC	logCPM	PValue	FDR
FBgn0039155	-4.68610492988647	6.03098899098003	5.67559613973167e-123	3.31349310601679e-119
FBgn0029167	-2.22179416128475	8.24421076784694	1.36882477184621e-55	6.40746875701213e-52
FBgn0034736	-3.48749671162214	4.04006374116452	1.4075253924686e-49	4.39241757476368e-46
FBgn0035085	-2.51385564715956	5.53462890050981	3.0858842886838e-49	7.22251217766443e-46
FBgn0039827	-4.25961693280824	4.59870730232648	1.68130004303576e-47	3.14806620058016e-44
FBgn0000071	2.75298722125534	4.68516991052067	6.74381730816232e-47	1.05226029398359e-43
FBgn0029896	-2.42499289598	5.18422350459525	2.30767413477857e-42	3.08634932139957e-39

DESeq2 result table

- baseMean = mean of counts (divided by size factors) taken over all samples
- **log2FoldChange** = log2 of the ratio meanB/meanA
- lfcSE = standard error of log2 fold change
- stat = Wald statistic
- pvalue = raw p-value
- **padj** = Benjamini-Hochberg adjusted p-value

	baseMean	log2FoldChange	lfcSE	stat	pvalue	padj
FBgn0026562	47282.42	-2.4	0.08	-30.26	4.159e-201	3.309e-197
FBgn0039155	924.27	-4.46	0.16	-27.04	4.476e-161	1.781e-157
FBgn0029167	4287.44	-2.21	0.08	-26.75	1.107e-157	2.937e-154
FBgn0035085	654.94	-2.5	0.11	-22.08	5.278e-108	1.050e-104
FBgn0034736	231.7	-3.29	0.18	-18.28	1.261e-74	2.006e-71
FBgn0000071	359.53	2.6	0.14	17.98	2.741e-72	3.635e-69
FBgn0034434	153.84	-3.69	0.21	-17.26	9.008e-67	1.024e-63
FBgn0039827	342.77	-3.83	0.23	-16.54	1.742e-61	1.733e-58
FBgn0029896	513.08	-2.34	0.14	-16.29	1.168e-59	1.033e-56
FBgn0052407	220.26	-2.2	0.15	-14.99	8.597e-51	6.841e-48
FBgn0037754	299.03	-2.23	0.15	-14.94	1.916e-50	1.386e-47

Statistical testing for differential expression: things to take into account

- **Biological replicates are important!**
- **Normalization is required in order to compare expression between samples**
 - Different library sizes
 - RNA composition bias caused by sampling approach
- **Raw counts are needed to assess measurement precision**
 - Counts are the "the units of evidence" for expression
 - No FPKMs thanks!
- **Multiple testing problem**

Summary of differential expression analysis steps and files

- **Quality control / Read quality with FastQC** → html report
- (Preprocessing / Trim reads with Trimmomatic → FASTQ)
- (Utilities / Make a list of file names → txt)
- **Alignment / HISAT2 for paired end reads** → BAM
- **Quality control / RNA-seq quality metrics with RseQC** → pdf
- **RNA-seq / Count aligned reads per genes with HTSeq** → tsv
- **Utilities / Define NGS experiment** → tsv
- **Quality control / PCA and heatmap of samples with DESeq2** → pdf
- **RNA-seq / Differential expression using DESeq2** → tsv
- **Utilities / Annotate Ensembl identifiers** → tsv

